Applications of Data Mining Techniques to Electric Load Profiling

A thesis submitted to the University of Manchester Institute of Science and Technology for the degree of Doctor of Philosophy

2000

Barnaby D Pitt

Electrical and Electronic Engineering

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university, or other institute of learning.

Abstract

Data Mining is a broad term for a variety of data analysis techniques applied to the problem of extracting meaningful knowledge from large, noisy databases. An important feature present in most of these techniques is an ability to adapt to the local characteristics of the data. Such techniques are applied to electric load profiling tasks; load profiling consists of modelling the way in which daily load shape (load profile) relates to various factors such as weather, time and customer characteristics. An implementation of an adaptive load profiling methodology is presented.

An atom is defined as a set of load profiles for which certain predictor attributes take identical values. Weather-dependent loads are recovered from the raw data by subtracting certain atomic profiles, and weather dependency modelled by the method of Multivariate Adaptive Regression Splines.

Nominally weather-free load profiles are constructed from this model, and aggregated into new atoms. These atoms are subjected to adaptive clustering algorithms, with the objective of condensing the vast amount of data in the original database into a small number of representative (*end*) profiles each pertaining to a particular subset of the domain of the database. The clustering of individual customers' profiles (rather than atoms) is investigated as an extension to clustering of atoms.

Various possible extensions to and alternatives to the methodology are discussed.

Chapter 1 — Why Data Mining?

> 'Computers have promised us a fountain of wisdom but delivered a flood of data' - A frustrated MIS executive, quoted in [1].

1.1 The Need For Data Mining

All manner of businesses and research organisations have vast collections of data stored in databases and flat files. As the cost of data storage becomes lower and lower, and the means for collecting data continue to multiply, the volume of data accessible to researchers can only be expected to increase further and further; inevitably, an ever increasing proportion of this data is never seen by human eyes. Outcomes of database queries, and the statistics and graphics produced by statistical software, are capable of answering some of the questions that the proprietors of databases may have about their data. However the sheer bulk of that data may be such that important underlying structures in the data may never be discovered: there are so many potentially 'good' questions we might ask about the data that only a tiny fraction of such questions are ever posed, less answered.

The term Data Mining (nearly synonymous with the term Knowledge Discovery in Databases) is a blanket term which describes the many ways in which statisticians and data engineers are attempting to automate the process by which intelligible knowledge can be derived from large databases. Frawley, Piatetsky-Shapiro and Matheus give a definition,

> 'The non-trivial extraction of implicit, previously unknown, and potentially useful information from data'

in their thorough overview of data mining in [1]. Another good introductory paper on the subject is found in [2].

1.2 Volume Versus Interpretability

It is common sense that a small volume of information (such as a concise set of rules about some data, or a well conceived graphical display representing features of the data) convey more meaning (whether to a data engineer, a field expert, or a lay person), than disks or reams filled with raw data. However it is equally obvious that the total amount of information contained in a large database is greater than that contained in any at-a-

glance distillation of the database; that is, we gain insight only at the expense of detail. We can regard data mining, in part, as the search for representations of data which strike the best compromise between volume and interpretability. Exactly how much volume reduction is desirable will vary enormously according to the intended use of the reduced data.

1.3 Specificity Versus Generality

In any relational data, the two extreme representations of the data are to present the entire database (so that every record in the database has a unique description); and to present a single 'average' data record (so that every record in the database is associated with some global modal or mean description). In between these extremes are representations of the data which agglomerate records by some criteria, so that every record has a description common to all records in the same agglomeration. Many data mining tasks can be seen as searches for the correct *data resolution*; that is, searches for partitions of records which are coarse enough that the number of cells is not overwhelming, but fine enough that all the records in a cell comply well with any generalisation we might make about them. A crucial feature of most data mining techniques is their ability to represent different regions in the total data space at *different resolutions*: where data are more diverse finer partitions are sought, with the objective that the subset of records in any cell are of comparable homogeneity to the subset of records in any other cell (see section 4.1).

1.4 Concepts of Informativeness and Utility

In the preceding three sections we have touched on ideas of informativeness and usefulness of data representations. A set of rules or generalisations derived from a database has less utility if very bulky, but carries less information if the representation is too coarse or indiscriminate.

By whatever method knowledge is to be 'mined' from data, it is important to establish measures for the utility and informativeness of discovered knowledge. Formal measures, such as statistical significance of derived rules; the amount of total variance explained by a model; and measures for the information content of data representations (deriving from *information theory*), can be used to guide searches through data. It may be equally important that informal criteria of utility and informativeness play a part in

the design and application of a data mining technique.

A practitioner of data mining who has a good understanding of the scientific or social scientific field from which the data derives (and in which discovered knowledge might be applied) has a much better chance of finding useful and informative representations of the data than a practitioner who sees the data as just tables of numbers and symbols. Domain heuristics, and intuition about the nature of hidden structures, should be utilised at every stage in the data analysis. Furthermore, if formal measures indicate that a particular representation is maximally informative, but a human with understanding of the problem domain find some modified representation more informative, the second representation is likely to be preferable.[1]

1. An exception might arise where the mined representation is to be used as input to another computer program, such as a knowledge based system or forecasting program, so that human interpretability of representations is not paramount.

Chapter 2 — The Scope of Data Mining

This chapter briefly describes the variety of approaches to the extraction of intelligible knowledge from large noisy databases, which fall under the umbrella of 'Data Mining'. The large variation in the nature and quality of data in databases is also covered and some notation introduced. Chapters 3-5 describe concepts which recur in several data mining methods (information theory; hierarchical partitioning and decision trees; variable reduction) and chapters 6-8 describe some important data mining techniques in some detail. Some techniques of only limited relevance to the task and methodology eventually selected are dealt with more scantly.

2.1 The Breadth of the Term 'Data Mining'

Data Mining (abbreviated *DM*) is currently a fashionable term, and seems to be gaining slight favour over its near synonym Knowledge Discovery in Databases *(KDD)*. Since there is no unique definition, it is not possible to set rigid boundaries upon what is and is not a data mining technique; the definition proffered in section 1.1 could conceivably cover virtually the entire body of statistics and of knowledge based systems, and a good deal of current research in database technology and machine learning. In this tract we shall somewhat limit the scope of the term to *exclude* techniques whose principal domain is intermediate or small databases which contain little or no discrepancies, anomalies, omissions or noise in the data.[1] Further, it is convenient for us to discriminate between 'data mining' and 'classical' statistical methods (like analysis of variance and parametric regression, which operate globally on a set of variables), although such techniques often have 'walk-on parts' in what we shall call data mining.

We are primarily concerned with techniques which seek to extract important features from *large*, *noisy*, *real-world* databases which may have many missing entries and inconsistencies. Real-world databases are characterised by the fact that unlike data derived from controlled experiments, such data tend to be sparse in most regions of the variable space — records or events which are less common usually have less representation in the database. Accordingly we seek methods which are capable of adapting well to varying levels of data density and noise; *adaptive* methods automatically search and analyse the

1. This type of problem is often termed 'learning from examples' in Artificial Intelligence and Knowledge Based Systems literature.

denser and more heterogeneous regions of variable space more thoroughly.

The principal areas of data mining, as it has been described above, might be broken down into

1. Exploratory data analysis and variable reduction
2. Visualisation techniques
3. Regression — particularly non-parametric regression, adaptive regression, hierarchical regression
4. Classification (aka supervised learning)
5. Clustering (aka unsupervised learning)
6. Hybrids of any of the above.

2.2 Types and Qualities of Data

2.2.1 Predictors and Responses

Let the variables (attributes) in the data set be denoted by $X_1, \ldots, X_j, \ldots, X_J; Y_1, \ldots, Y_k, \ldots, Y_K$ where the $X_j; (1 \le j \le J)$ are *predictor* (independent) variables and the $Y_k; (1 \le k \le K)$ are *response* (dependent) variables. The selection of this division is not always trivial, and is part of the task definition in a data mining exercise. Moreover, there may be tasks in which some or all of the variables are to be considered as both predictors and responses.

2.2.2 Types of Domain

Let the *cases*, which we will sometimes refer to as *records* or *observations*, be denoted $C_1, \ldots, C_i, \ldots, C_N$ and let the ith case, $1 \le i \le N$, have associated attribute values

$$X_{1i} = x_{1i} \in \underset{\sim}{X}_1$$
$$\ldots$$
$$X_{Ji} = x_{Ji} \in \underset{\sim}{X}_J$$

$$Y_{1i} = y_{1i} \in \underset{\sim}{Y}_1$$
$$\ldots$$
$$Y_{Ki} = y_{Ki} \in \underset{\sim}{Y}_K$$

where $\underset{\sim}{X}_j$, $\underset{\sim}{Y}_k$ are the domain sets or *domains* of the respective X_j, Y_k. The domains generally fall into one of four categories; consider some predictor X_j (analogous descriptions apply to responses Y_k):

1. Categorical. Categorical variables take one of a finite number $|\underset{\sim}{X}_j|$ of unordered discrete values $x_{ji} \in \underset{\sim}{X}_j = \{x_{j1}, x_{j2}, \ldots, x_{j|\underset{\sim}{X}_j|}\}$.

2. Ordered. Ordered variables take one of a number (possibly infinite) of discrete values $x_{ji} \in \underset{\sim}{X}_j = \{x_{j1} < x_{j2} < \ldots < x_{j|\underset{\sim}{X}_j|}\}$. Often $\underset{\sim}{X}_j$ is a finite set of contiguous integers.

3. Hierarchical. Hierarchical variables are categorical variables whose categories are arranged in some hierarchy, usually an *'is-a'* (i.e. transitive) hierarchy. For example, if X_j records the type of animal in a vetinary database, taking values {boxer, terrier, cat, dog, iguana, mammal, reptile}, it admits an *is-a* hierarchy including relationships like {terrier *is-a* dog, cat *is-a* mammal, terrier *is-a* mammal,...}.

4. Continuous. Real variables, whose domain is a (possibly infinite) range of real numbers, $[X_j^{min}, X_j^{max}]$.

2.2.3 Noisy Data

There are principally two sources of noise which arise in databases (although they are generally treated alike). First, some or all of the attribute values for any given observation might be of dubious accuracy: if they are measurements they may be imperfect or may be inexact due to rounding (continuous quantities cannot be measured exactly); if they are derived from questionnaires, they may be subjective responses to questions and hence not wholly reliable.

Second (and generally more importantly) the attribute values, particularly for response variables, are often samples drawn from populations of random variables. To make matters worse, the underlying probability distribution for these random variables is almost never known in real-world databases.

For a single continuous response variable Y we might propose a model

$$Y_i = \bar{Y}_i + \varepsilon_i^{(m)} + \varepsilon_i^{(s)}; (1 \le i \le N) \quad \text{(EQ 1)}$$

where Y_i is the value observed, $\bar{Y}_i$ the expectation of Y_i (the population mean, conditional on the values of $X_{1i}, \ldots, X_{Ji}$).

$\varepsilon_i^{(m)}$ is the (additive) error due to measurement, $\varepsilon_i^{(s)}$ the (additive) error due to sampling (i.e. the deviation of Y_i from $\bar{Y}_i + \varepsilon_i^{(m)}$ due to the inherent randomness of Y_i. Since the ε's cannot (usually) be separated, we write $\varepsilon_i^{(ms)} = \varepsilon_i^{(m)} + \varepsilon_i^{(s)}$, and make some assumption about the distribution of $\varepsilon_i^{(ms)}$, often that it has zero-mean Gaussian distribution of unknown variance σ^2 (which may be approximated from the sample variance S^2).

Where there are multiple continuous response variables, the situation becomes far more complicated, since we are concerned with the *joint* distribution of $\underline{Y} = (Y_1, \ldots, Y_K)^T$. We might write

$$\underline{Y}_i = \underline{\bar{Y}}_i + \underline{\varepsilon}_i^{(ms)}; (1 \le i \le N, 1 \le k \le K) \quad \text{(EQ 2)}$$

on case C_i (with $\underline{\varepsilon}_i^{(ms)}$ a k-vector of errors due to measurement and sampling). However $\bar{Y}_{ki}$ is now the multivariate population mean conditional on the values of $X_{1i}, \ldots, X_{Ji}, Y_{1i}, \ldots, Y_{(k-1)i}, Y_{(k+1)i}, \ldots, Y_{Ki}$, and any assumption that the observed deviations $\underline{\varepsilon}_i^{(ms)}$ have a certain multivariate probability distribution is likely to be theoretically tenuous, and demonstrable only empirically.

Where noise (due to measurement and sampling) is appreciable, any rules and/or representations which we may derive for the data *must* be 'fuzzy' (inexact) in some sense. We can associate confidence intervals with estimators of continuous variables; we can give measures of confidence for rules which assert exact logical truths; allow fuzzy membership of any discovered groupings of cases; and allow discrete probability distributions for class memberships (rather than a uniquely determined class) where cases are to be classified.

Noise in data can be reduced by at least three means:

5. Data Smoothing. This includes exponential smoothing of time series and the fitting of smoothing curves/surfaces/hypersurfaces to noisy data.

6. Data Aggregation. We aggregate cases which have the same or similar values amongst their predictors. Their aggregate response(s) should be determined in a sensible manner, such as taking mean or modal value(s). Conversely we may aggregate those cases which have the same or similar values amongst their response variables. In this case the aggregate predictor variables should be determined in a sensible manner, often by partitioning the predictors' domains (e.g. values {Mon, Tue, Wed, Thu, Fri, Sat, Sun} replaced by {workday, Sat, Sun}, for a day-of-week variable). A further option is to aggregate those cases which are similar in the values taken by both their predictors *and* responses.

7. Identification and Exclusion of Outliers. Outliers may be 'good' or 'bad' outliers. A bad outlier contains erroneous (severely mismeasured) data. A *type-1 good outlier* has been recorded with sufficient fidelity, but due to the inherent randomness in the data, has response values which are exceptional given the values taken by its predictors. A *type-2 good outlier* is a case which has exceptional values in its predictor variables— the values taken by its predictors are not 'near' those of the other cases. Type 2 good outliers may be discarded (since one cannot reliably make inferences on the basis of a unique case) or retained according to discretion.

   Type 1 good outliers can be particularly instructive, whereas we would usually prefer to identify and exclude bad outliers from consideration. Unfortunately they are usually very difficult to distinguish from one another, although a series of exceptional cases, say, over a certain time period, might point to faulty measuring equipment. Outliers of any type are often excluded or given reduced weight whilst building models, but then analysed in relation to the constructed model.

Any form of noise reduction may take place wholly in advance of other data analysis, but is often an ongoing and integral part of the data mining process. Noise reduction, and the production of inexact rules and representations, are both crucial to the analysis of noisy data.

2.2.4 Incomplete Data

Large real-world databases more often than not contain missing values for some cases in some attributes. The simplest way to deal with such cases is to discard them, but this might 'throw out the baby with the bathwater' and is infeasible when many or most

of the cases have a missing value. Seeking good ways to deal with missing values is an important area of data mining research, and certain techniques are capable of coping with very substantial numbers of missing entries. Where only a few data are missing, it is common practice to replace each missing datum with the variable value which is mean or modal for the variable in question, *given the values of the known attributes*, which may be variously determined. Other schemes allow fuzzy values for missing entries, so that a case may fall partly into one part of the model, and partly into another or others.

2.2.5 Inconclusive Data

Due to the local sparsity of the data, the randomness of some attributes, and the random manner in which the cases in the database are selected from all possible cases, the data may be inherently inconclusive: there may be parts of the attribute space for which no reliable rules apply. A further and very important source of inconclusiveness in data is the absence of certain predictors which would be necessary to fully describe the variation in the responses. Typically there are practically limitless numbers of predictor variables which might possibly affect responses, only a fraction of which can feasibly be recorded.

Inconclusive data is commonly handled by the use of inexact rules, and data aggregation (as with noisy data, section 2.2.3). It may sometimes be necessary to accept the inconclusiveness of the data, particularly in the sparsest parts of the domain, and to avoid making any assertions about parts of the data where no assertions can be relied upon. ‘Giving up’ on part of the data can still be instructive if we can use the results of data mining to guide future database design— particularly, identifying regions of data space in which records are too sparse to draw conclusions, so that more data of that ilk can be collected.

Chapter 3 — A Crash Course in Information Theory for Data Mining

3.1 Introduction

Most of the statistics used to describe data in this tract are widely known: means, modes, variances, Euclidean distances, and so on. Information Theory is an area of study which was initiated by theorists studying the communication and coding of signals, and accordingly the nomenclature (sources, receivers, messages, channels, codes, and so on) may be less familiar. Information theory seeks to measure the amount of information in a *message*— a sample of a random variable, or a time series of a random variable; the amount of information preserved in the presence of noise; and the amount of information conveyed by one random variable (*source*) about another random variable (*receiver*). Information theoretic measures appear frequently in data mining literature, particularly in the construction of decision trees (chapter 4). This chapter is primarily inspired by [4] and [5].

Consider, by way of example, a digitally stored data file. Such data files are frequently compressed (or *coded*) before being electronically mailed, in order to lessen transmission time. It is possible to compress data files (by factors of ten or more in extreme cases) because there is redundancy in the manner in which the uncompressed file is stored. For example, digital images may contain large areas with little spacial variation in shading; a database of customer records is likely to contain many groups of conceptually similar customers, whose recorded data differ only in a few variables, or only by a small margin in some variables.

Commercial data compression programs are generally ignorant of the precise nature and meaning of the data sets they are employed on. When certain information is known about the nature of the data (in particular, information about the statistical distributions of variables, and certain measures of correlation between variables), it is generally possible to compress the data much further. Loosely speaking, the *total information content* in a data set can be thought of as the size (in bits) of the most efficient coding possible— various theorems (known as fundamental coding theorems) state that codings exist[1] which can compress data sets to within an arbitrarily small margin of their theoretical

1. These theorems do not construct the actual codings, which are generally unknown.

total information contents.

Whilst the data analyst is not generally interested in coding his or her data set for efficient transmission, the theoretical information can be of interest. In particular, when a data set is simplified or represented in a new form, it is desirable that the new representation contains as much as possible of the information conveyed by the original data.

3.2 Discrete Memoriless Sources

Let a source S transmit a series of values of a discrete random variable X, which takes the values $\{x_1, \ldots, x_l, \ldots, x_L\}$ with respective probabilities $\{p_1, \ldots, p_l, \ldots, p_L\}$. If these probabilities are independent of the values of X already transmitted, then S is a discrete memoriless source. Let the transmission of a particular value of X be called an *event* $E_l; \{1 \le l \le L\}$.

Define the *self-information* of an event to be

$$I(E_l) = -\log_2 p_l; (1 \le l \le L) \quad \text{(EQ 3)}$$

and the *entropy*— the average (self) information— of the source variable to be

$$H(X) = \overline{I(E_l)} = -\sum_{l=1}^{L} p_l \log_2 p_l \quad \text{(EQ 4)}$$

Base 2 logarithms are used so that the information is measured in bits. We shall drop the 2 in subsequent equations. The entropy of a memoriless source is entirely determined by the probability distribution of the associated random variable X.

The presence of the log in the definition of self information can be justified thus: suppose an experiment with l possible outcomes, probabilities $\{p_1, \ldots, p_l, \ldots, p_L\}$ is performed twice, and the results transmitted to a third party. If the two outcomes are independent, the information associated with transmitting the two results one by one should be the same as the information associated with transmitting the outcome of a single compound equivalent experiment with l^2 possible outcomes with probabilities $\{p_{l_1 l_2} \mid 1 \le l_1, l_2 \le L\}$. We require $I(E_{l_1 l_2}) = I(E_{l_1}) + I(E_{l_2})$, satisfied by (EQ 3). The entropy measures the informativeness of the source as a whole, and accordingly weights each of the self informations with their frequencies of occurrence. $H(p_1, \ldots, p_L)$ is maximised when $p_l = L^{-1}$ for $1 \le l \le L$, i.e. for a uniform distribution.

3.3 Discrete Memoriless Channels

A discrete *channel* is defined by its joint probability structure for its discrete source variable *X* and receiver variable *Y* (receiver random variables have the same associated information properties as source random variables).

Let X take values $\{x_1, \ldots, x_n\}$ with probabilities $\{p_1, \ldots, p_n\}$ and let Y take values $\{y_1, \ldots, y_m\}$ with probabilities $\{p_1, \ldots, p_m\}$. Let $\{p_{ij} \mid (1 \le i \le n)\ (1 \le j \le m)\}$ be the joint probabilities for *X* and *Y*; if these are independent of previous transmissions, the associated channel is a discrete memoriless channel. Such a channel is noise-free only if $p_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$, and we can think of the joint distribution as characterising the noise properties of a channel, and vice versa. We associate five entropy measures with a communications scheme:

$$H(X) = -\sum_{i=1}^{n}\left[\sum_{j=1}^{m} p_{ij}\log\left(\sum_{j=1}^{m} p_{ij}\right)\right] \quad \text{(EQ 5)}$$

$$H(Y) = -\sum_{j=1}^{m}\left[\sum_{i=1}^{n} p_{ij}\log\left(\sum_{i=1}^{n} p_{ij}\right)\right] \quad \text{(EQ 6)}$$

$$H(Y|X) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}\log p\{y_j|x_i\} \quad \text{(EQ 7)}$$

$$H(X|Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}\log p\{x_i|y_j\} \quad \text{(EQ 8)}$$

$$H(X, Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}\log p_{ij} \quad \text{(EQ 9)}$$

These are, respectively, the source and receiver entropies (or marginal entropies); the entropy of *Y* conditional on *X* (the average information per event received, given we know the event transmitted); the entropy of *X* conditional on *Y* (vice versa); and the joint entropy—the average information per pair of transmitted/received events: i.e. the average uncertainty (noise) of the channel. The entropies are related by

$$\begin{aligned} H(X, Y) &= H(X|Y) + H(Y) \\ H(X, Y) &= H(Y|X) + H(X) \end{aligned} \quad \text{(EQ 10)}$$

$$H(X) \geq H(X|Y)$$
$$H(Y) \geq H(Y|X) \qquad \text{(EQ 11)}$$

For a noise-free channel, the conditional entropies are zero, and the marginal entropies and the joint entropy are all equal. For a *random* channel (*X* and *Y* independent) the inequalities in (EQ 11) become equalities.

Finally define the *mutual information* between an event pair E_{ij} as

$$I(x_i;y_j) = \log\left(\frac{p_{ij}}{p_i p_j}\right) \qquad \text{(EQ 12)}$$

and the *mutual information between X and Y* as

$$I(X;Y) = \overline{I(x_i;y_j)} = \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}\log\left(\frac{p_{ij}}{p_i p_j}\right) \qquad \text{(EQ 13)}$$

and note that

$$I(X;Y) = H(X) + H(Y) - H(X, Y) \qquad \text{(EQ 14)}$$

$I(X;Y)$ is the amount of information conveyed by one random variable about the other; for a noise free channel, it is equal to the joint entropy; for a random channel, it is zero.

3.4 Continuous Memoriless Sources and Channels

Consider now a source variable *X* and received variable *Y* which take values in a continuous range [*a, b*]. We might try to approximate the information content of the source by discretising the range [*a, b*] into *n* equal cells and calculating the associated discrete probabilities and entropy. This approach is fundamentally flawed, as the resulting entropy is strongly dependent on *n*, and always tends to infinity as *n* tends to infinity; such an entropy measure is arbritary and meaningless. It is only to be expected that a variable taking unrestricted continuous values should have infinite entropy—consider that an arbritary real number cannot be completely stored in any amount of computer memory, and entropy is measured in bits. However we can derive meaningful measures which describe the information content of one message relative to another. The five entropies for a continuous memoriless channel are exact continuous analogues of (EQ 5) to (EQ 9), with integrals replacing the summations and continuous probability density functions *f*

replacing probabilities *p*. It can be shown (see e.g. [4]) that if *X* has range $(-\infty, \infty)$ and known finite variance σ^2, then *X* has maximal marginal entropy when it has a Gaussian distribution.

These entropies can be problematic, however: they may be negative; they may be infinite; and they are not invariant under linear transformations of the coordinate system. By analogy with the discrete case, define the *mutual* (or *trans-) information between continuous variables X, Y* as

$$I(X;Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\left(\log\frac{f(x, y)}{f(x)\cdot f(y)}\right)\mathrm{d}x\mathrm{d}y \qquad \text{(EQ 15)}$$

(EQ 14) also holds in the continuous case. Mutual information does not suffer from the above-mentioned problems.

Under the assumptions that *X* and *Y* have a joint Gaussian distribution with correlation parameter ρ, and known marginal variances,

$$I(X;Y) = -\frac{1}{2}\ln(1-\rho^2) \qquad (|\rho| \neq 1) \qquad \text{(EQ 16)}$$

3.5 Additive Gaussian Noise

The transinformation $I(X;Y)$ is also known as the rate of transmission; the maximum rate of transmission possible is known as the *channel capacity, I.* The *additive noise assumption* is that

$$Y = X + Z; \qquad Z \sim \phi(z) \qquad \text{(EQ 17)}$$

where ϕ is the distribution of additive noise *Z*. We further assume that *Z* and *X* are independent. The *Gaussian additive noise assumption* further stipulates that noise is zero-mean with a Gaussian (normal) distribution, with known *power* (i.e. variance) σ_z^2 and that the signal is zero-mean with known power σ_x^2 (the zero-mean assumptions are not strictly necessary). It can be shown (see e.g. [4]) that channel capacity (maximum transinformation) under these assumptions occurs when input (equivalently, output) is Gaussian, in which case

$$I = \frac{1}{2}\ln\left(1+\frac{S}{N}\right) \qquad \left\{\begin{array}{l} S = \sigma_x^2 \\ N = \sigma_z^2 \end{array}\right\} \qquad \text{(EQ 18)}$$

where *S/N* is known as the *signal-to-noise ratio.*

3.6 Continuous Band-Limited Signals

Now suppose that the source *X* is a continuous function of a continuous variable *t* (time). We no longer have a discrete sequence of continuous events; furthermore, the continuity constraint implies that the source has memory (is *stochastic*)— the probability density function for *X* at time *t* is not independent of previous transmissions. In general, the maximum transinformation for channels carrying continuous stochastic time signals is unknown. Many simplifications and assumptions must be made in order to derive a channel capacity for such channels.

First consider the simpler case in which a discrete sequence of continuous variables $X_t; (1 \le t \le n)$ are transmitted. Under Gaussian additive noise assumptions, with independent noises at any two time points, maximum transinformation has been shown to occur when the X_t are *n* independent Gaussian random variables, in which case

$$I = \frac{n}{2}\log\left(1 + \frac{S}{N}\right) \qquad \text{(EQ 19)}$$

Returning to the case where *X* and *Y* are continuous functions of time, we wish to reduce the infinity of time points to a discrete, finite sequence of time points (a *sample*) without loss of information in order to enable a calculation of channel capacity. In fact, the class of *band-limited, time-limited continuous signals* admits such an analysis. A continuous signal is *band-limited* if its fourier integrals have no frequency content beyond some frequency range $(-W, W)$, and time-limited if the signal is negligible outside some timespan [*-T/2*, *+T/2*]. The Sampling Theorem (see e.g. [4]) tells us that a band-limited signal is completely determined by its values at the points $\frac{\pm\pi n}{W}$ $n = 1, 2, 3, \ldots$, and time-limitation allows us to replace this infinite sample with a finite sample. Under independent Gaussian additive noise assumptions, and the assumption of constant signal power in [*-T/2*, *+T/2*], the Channel Capacity Theorem for Time-Limited, Band-Limited Signals (Shannon, see e.g. [4]) derives

$$I = \lim_{T \to \infty} \frac{1}{T}\ln\left(\frac{S+N}{N}\right)^{TW} \approx W\ln\left(1 + \frac{S}{N}\right) \qquad \text{(EQ 20)}$$

The assumption of band-limitation is essentially an imposition of a smoothness constraint on the signal. A continuous signal typically has a frequency spectrum which tails

off for higher frequencies; the frequency for which the fourier integral $F_{\omega}[x(t)]$ becomes ‘negligible’ is greater when the signal is less smooth. We can interpret the result (EQ 20) as a quantitative expression of the intuitive notions that more information is conveyed in the transmission of ‘bumpy’ signals than smooth signals, and that more information is conveyed when the signal-to-noise ratio is greater.

Applications of information theory to more general source variables and channels have been studied (particularly more general stochastic sources, and stochastic channels), though the models become rapidly more complex as simplifying assumptions are dropped.

Chapter 4 — Decision Trees and Hierarchical Partitioning

4.1 Adaptivity

Many Data Mining techniques are distinguished by their adaptive nature. An adaptive methodology is one which modifies its strategy according to the local nature of the data. Such adaptivity allows a model to tailor itself to the local data qualities; where data is denser or more heterogeneous, it can be modelled at a greater resolution (specificity); where data are noisier, a smoother model can be imposed; and so on.

Commonly, adaptive methods employ a '*Divide and Conquer*' strategy (or *recursive hierarchical partitioning*). The principle is to recursively subdivide the population into exclusive exhaustive subsets (partitions) in such a way that records in one partition behave by some criterion similarly, and records from different partitions behave relatively dissimilarly. The process is repeated for each partition, but the type and/or details of the algorithm employed are allowed to vary according to local criteria (that is, based on the data in the partition being processed). The decision of when to stop subdividing each partition is also based on local criteria, i.e. when the records contained in a partition are deemed sufficiently similar, or model fit sufficiently good. Accordingly, different parts of the data find themselves represented at different resolutions: where data is relatively sparse and/or uniform, fewer subdivisions occur; where data is dense and/or heterogeneous, the domain is more subdivided, and so the data more intensively processed. A consequence is that each of the final partitions in such a model display a similar degree of within-partition heterogeneity (each cluster bears a similar share of the data's total variability).

4.2 Notation

Recall the notation of section 2.2, where the domains of the predictors are denoted $\underset{\sim}{X}_1, \ldots, \underset{\sim}{X}_j, \ldots, \underset{\sim}{X}_J$, and define the ($J$-dimensional) *predictor space* (or the *domain*) to be their cross product

$$\underset{\sim}{X} = \underset{\sim}{X}_1 \times \ldots \times \underset{\sim}{X}_j \times \ldots \times \underset{\sim}{X}_J \qquad \text{(EQ 21)}$$

Also, call the product of the response domains the *response space* or *codomain*.

Every point $\underline{x}$ in the domain is specified by its J predictor values. Consider the case where all predictors are discrete[1]; then there are an enumerable number of such points in

$\underset{\sim}{X}$. We can assume all domains $\underset{\sim}{X}_j$ are finite (since we have a finite number of records). Denote a partition of $\underset{\sim}{X}$

$$\tilde{\underset{\sim}{X}} = \{\underset{\sim}{X}_{(1)}, \ldots, \underset{\sim}{X}_{(p)}, \ldots, \underset{\sim}{X}_{(P)}\} \qquad \begin{cases} \underset{\sim}{X}_p \cap \underset{\sim}{X}_q = \varnothing; (1 \le p \ne q \le P) \\ \text{and} \quad \bigcup_{p=1}^{P} \underset{\sim}{X}_{(p)} = \underset{\sim}{X} \end{cases} \qquad \text{(EQ 22)}$$

calling the *exclusive, exhaustive* subsets $\underset{\sim}{X}_{(p)}$ the *cells* of the partition $\tilde{\underset{\sim}{X}}$.

A *hierarchical partition* $\widehat{\underset{\sim}{X}}$ is a series of partitions of $\underset{\sim}{X}$, which starts with the universal partition (whose only cell is the entire domain), and in which each partition is derived from the previous partition by splitting one or more cells. Formalising,

$$\widehat{\underset{\sim}{X}} = (\tilde{\underset{\sim}{X}}_1, \ldots, \tilde{\underset{\sim}{X}}_q, \ldots, \tilde{\underset{\sim}{X}}_Q) \qquad \text{(EQ 23)}$$

is a hierarchical partition of $\underset{\sim}{X}$ if

$$\begin{aligned} &\tilde{\underset{\sim}{X}}_1 = \{\underset{\sim}{X}\} \qquad \text{and} \\ &\tilde{\underset{\sim}{X}}_q < \tilde{\underset{\sim}{X}}_{q-1}; (2 \le q \le Q) \end{aligned} \qquad \text{(EQ 24)}$$

where we use $\tilde{\underset{\sim}{X}}' < \tilde{\underset{\sim}{X}}$, to denote that $\tilde{\underset{\sim}{X}}'$ is a proper subpartition of $\tilde{\underset{\sim}{X}}$. We say $\tilde{\underset{\sim}{X}}' = \{\underset{\sim}{X}'_{(0)}, \ldots, \underset{\sim}{X}'_{(P')}\}$ is a subpartition of $\tilde{\underset{\sim}{X}} = \{\underset{\sim}{X}_{(0)}, \ldots, \underset{\sim}{X}_{(P)}\}$ if every $\underset{\sim}{X}'_{(p')} \in \tilde{\underset{\sim}{X}}'$ is a subset of some $\underset{\sim}{X}_{(p)} \in \tilde{\underset{\sim}{X}}$; the subpartition is proper if $p' > p$ (so that at least one cell in $\tilde{\underset{\sim}{X}}'$ is a proper subset of some cell in $\tilde{\underset{\sim}{X}}$).

4.3 Decision Trees

A hierarchical partition $\widehat{\underset{\sim}{X}}$ is most easily represented by a tree. The root represents the whole domain. Each level of the tree represents a partition $\tilde{\underset{\sim}{X}}_q$, with nodes at that level representing partition cells (domain subsets); the leaves represent cells of $\tilde{\underset{\sim}{X}}_Q$. We can mark the branches of the tree with conditions on attributes; these are the conditions which exactly specify the child cell as a subset of the parent cell—thus the branches descending from any one node have conditions which are mutually exclusive and exhaustive. If the tree is such that each branch condition involves just a single predictor attribute, the tree is called a *decision tree*, since any case in the database belongs to exactly one cell (node) at any given level of the tree, and the conditions on each branch

1. Since continuous variables can be *discretised* by partitioning their domains.

decide in which cell each case should reside at the next level. (FIGURE 1.) gives an incomplete example. A *binary* decision tree is one in which at most one cell is split at each level, and always into exactly two children.

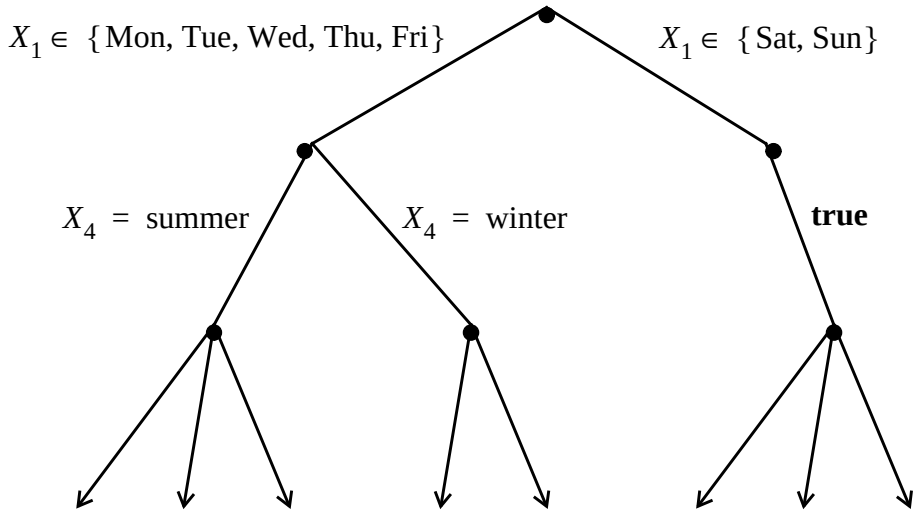


FIGURE 1.

Top-down tree building, also known as *general-to-specific* partitioning, starts with the universal partition and recursively splits cells until some (usually local) criteria limiting tree size is met. Bottom-up tree building, or *specific-to-general* partitioning, starts with the maximal possible number of leaves, each of which contains either just one case from the database, or all those cases which have identical values for every predictor. At each stage, some of the cells are combined into a larger cell so as to form a superpartition; at the final stage, all cells have been combined to form the universal partition at the root.

Top-down methods are more common though are often augmented with bottom-up methods. This is known as *overfitting and pruning* a tree: a tree is grown top-down until it is (deliberately) ‘too’ large, then pruned back by recursively combining cells, until a ‘right’ sized tree is found (this may be repeated iteratively). The pruning (cell joining) criterion must, of course, be different from the growing (cell splitting) criterion.

Decision trees are used for regression, classification and clustering. The objective of a decision tree is usually to find partitions of the predictor space whose cells are such that the response variables of the cases in a cell behave similarly; or, such that the response variables of two cases in different cells behave *dissimilarly*. Thus in a regression tree, the objective is to seek partitions whose each cell contain cases whose responses fit the

same regression model well. In a classification tree there is one response, *class*, and cells should ideally contain cases which are all in the same class with as few exceptions as is possible. In decision tree clustering, cells should contain cases whose multivariate responses are ‘close’, according to some multivariate distance measure.

Chapter 5 — Variable Reduction and Data Visualisation

5.1 Introduction

In order to visualise multidimensional data we wish to reduce its overall dimensionality to two or three to enable a graphical representation. The most common methods are factor analysis, principal components analysis, and discriminant analysis, each of which aims to replace a large variable set with a smaller variable set, the latter of which captures as much as possible of the interactional structure of the former. Of these, principal components analysis (PCA) is the simplest, and arguably the most widely useful. The new variables discovered by these methods (particularly PCA) have much utility in a number of data analysis techniques, besides their usefulness in data visualisation. In high dimensional data mining tasks in possibly noisy databases, variable reduction can prove very useful in reducing computational complexity, and improving human conceptualisation of problems and data structures. It can prove especially so when the database has a large number of strongly linearly dependent variables.

Other visualisation techniques are also important in representing complex underlying structure in databases. Of course, graphs and charts of all manner can be starting points in determining structures. Decision trees (section 4.3) are one example of an at-a-glance distillation of multidimensional data structure. Other types of tree (e.g. dendrograms in intrinsic cluster analysis, see section 8.5) can also be of use. Problem-specific visualisations may suggest themselves to the data analyst, whether they serve as exploratory tools, or as final representations of discovered knowledge.

5.2 Principal Components Analysis

Suppose certain linear combinations of the continuous variables[1] $u_1, \ldots, u_p, \ldots, u_P$ are to be introduced as replacements. Call them $v_1, \ldots, v_k, \ldots, v_K$ where

$$v_k = a_{k1}u_1 + \ldots + a_{kp}u_p \qquad 1 \le k \le K \qquad \text{(EQ 25)}$$

or in matrix form,

1. The variable set to be reduced may be a set of predictor variables or of response variables, though not usually a mixture.

$$v_k = \underline{a}_k^T \underline{u} \quad \text{(EQ 26)}$$

The *k*th *principal component* of the data $\underline{u}_i; (1 \le i \le N)$ is denoted v_k.

v_1 is defined as that linear combination which maximises the variance of the combination over the *N* observations, subject to the unity constraint,

$$\sum_{p=1}^{P} a_{1p}^2 = \underline{a}^T \underline{a} = 1 \quad \text{(EQ 27)}$$

The variance of a linear combination $\underline{a}_k^T \underline{u}$ of $\underline{u}$ is defined

$$\text{var}(\underline{a}_k) = \sum_{i=1}^{p} \sum_{j=1}^{p} a_{ki} a_{kj} \sigma_{ij} \quad \text{(EQ 28)}$$

where $\sigma_{ij} = \text{cov}(u_i, u_j)$, the covariance of predictors *i*,*j* over the observations; in matrix algebra

$$\text{var}(\underline{a}_k) = \underline{a}_k^T C \underline{a}_k \quad \text{(EQ 29)}$$

where *C* is the covariance matrix of $\underline{u}$ over the observations. Often the variables are first normalised to have unit variance; in this case *C* becomes a correlation matrix (usually denoted *R*).

The second PC v_2 is defined as that linear combination of $\underline{u}$ which maximises $\text{var}(\underline{a}_2) = \underline{a}_2^T C \underline{a}_2$ subject to the constraints

$$\begin{cases} \underline{a}_2^T \underline{a}_2 = 1 \\ \underline{a}_1^T \underline{a}_2 = 0 \end{cases} \quad \text{(EQ 30)}$$

The second constraint ensures linear independence (orthogonality) of the first two PCs. The third PC maximises $\underline{a}_3^T C \underline{a}_3$ subject to $\underline{a}_3^T \underline{a}_3 = 1$ and mutual linear independence of the first three PCs, and so on, so that any two principal components are guaranteed orthogonal.

Often $\underline{v}_1$ represents that linear combination of variables which best typifies the behaviour of $\underline{u}$ amongst the observations, and $\underline{v}_2$ can be interpreted as the combination orthogonal to $\underline{v}_1$ which best *distinguishes* the different behaviours of $\underline{u}$ amongst the observations. Further PCs often have clear interpretations, dependent on knowledge of the field of study.

The $\underline{a}_1, \ldots, \underline{a}_k$ are the first *k* eigenvectors of the covariance matrix *C*. Each eigenvector has a corresponding eigenvalue $\lambda_1, \ldots, \lambda_k$; these are proportional to the proportion of total variance in the data accounted for by the corresponding eigenvector, and $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_k$. Thus the first *k* PCs account for $\left(\sum_{k=1}^{K} \lambda_k \right) \left(\sum_{p=1}^{P} \sigma_p \right)^{-1} \times 100\%$ of the total variance where σ_p are the variances of the original variables. The eigenvector matrix $A = [a_{ij}]$ relates the PCs to the original variables,

$$\underline{v} = A^T \underline{u} \qquad \text{(EQ 31)}$$

and $A^T A = I$. We can spectrally decompose the covariance matrix as

$$R = A \Lambda A^T \qquad \text{(EQ 32)}$$

where Λ is the diagonal $p \times p$ matrix whose diagonal entries are the eigenvalues $\lambda_1, \ldots, \lambda_p$, which expands to

$$R = \sum_{p=1}^{P} \lambda_p \underline{a}_p \underline{a}_p^T \qquad \text{(EQ 33)}$$

5.3 Rotation of Principal Components

The *loadings* l_k for the *k*th PC are obtained by scaling the coefficients $\underline{a}_k$ by $\sqrt{\lambda_k}$:

$$l_k = \sqrt{\lambda_k} \cdot \underline{a}_k \qquad \text{(EQ 34)}$$

and the l_k together form the $p \times k$ *loadings matrix*, *L*. Note $R = LL^T$. Manual examination of the loadings is generally performed when trying to interpret the principal components. Now if the first *k* PCs account for a ‘significant’ proportion of total variance, we know that the original data lie ‘close’ to a *k*-plane, the plane defined by the *k* eigenvectors. If these PCs are rotated in the *k*-plane, the rotated vectors still define the plane with no loss of information; however, certain rotations of the components admit more obvious interpretations. The *varimax* rotations are the unique orthogonality-preserving rotation of the PCs which maximise the sum of variances of the loadings matrix (obtained iteratively). The varimax-rotated components tend to have loadings which are close to either 0 or 1 and hence have obvious interpretations as indicators of similarities and dissimilarities between certain variables.

5.4 Applications and extensions

We may wish to retain a subset of the original variables rather than linear combinations, but use the PCs to select a reduced variable set. One method is to include the original variable which has the greatest loading in v_1, then that with the greatest loading in v_2 (unless already included), and so on.

PCs have particular validity in multiple simple linear regression. If the original *p* predictors are replaced by their *p* PCs, the resulting simple linear regression parameters have variances inversely proportional to the variance of the corresponding PC. Thus low variance (low eigenvalue) PCs are unreliable as simple linear regression predictors, and are often omitted. Furthermore, the regression coefficient for a particular PC remains constant regardless of how many other PCs are included in the model (since PCs are mutually uncorrelated) and can thus be determined separately.

Two dimensional scatter plots for the cases' loadings for the first two PCs may be informative representations for multidimensional data (see FIGURE 2.). In particular such plots can be used for the visual identification of outliers (marked x, y) and clusters (marked a, b).

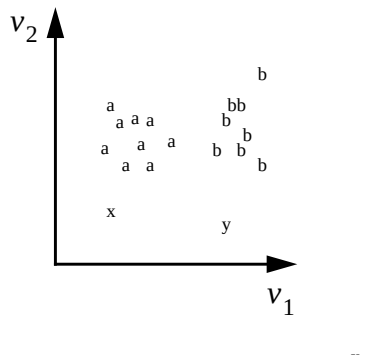


FIGURE 2.

Use of the first few PCs of the response (*pattern*) variables, rather than all the variables, can reduce the size of a cluster analysis task.

In *discriminant analysis*, which relates to classification (chapter 7) in the same way in which PCA relates to cluster analysis (chapter 8), the original cases $C_i; (1 \le i \le N)$ (variables $\underline{u}_i$) each have an associated *class* variable (discrete response Y_i). The idea is to obtain linear combinations of the predictors which have maximal discriminatory power between classes. Eigenvectors of $S_W^{-1}S_B$ provide the coefficients for the linear combinations, where S_W, S_B are (respectively) the within-group and between-group scatter matrices defined by the class variable (related to the scalar scatters of section 8.3, which

are scatter matrices summed over rows and over columns).

Factor analysis has similar aims to PCA but a more complex underlying model which relies on the notion of a set of hypothetical, unobservable *common factors*; each variable has an expression as a linear combination of *k* common factors and one unique factor. Factor analysis is popular in the social sciences, where data often contains substantial measurement error, and an underlying factor model can be postulated from theory or experience.

Correspondence analysis is a form of PCA applicable to categorical variables, which can be used to visualise the relationships between two categorical variables. Principal components are induced from the contingency table of the two variables, and the *categories* of each variable are plotted as points on a graph which has principal components as axes. Points which appear nearby on this diagram represent either similar categories of the same variable, or highly contingent categories of the different variables.

Chapter 6 — Regression in Large Noisy Databases

6.1 Formulation

It is convenient for us to suppose at this stage that the predictors $X_1, \ldots, X_j, \ldots, X_J$ are all continuous. Regression on discrete predictors will be considered shortly. Regression requires continuous responses; usually multiple responses are dealt with separately or somehow combined into one, so assume a single response Y. The general parametric regression model (with additive errors) is

$$Y_i = f(\underline{X}_i;\underline{\theta}) + \varepsilon_i \qquad 1 \le i \le N \tag{EQ 35}$$

where $\underline{X}$ is the *J*-vector of predictors, $\underline{\theta}$ is a vector of parameters $(\theta_1, \ldots, \theta_L)^T$, and ε_i are the errors in the model for each case $1 \le i \le N$. If errors are assumed multiplicative, we write

$$Y_i = f(\underline{X}_i;\underline{\theta}) \cdot \varepsilon_i^* \tag{EQ 36}$$

which can be transformed to the additive error model by taking logs:

$$\log Y = \log f(\underline{X}_i;\underline{\theta}) + \varepsilon_i \tag{EQ 37}$$

where $\varepsilon_i^* = \exp\varepsilon_i$. The additive errors (or additive log errors in (EQ 37)) are assumed independent and identically distributed (*i.d.d*), and generally assumed Gaussian.

The parametric model is linear when the regression function *f* can be written in a form

$$f(\underline{X};\underline{\theta}) = \alpha_1 f_1(X_1) + \ldots + \alpha_J f_J(X_J) \tag{EQ 38}$$

(whether or not the f_j are linear). Thus fourier regression and polynomial regression are linear.

Visual examination of scatter plots of the errors will usually be enough to determine whether additive or multiplicative error assumptions are more appropriate. If the distribution of errors in both cases is too far from Gaussian, a more complex transformation of the data may be considered.

The *regression equation* is the criterion for parameter selection. Most common is the *least-square-error criterion*:

$$\min S(\underline{\theta}) = \sum_{i=1}^{N} w_i [Y_i - f(\underline{X}_i;\underline{\theta})]^2 \quad \text{(EQ 39)}$$

Weights w_i may be absent, or selected according to various criteria. Least-square-error minimisation is particularly sensitive to outliers, which may distort the final regression function. Outliers can be removed before regression modelling, or an error measure less punitive to outliers may be adopted.

Non-linear parametric regression models (EQ 35) have many associated problems [6]. Firstly it is difficult to select a form for the regression function unless there is a sound domain-dependent precedent for choosing one. Secondly, different parametrisations are possible for each candidate function, some of which may lead to poorly conditioned equations. Thirdly, the regression equation is usually insoluble except by iterative methods, often with poor convergence rates. On the other hand, linear multivariate parametric models (which have easily soluble regression equations) can rarely be found which fit the data well in all parts of the predictor space. Since data mining tasks often have high dimensional domains, and high-noise response variables which do not vary smoothly, non-classical regression techniques which have greater flexibility are often preferable.

6.2 Stepwise Regression

It is supposed that the response $Y_{(0)}$ has a univariate, possibly weak, relationship with each of the predictors X_j individually. Each univariate regression model is usually linear with few parameters unless there is a particular reason to adopt a non-linear model. Different models may be used for each predictor.

The basic idea behind stepwise regression is that each predictor is used in turn to model the response. Having found the regression function for the first predictor, the actual values of the response $Y_{(0)}$ are differenced with the values $\hat{Y}_{(0)}$ predicted by the regression function, to create a new response variable $Y_{(1)}$. This can be thought of as the original response ‘filtered for’ the effect of the first predictor. Next a new predictor is selected and used to model the residual response $Y_{(1)}$. This continues until no more significant relationships can be found.

The order in which predictors are used may be decided by heuristics; a simple linear correlation coefficient can be computed for each predictor with the response and the most highly correlated predictor used; or the regression function can be determined for

each predictor, and the predictor resulting in the closest fit selected. Predictors may be selected more than once.

It is simple to reconstruct a single equation for $Y_{(0)}$ in terms of the predictors by chaining backwards, but there is no guarantee that the reconstructed model will be globally least-square-error.

6.3 Hierarchical Regression

In a hierarchical regression, an initial regression is used as a means of variable reduction. An initial regression model

$$Y_i = f(\underline{X}_i;\underline{\theta}) + \varepsilon_i \qquad 1 \le i \le N \qquad \text{(EQ 40)}$$

is postulated, with the number of parameters in $\underline{\theta}$ significantly smaller than the number of responses in $\underline{X}$. In general, not all of the predictors $X_1, \ldots, X_J$ will be used; and in general, not all of the cases $1 \le i \le N$ will be used at once—sometimes a separate fit is determined for each case, or for each cell in a partition of the cases.

In the second phase, the discovered parameters $\underline{\theta} = (\theta_1, \ldots, \theta_L)$ are now treated as new response variables, and each in turn is regression modelled as a function of the $X_1, \ldots, X_J$, usually only using those predictors which were not used in the initial stage. The process is not normally extended beyond the second tier of regressions. Via back-substitution, a single regression function for Y in terms of $\underline{X}$ can be recovered, although as with stepwise regression the errors are not least-square globally.

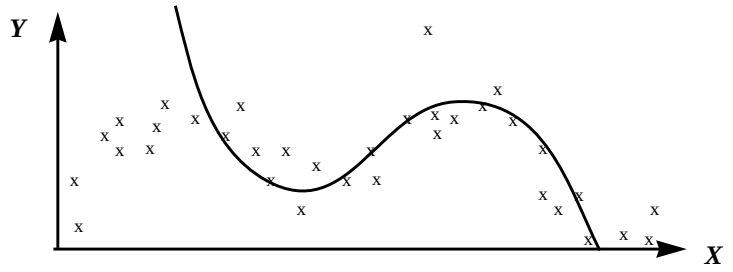
Example. Suppose we wish to model a database of analogue communications signals. As a first step, we might decompose each signal into a linear combination of a handful of preselected sinusoids, using linear regression; here the only predictor is time. Next, any remaining predictors can be used to regression model the coefficients for each sinusoid.

6.4 Piecewise Regression and Non-Parametric Regression

6.4.1 Regression with Piecewise Polynomials

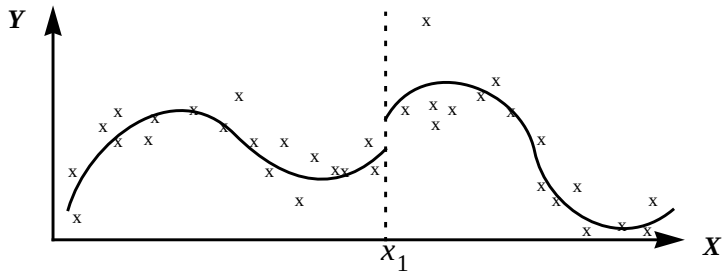
Consider fitting a polynomial regression model to the data in (FIGURE 3.). Simple linear or quadratic curves do not fit the data well; even a cubic or a quartic fares badly,

FIGURE 3. [1]



and the danger of fitting higher and higher order polynomials is overfitting the data—high order polynomial regressions tend to fit the noise rather than smooth the data. Extending to a multivariate case, where we would like to fit a surface or hypersurface to noisy data with many local minima and maxima, any attempt to fit the data with a single global regression function is almost certainly doomed, however complicated the form of the function. In (FIGURE 4.), the same data have been fitted with two cubic equations, each of which is least-square-error for a subdomain of *X*.

FIGURE 4.



Not only is the fit much better than the single cubic fit in (FIGURE 3.), but the possibility of fitting the noise rather than the trend is less likely than with a single high-order polynomial. This is an example of piecewise polynomial regression.

6.4.2 Splines

Note the discontinuity at x_1 in (FIGURE 4.) It is highly unlikely that the true nature

1. (FIGURE 3.) and (FIGURE 4.) depict hand drawn approximations to least-square-error cubic fits only.

of the data is discontinuous at this point, or that the gradient should be discontinuous at this point. If we fit the two cubics again, with the additional constraints that the compound curve is both *continuous* and *once differentiable* at x_1, we obtain a (cubic) *regression spline*. A spline is essentially a piecewise polynomial fit to data with additional constrains at the junctions between constituent curves. These junctions (such as x_1 in (FIGURE 4.)) are known as knots (since the constituent curves are 'tied together' by continuity constraints at these points). An *interpolating* cubic spline for the above data would pass through every data point, be continuous with continuous first derivative everywhere, and twice differentiable everywhere excepting the knots—every data point is a knot for an interpolating spline. Clearly this is unsuitable for noisy data and what is required is a *smoothing* spline. A smoothing spline may have a knot at every data point. We do not wish to interpolate every (or even any) point, so the regression equation consists of restrictions on the least-square-error of the fit, *and* of continuity constraints.

Smoothing splines are an example of non-parametric regression—there is no preordained fitting function or number of parameters; any 'good' description of the data will do. Our univariate non-parametric model is

$$Y_i = f(X_i) + \varepsilon_i \qquad \text{(EQ 41)}$$

We do not impose a preset form for f, but instead insist on certain constraints. If we insist on continuity for, and continuity of first derivative for, f; continuous second and third derivatives except at knots; and piecewise constant third derivative everywhere; then f is a cubic spline. Another way of phrasing these constraints is that the second derivative is continuous and piecewise linear. If $S(X)$ is a cubic spline on the set of knots $\underset{\sim}{K} = \{k_1, \ldots, k_{|\underset{\sim}{K}|}\}$, then equivalently it can be written in the form

$$S(X) = \alpha_1 X^3 + \alpha_2 X^2 + \alpha_3 X + \alpha_4 + \sum_{i=2}^{|\underset{\sim}{K}|-1} a_i [X - k_i]_+^3 \qquad \text{(EQ 42)}$$

The first four terms form a cubic polynomial and the last term is a sum of *kernel* functions centred at (internal) knots. The kernel functions are translates of

$$\phi(x) = [x]_+^3 \equiv \begin{cases} x^3 & x \geq 0 \\ 0 & x < 0 \end{cases} \qquad \text{(EQ 43)}$$

There is an alternative and (perhaps surprisingly) equivalent formulation for cubic splines: define

$$J(f) = \int_{k_1}^{k_{|\underset{\sim}{K}|}} [f''(X)]^2 \mathrm{d}X \qquad \text{(EQ 44)}$$

and $J_2[k_1, k_{|\underset{\sim}{K}|}]$ the set of all functions for which the integral in (EQ 44) exists. Then the function in $J_2[k_1, k_{|\underset{\sim}{K}|}]$ which interpolates data at the knots and minimises (EQ 44) is unique, and is a *natural interpolating cubic spline* (a natural cubic spline is one which has zero second derivatives at the external knots $k_1, k_{|\underset{\sim}{K}|}$). The knot set is usually the entire data set. When smoothing rather than interpolation is required, we drop the interpolation constraint, but by adding a least-square-error component to (EQ 44), we obtain the formulation for the *natural cubic smoothing spline*:

$$\min E_{\underset{\sim}{\lambda}}(f) = J(f) + \sum_{i=1}^{N} \lambda_i \cdot (f(X_{(i)}) - f_i)^2 \qquad \text{(EQ 45)}$$

Here the $X_{(i)}$ are the predictor values for the N observations and f_i the N observed responses. The parameters of (EQ 45) are the weights λ_i, and the knot set $\underset{\sim}{K}$ and are known as *smoothing parameters*[1]. The knot set is usually the entire data set. The sparser the knots the smoother the spline $f(X)$ (and the less closely it fits the data); the smaller the weights λ_i, the smoother the spline also. Careful choice of smoothing parameters is vital in spline smoothing, since too much smoothing produces near-linear splines with poor fit, but too little smoothing tends to overfit the data—i.e. to fit the noise, not the trend. $J(f)$ itself can be seen as a measure of smoothness for a function f.

One common simplification is to take $\lambda_i = \lambda / \sigma_i^2$ where λ is a common smoothing parameter and σ_i^2 some estimate of the variance of f_i.

6.4.3 Multivariate Smoothing Splines

Now extend the non-parametric model to a multivariate case:

$$Y_i = f(X_1, \ldots, X_j, \ldots, X_J) + \varepsilon_i \qquad \text{(EQ 46)}$$

The case where $J = 2$ is rather common, higher dimensionality less so. When $J = 2$ the

1. An irony of the term ‘non-parametric regression modelling’ is that such models usually have more parameters (albeit of the smoothing variety) than comparable parametric models.

problem is to fit a smoothing *surface* to the data. Recall the two derivations of univariate smoothing splines. We can consider them as piecewise polynomial regression curves tied by continuity constraints at the knots; or as minimisers of a compound least-square-error/smoothness criterion. These extend to two different concept of multivariate smoothing splines, respectively *finite element methods* and *(hyper-)surface splines*. A third concept of multivariate smoothing splines is provided by the *blending function methods* and *tensor product* splines. These methods derive univariate splines for each predictor X_j and blend them into a surface or hypersurface.

Finite element methods require reliable estimates for the partial derivatives at knots, and will not concern us.

Surface splines for smoothing are analogues of univariate smoothing splines for the bivariate case. Recall (EQ 42) which decomposes a univariate cubic spline into a univariate cubic and a sum of kernel functions centred at knots. Note that the kernel functions are translates of a basic function $\phi(x)$. For surface splines we seek a basic function $\phi(x_1, x_2)$ which has rotational symmetry about the origin. Thus ϕ can be expressed as a function of $\rho = \sqrt{x_1^2 + x_2^2}$. Common choices are $\phi(\rho) = |\rho^3|$ (this is the direct analogue of cubic splines); $\phi(\rho) = |\rho^2| \ln|\rho|$, which is used in the *thin plate spline*; and $e^{-\alpha|\rho|^2}$, the *rotated Gaussian spline*. The various basic functions optimise various different smoothness measures $J(f)$. These ideas have natural extensions to higher dimensional splines.

6.4.4 Kernel Estimators

Surface and hypersurface splines are examples of a broader class of non-parametric models, *multivariate kernel estimators*. These methods are based around the summation of kernel functions centred at various points, but do not necessarily obey any smoothness criteria, like splines.

An example is *near neighbour regression*, a family of kernel estimation methods which estimate the value $f(\underline{X})$ as some function of the response values taken by points in the sample data with predictor values ‘near’ to $\underline{X}$; for example, take the mean of the response values for the k nearest neighbours to $\underline{X}$ in the sample data.

6.5 Multivariate Adaptive Regression Splines and Related Models

The Multivariate Adaptive Regression Spline (MARS) [9], [10] is an ingenious non-

parametric regression technique which combines ideas from regression tree models (recursively fitting separate models to cells of a hierarchical partition) and from the method of interaction splines. It is adaptive in the sense that its strategy adjusts according to the local behaviour of the function to be approximated. It was designed for data sets with 50 to 1000 observations and 3 to 20 predictors though has been fruitfully applied to larger problems [9].

6.5.1 Interaction Splines

The interaction spline is designed to model multidimensional data as a linear combination of low dimensional splines. The model is

$$\hat{f}(\underline{X}) = \sum_{j=1}^{J} \hat{g}_j(\underline{Z}_j) \qquad \text{(EQ 47)}$$

where $\underline{Z}_j$ are low (usually one or two) dimensional subsets of arguments from $\underline{X}$, the set of predictors, and $\hat{g}_j$ are splines of appropriate dimension. The model is appropriate when interactions between predictors are thought to be of low dimension. The criteria for optimising the $\hat{g}_j$ is a global version of (EQ 45),

$$\min \sum_{i=1}^{N} \left[y_i - \sum_{j=1}^{J} \hat{g}_j(\underline{Z}_{ij}) \right]^2 + \sum_{j=1}^{J} \lambda_j J(\hat{g}_j) \qquad \text{(EQ 48)}$$

Choice of the predictor subsets $\underline{Z}_j$ is a crucial factor and requires prior knowledge of which predictors interact and at what level.

6.5.2 Recursive Partitioning Recast

The basic regression tree model is

$$\underline{X} \in \underset{\sim}{X}_{(p)} \Rightarrow \hat{f}(\underline{X}) = \hat{g}_p(\underline{X};\underline{\theta}) \qquad 1 \le p \le P \qquad \text{(EQ 49)}$$

where $\{\underset{\sim}{X}_{(1)}, \ldots, \underset{\sim}{X}_{(P)}\}$ partition predictor space $\underset{\sim}{X}$. The partition and the parameters are estimated simultaneously using recursive hierarchical partitioning (divide and conquer strategy). A simple but popular choice for the $\hat{g}_p$ is constant functions. We can recast the regression function for the regression tree with piecewise constant fitting function into a single regression function

$$\hat{f}(\underline{X}) = \sum_{p=1}^{P} a_p B_p(\underline{X}) \quad \text{(EQ 50)}$$

where

$$B_p(\underline{X}) = I[\underline{X} \in \underset{\sim}{X}_{(p)}] \equiv \begin{cases} 1 & \text{if } \underline{X} \in \underset{\sim}{X}_{(p)} \\ 0 & \text{otherwise} \end{cases} \quad \text{(EQ 51)}$$

If $H[\nu]$ is a step function $H[\nu] = \begin{cases} 1 & \nu \geq 0 \\ 0 & \nu < 0 \end{cases}$ then the *basis functions* B_p can be expressed

$$B_p(\underline{X}) = \prod_{k=1}^{K_p} H[S_{kp}(X_{j(k,p)} - t_{kp})] \quad \text{(EQ 52)}$$

Here the t_{kp} are the K_p split points in the *p*th recursive partition, called for convenience knots. Knot t_{kp} splits the $j(k, p)$-th predictor; $S_{kp} = \pm 1$ carries the left/right information about that split. Predictors may be selected for splitting several times.

6.5.3 Piecewise Linear MARS Model

(EQ 52) produces a discontinuous fitting function; the basic idea behind the MARS model rids us of the discontinuities by replacing *H*, the step function, by a truncated power function

$$b_q^{\pm}(x - t) = [\pm(x - t)]_+^q \quad \text{(EQ 53)}$$

where the right hand side is defined as $[\pm(x-t)]^q$ if $\pm(x-t) \geq 0$, or 0 otherwise. This is the *two-sided truncated power basis* (compare (EQ 43)); in fact $q = 1$ is used, so that the basis functions are piecewise linear.

The multivariate spline basis functions are

$$B_p(\underline{X}) = \prod_{k=1}^{K_p} S_{kp}[X_{j(k,p)} - t_{kp}]_+^1 \quad \text{(EQ 54)}$$

along with one constant term. Knots are always located at the projection of a data point onto the axis of the relevant predictor. A problem arising with this model is that when some attribute is selected for splitting more than once, corresponding basis functions have power greater than one.

However MARS adds a further generalisation to the model which rids us of this problem, and also allows us to use basis functions of arbitrarily low dimension at any stage in the recursion. The idea is to generalise the set of basis functions (EQ 54) to include interactions of every dimension $\leq K_p$. Lower order interaction terms are always available for selection or reselection at any stage. The same attribute can be selected many times without incurring high order basis functions: each basis function added can only feature that attribute once in its product. The process begins with only the constant basis function $B_0(\underline{X}) = 1$, and after the $m+1$ st iteration ($m \geq 0$) adds two new basis functions

$$\begin{aligned} B_{2m+1}(\underline{X}) &= B_{l(m+1)}(\underline{X}) \cdot [+(X_{v(m+1)} - t_{m+1})]_+^q \\ B_{2m+2}(\underline{X}) &= B_{l(m+1)}(\underline{X}) \cdot [-(X_{v(m+1)} - t_{m+1})]_+^q \end{aligned}, \qquad \text{(EQ 55)}$$

where $B_{l(m+1)}(\underline{X})$ is one of the basis functions already selected, $v(m+1)$ is one of the variables *not already present in* $B_{l(m+1)}(\underline{X})$, and t_{m+1} is the knot location on that variable. These three parameters are chosen so as to most improve goodness of fit of the new model

$$\begin{aligned} \left(\sum_{i=0}^{2m} a_i B_m(\underline{X}) \right) &+ a_{2m+1} B_l(\underline{X}) [+(X_{v(m+1)} - t_{m+1})] \\ &+ a_{2m+2} B_l(\underline{X}) [-(X_{v(m+1)} - t_{m+1})] \end{aligned} \qquad \text{(EQ 56)}$$

(optimisation is with respect to the parameters $l, v, t, a_0, a_1, \ldots, a_{2m+2}$). It is not feasible (for problems of any reasonable size) to compute exactly optimal values for all these parameters, and a number of sacrifices of optimality for speed are made by the MARS procedure (see [11]) to achieve more reasonable computation times.

Note that the resulting model is *not* a partition (it is exhaustive but not exclusive—some regions of predictor space are relevant to several basis functions). This is best illustrated by the *ANOVA decomposition*[1] of the MARS model:

$$\hat{f}(\underline{X}) = a_0 + \sum_{K_p=1} f_i(X_i) + \sum_{K_p=2} f_{ij}(X_i, X_j) + \sum_{K_p=3} f_{ijk}(X_i, X_j, X_k) + \ldots \qquad \text{(EQ 57)}$$

The $f_i, f_{ij}, f_{ijk}, \ldots$ are sums of basis functions involving exactly $1, 2, 3, \ldots$ predictors.

1. So called because the decomposition looks like that used in Analysis of Variance (ANOVA)

For example,

$$f_1(X_1) = \sum_{m=1}^{M} a_m B_m(X_1) \quad \text{(EQ 58)}$$

would be a piecewise linear spline in X_1 formed from *M* univariate basis functions of the form of (EQ 53).

The lack-of-fit function used when deciding which basis function to add next is a modified form of the generalised cross-validation score, GCV:

$$E(\hat{f}_p) = \frac{\frac{1}{N}\sum_{i=1}^{N} [Y_i - f_p(\underline{X}_i)]^2}{\left[1 - \frac{C(P)}{N}\right]^2} \quad \text{(EQ 59)}$$

Normally, *model complexity* $C(P)$ is just equal to P, the number of parameters being fitted. The denominator is a term which is intended to counter the bias introduced by the increased variance due to greater model complexity. GCV is comparable at any level in the hierarchy (see [9]). In MARS it is modified by choosing

$$C(P) = P \cdot \left(\frac{d}{2} + 1\right) + 1 \quad \text{(EQ 60)}$$

($2 < d < 4$ usually), where P is the number of non-constant basis functions being considered [10]. Large *d* penalises knot addition more severely, and results in fewer knots, hence smoother fitting functions.

The MARS methodology opts for an overfitting and pruning approach to determine a right-sized model. The user supplies a maximum number of basis functions to be discovered in the forward selection algorithm; the backward pruning algorithm consequently removes some (typically about half) of them from the model. Each iteration seeks the basis function whose removal either least degrades or most improves fit. The constant basis function $B_1(\underline{X}) = 1$ cannot be removed (so there is never a ‘hole’ in the domain in which no function is applicable).

The overfitting and pruning approach allows low order interaction terms to be selected in the stepwise-forward part of the algorithm that will eventually be removed, but which nevertheless serve as building blocks on the way to building higher order terms.

6.5.4 Piecewise Cubic MARS Model

The piecewise linear MARS model with $q = 1$ does not have a continuous first derivative, and tends to have a poor fit near to knots. Truncated cubics of a special form can be used in place of the two-sided truncated *q*=1 power basis. The positive versions have the form

$$C(x;t_-, t, t_+) = \begin{cases} 0 & x \le t_- \\ p_+ (x - t_-)^2 + r_+ (x - t_-)^3 & t_- < x < t_+ \\ x - t & x \ge t_+ \end{cases} \quad \text{(EQ 61)}$$

with analogous negative forms. p_+ and p_- are functions of $t, t_-,$ and t_+ chosen so that the functions are continuous, with continuous first derivatives (second derivatives being discontinuous only at t_+, t_-). This results in a final fitting function which is continuous, with continuous partial derivatives. The positive form of the truncated cubic is illustrated in (FIGURE 5.).

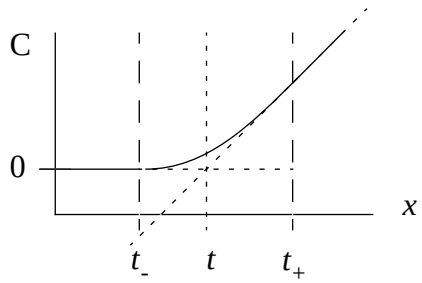


FIGURE 5.

Central knots (*t*) are placed at data points; side knots (t_-, t_+) are placed at the midpoints between central knots. An additional smoothing parameter limits the closeness of consecutive knots of any one predictor: there are at least parameter *L* data points between any two central knots. *L*=5 is typical. Note that the MARS model lack-of-fit function (EQ 59) is always calculated according to a piecewise linear MARS model; the piecewise cubic basis functions are used to replace the piecewise linear basis functions only once the model has been built.

MARS has been empirically shown not to claim a low error when presented with totally random data, suggesting that MARS does not overfit (fit to noise). Also, when asked to model data whose true underlying function has no interaction terms, research has shown that MARS seldom produces terms with interactions.

6.5.5 Extensions and Remarks

Semi-parametric models (regression models including some parametric terms suggested by experience or theory, in addition to a non-parametric model) can be easily incorporated into the MARS model; the term in question can simply be calculated for each datum, and included as an extra predictor variable. A priori restrictions on the maximal dimension of interaction terms, as well as on the total number of terms, can be imposed. Variables known or expected to have little or no mutual interaction can be prohibited from appearing in the same interaction term. Model parsimony can be improved by specifying a penalty for introducing a new variable to the model, so that those variables which have already been used for splitting are more likely to be picked again.

The total square error arising when using the piecewise cubic technique is generally found to be a little higher than that arising when using the piecewise linear technique [9]; however, the piecewise cubic fitting functions usually entail lower square error than the piecewise linear functions when the fitted models are tested against new data.

MARS uses a lack-of-fit criterion based on least-square-error, and is thus not locally robust (i.e. extreme outlying records may distort the model locally). Friedman [9] recommends considering the removal of outliers before applying MARS. However, MARS models are globally robust (altering an observation only has a significant effect on the model near that observation).

6.6 Regression with Discrete Predictors; Mixed MARS Model

6.6.1 General Regression with Discrete Predictors

Theoretically there is no bar to using discrete variables as predictors in regression equations. Ordered discrete attributes can be transformed to integer attributes which are treated exactly as if they were real variables. Categorical variables taking V distinct values can be transformed into V-1 binary attributes (taking values 0 or 1). For example a variable A taking categorical values {cat, dog, mouse, rat} can be transformed to the three variables C, D, M, which take the value 1 if and only if A takes the values cat, dog, mouse respectively. When A = rat, $C = D = M = 0$. Such binary values can also be treated exactly as if they were real variables. The principal problem with this approach is that very high dimensional predictor spaces are likely to arise, and consequently very sparse data.

An alternative means of dealing with discrete predictors involves partitioning the domain of the discrete predictors, and deriving separate regressions (using just the continuous predictors) in each cell of the partition. The aim is to seek partitions for the discrete predictors for which the resulting regressions have the best fits. If the partition is hierarchical, a regression tree results (see section 4.2). The principal drawback of this approach is again data sparsity: each regression fit has only a fraction of the original cases with which to construct a model.

6.6.2 MARS Model for Mixed Discrete & Continuous Predictors

An extension to the MARS model incorporates the use of categorical variables within the same hierarchical partitioning procedure used for the continuous variables [10]. The idea is that basis functions based on simple indicator functions can be defined on categorical variables, and these are allowed to compete on entirely equal terms with the continuous basis functions of (EQ 53), (EQ 54). Let X be a categorical variable with domain $\{x_1, \ldots, x_k\}$, and let A_l be non-empty subsets of categories, $A_l \subset \{x_1, \ldots, x_k\}$, $1 \le l \le L$. Then define a fitting function

$$\hat{f}(X) = \sum_{l=1}^{L} a_l I(X, A_l) \qquad L \le k \tag{EQ 62}$$

where *indicator function* $I(X, A_l)$ is defined to be 1 if $X \in A_l$, or 0 otherwise. Coefficients a_l are determined by least squares. For a given L the goal is to choose that set of subsets for which (under least squares optimisation of the coefficients a_l) the best fit is obtained. L is a smoothing parameter, with lower L producing smoother models (smoother in the sense that they tend vary less as X changes).

For a multivariate model with n categorical variables, multivariate indicator basis functions I can be formed by taking the tensor product over all of the variables of the univariate basis indicator functions, exactly as tensor products of spline basis functions are used in a continuous multivariate MARS model. Thus an n variate indicator basis function takes the form

$$I(X_1, \ldots, X_n) = \prod_{j=1}^{n} I(X_j \in A_{lj}) \tag{EQ 63}$$

where A_{lj} is some subset of values of X_j for $1 \le j \le n$. When there are n categorical variables and m continuous variables, the model simply allows tensor products between

discrete indicator functions and continuous spline basis functions. When a variable is being considered for addition into the model, if it is a continuous variable, tensor products of the existing functions with various spline basis functions featuring that variable are computed; if it is a categorical variable, products of existing functions with various indicator basis functions featuring that variable are computed.

The ANOVA terms for a mixed continuous/categorical MARS model are similar to those of the continuous model, but with an extra (optional) product of indicator basis functions introduced into each of the sums in (EQ 57). Plotting of ANOVA terms with 1 or two variables (which can be generated from higher order terms by slicing) requires that a separate plot may need to be generated for each subset A_{lj} occurring in the term (which can become unmanageable where many categorical variables are present).

This scheme has a distinct advantage over the general schemes of 6.6.1 for using categorical predictors in regression models. There is no need to present more than one new variable to the model for each categorical predictor; yet there is no need to partition the entire domain into separate subdomains for every possible combination of categorical variable values; instead, the domain for a *particular* multivariate basis function becomes split into two subdomains when and only when an indicator basis function of a categorical variable is selected to be multiplied with that multivariate basis function. Thus only at a *local* level, where deemed appropriate, is the domain split according to the values of a categorical variable, which helps to ensure that data sparsity does not become too much of a problem.

Chapter 7 — Classification Problems

7.1 Task Definition

Classification is also known as pattern recognition, discrimination, or supervised learning. In a classification task there is one response attribute, which is discrete. The response is seldom ordered—it is usually categorical, and may have some hierarchical structure. The response variable is known as the *class variable*, and its values as the *classes*. If there are several discrete variables several separate classifications are usually required. There may be any number of discrete and/or continuous predictors.

The objective of a classification task is to derive a rule or set of rules which determine, given a set of cases (the *test data*) with known predictor values but of unknown classes, which class each case does, or is most likely to, belong to. These rules must be determined from another set of cases (the *training data*), whose predictor values *and* class values are known. There are many, many data mining tasks which are classification tasks, such as diagnosis of illness from databases of medical records, fault diagnosis from engineering data, and classification of customers in sales databases.

As well as long established classical statistical methods for classification, there is a plethora of more recent approaches, including the use of decision trees and rule induction; density estimation; and artificial neural networks.

The best known statistical classifier is the method of discriminants, which works only for binary class variables (general discrete variables can be converted to binary variables; see 6.6). They aim to find the (rather simple) surface or hypersurface in predictor space which best divides the cases into the two classes. Linear and quadratic discrimination surfaces are used most frequently.

Artificial neural network approaches to classification are somewhat fashionable, but suffer from very slow training (i.e. model calculation) times. Moreover, a trained neural network is a ‘black box’—given an input, a neural network always returns an output, but it is virtually impossible to determine any reasoning behind the output. Thus they tend to add very little to the understanding of the modelling task.

Density estimation methods employ non-parametric (distribution-free) statistical models to estimate the multivariate probability distribution functions for each class value. At any point in predictor space, the class value which has the highest probability density estimate is predicted as the true class value corresponding to that point in predictor

space. *k*-nearest neighbour (*k*NN) classification is a very simple (but sometimes effective) example; the relative proportions for the observed class values amongst the *k* cases 'closest' in predictor space are used as the estimates for the probability densities. Euclidean distance is often used as the proximity measure.

There are many decision tree classifiers available, each with their own strengths and weaknesses. With each, the principal motive is to seek hierarchical partitions in predictor space within whose cells as many cases as possible belong to the same class or classes. Information theoretic measures, most commonly the information gain from branching on the values of a predictor, are often used in deriving the partition.

ID3 [12] is an early decision tree classifier which can be seen as the forbearer of many decision tree classifiers. We briefly describe its methodology. Suppose an attribute X_j with values $x_{j1}, \ldots, x_{j|\underset{\sim}{X}_j|}$ is used to partition the set of cases $C_1, \ldots, C_i, \ldots, C_N$ into exhaustive exclusive subsets $\{C_{(1)}, \ldots, C_{(|\underset{\sim}{X}_j|)}\}$ so that $C_i \in C_{(k)} \Leftrightarrow C_{ij} = x_{jk}$, where $C_{ij} \equiv X_{ji}$ is the value of the *i*th case in the *j*th variable. Let there be two classes, denoted ***P*** (positive cases) and ***N*** (negative cases) and define *class entropy* over any subset of cases C^* as

$$H_C(C^*) = -\frac{p_{C^*}}{p_{C^*}+n_{C^*}}\log_2\frac{p_{C^*}}{p_{C^*}+n_{C^*}} - \frac{n_{C^*}}{p_{C^*}+n_{C^*}}\log_2\frac{n_{C^*}}{p_{C^*}+n_{C^*}} \qquad \text{(EQ 64)}$$

where p_{C^*} is the number of positive-class cases in C^*, and n_{C^*} the number of negative-class cases (the usual entropy formulation, restricted to a binary variable). Define the *entropy of partitioning on* X_j to be

$$H(X_j) = \sum_{k=1}^{|\underset{\sim}{X}_j|} \frac{|C_{(k)}|}{|C|} \cdot H_C(C_{(k)}) \qquad \text{(EQ 65)}$$

Then the information gain from partitioning on X_j is

$$\text{gain}(X_j) = H_C(C) - H(X_j) \qquad \text{(EQ 66)}$$

which is maximised by minimising $H(X_j)$ over the candidate variables $X_1, \ldots, X_J$. Note that $H(X_j)$ is the mutual (or trans-)information between X_j and the class variable.

ID3 uses a χ^2 dependence test to determine (locally) when to stop partitioning. ID3 is a very simple decision tree classifier; descendents of ID3 have been adapted to non-binary class variables, and to deal with missing data, continuous predictors, structured

categorical predictors, and so on. Variant methods differ significantly in tree building direction, their splitting/joining criteria, the criteria for ceasing splitting/joining, and the amount and type of tree pruning.

Rule sets governing in which class a test datum should reside can trivially be induced from a decision tree. A few approaches to classification seek a rule set without using decision trees. For example, ITrule (see [3]) seeks simple inference rules in multivariate data using a beam-search.

Chapter 8 — Cluster Analysis

8.1 Task Definition

As noted, classification is also known as supervised learning. Because the training data are of known class, the task of predicting class for the test data is supervised by what the training data have told us about the nature of the class variable. Suppose now that there is no class variable, but that we wish to group the cases into (usually exhaustive and exclusive) subsets, within which cases have similar attribute values. In effect we are aiming to *discover* a meaningful class variable (the class being the index of the subset to which a case belongs); this is sometimes called unsupervised learning, and also known as cluster analysis or clustering.

We can divide clustering problems into *intrinsic* clustering and *extrinsic* clustering [14]. In intrinsic clustering, all variables can be regarded as both predictors *and* responses; cases are known as patterns. Our brief is simply to find clusters of patterns which are similar—or alternatively to find clusters of patterns such that patterns from different clusters are *dissimilar*.

In extrinsic clustering, there are distinct predictors and responses; we use pattern to mean the values taken by the *response* variables of a case. Our aim is to find clusters of similar patterns as before; but the rules governing which cluster a case belongs to are dependent only on the values of the non-pattern (*predictor*) variables of that case (i.e. must be extrinsic to the patterns).[1]

In either type of clustering, it is possible (but not always desirable) to reduce the pattern data to an $N \times N$ *proximity matrix* between N patterns, so that only the distances between patterns, rather than their attribute values, are used to determine the clusterings.

Clustering methods can be further dichotomised into hierarchical methods and non-hierarchical methods. In a hierarchical method a hierarchical partition is discovered (which can be formed top-down by splitting or bottom-up, by joining; see section 4.3). In a non-hierarchical method, a single partition is sought (rather than a tree-structurable hierarchy of partitions). This may require an a priori and somewhat arbitrary choice for the number of cells (clusters) in the model.

Sections 8.2 and 8.3 deal with metrics and notation for cluster analysis. Sections

1. Intermediate methodologies may use predictor variables to guide *but not dictate* cluster membership.

8.4 through 8.7 describe various intrinsic clustering algorithms, and section 8.8 describes ways in which extrinsic variables are used to determine clusters.

8.2 Distance Measures

Given any two patterns, we require a *distance* or *dissimilarity* measure between them. If the response (pattern) attributes are all continuous, Euclidean measures are most common; *simple Euclidean distance* is

$$\mathrm{d}(C_i, C_j) = \mathrm{d}_{ij} = \sqrt{\sum_{k=1}^{K} (Y_{ki} - Y_{kj})^2} \qquad \text{(EQ 67)}$$

and *weighted Euclidean distance* is

$$\mathrm{d}^{\underline{w}}(C_i, C_j) = \mathrm{d}_{ij}^{\underline{w}} = \sqrt{\sum_{k=1}^{K} w_k (Y_{ki} - Y_{kj})^2} \qquad \text{(EQ 68)}$$

where $\underline{w} = (w_1, \ldots, w_k)^T$ is a vector of weights for the responses. The weights may be chosen subjectively according to an experts assessment of the relative importance of the responses; or they can be set to be inversely proportional to the sample variance of the *k*th response, though this weighting presents a problem in that it tends to maximise within-cluster variance, contrary to the objectives of clustering.

Natural extensions of Euclidean distance to discrete responses can be applied where patterns consist partly or entirely of discrete variables. There are also natural Euclidean measures for the distance between two continuous *variables*, given all the cases. The natural distance measure between discrete variables is their mutual information (section 3.3).

8.3 Notation

Recall the notation for partitions and hierarchical partitions of predictor space $\underset{\sim}{X}$ (section 4.2). An intrinsic clustering is a partition on the set of all patterns, *C*. The notation of section 4.2 can be analogously applied to partitions and hierarchical partitions of the set of all cases (replace all instances of $\underset{\sim}{X}$ with C). An extrinsic clustering can be thought of as a partition on predictor space or a partition on the set of cases/patterns, and the notations are interchangeable.

Call a cell of a clustering $\tilde{C}$ a *cluster*. If $\tilde{C} = \{C_{(1)}, \ldots, C_{(p)}, \ldots, C_{(P)}\}$, we define the *P centroids* (or cluster centroids) of $\tilde{C}$ to be new pseudo-patterns

$\bar{C}_{(1)}, \ldots, \bar{C}_{(p)}, \ldots, \bar{C}_{(P)}$, each of which is somehow representative of the patterns contained in $C_{(1)}, \ldots, C_{(p)}, \ldots, C_{(P)}$ respectively. If the pattern attributes are continuous, means are taken; if pattern attributes are discrete, modes or medians may be appropriate.

In the continuous case, define for $1 \le p \le P$

$$\bar{C}_{(p)k} = \sum_{\substack{1 \le i \le N \text{ and} \\ C_i \in C_{(p)}}} C_{ik} / \left| C_{(p)} \right| \qquad \text{(EQ 69)}$$

where $\bar{C}_{(p)k}$ is the value of the kth attribute for centroid $\bar{C}_{(p)}$, C_{ik} the value of the kth attribute for the ith pattern, and $\left| C_{(p)} \right|$ the number of patterns in $C_{(p)}$.

Given a clustering and its centroids, we can decompose the amount of (Euclidean) *scatter* (variability) in the pattern data into the *within-cluster scatter* and the *between-cluster scatter*. Define the *grand* or *pooled* centroid to be

$$\bar{\bar{C}}_k = \frac{1}{N} \sum_{i=1}^{N} C_{ik} \qquad 1 \le k \le K \qquad \text{(EQ 70)}$$

where $\bar{\bar{C}}_k$ is the value of the pooled centroid in the kth pattern attribute.

Then the *pooled scatter S* is defined

$$S = \sum_{p=1}^{P} \sum_{i; C_i \in C_{(p)}} \sum_{k=1}^{K} \left(C_{ik} - \bar{\bar{C}}_k \right)^2 \qquad \text{(EQ 71)}$$

The pth *cluster scatter* is defined

$$S^{(p)} = \sum_{i; C_i \in C_{(p)}} \sum_{k=1}^{K} \left(C_{ik} - \bar{C}_{(p)k} \right)^2 \qquad \text{(EQ 72)}$$

The *within-cluster scatter* is defined

$$S_W = \sum_{p=1}^{P} S^{(p)} \qquad \text{(EQ 73)}$$

and the *between cluster scatter* is defined as the scatter for the centroids:

$$S_B = \sum_{p=1}^{p} \sum_{i; C_i \in C_{(p)}} \sum_{k=1}^{K} \left(\bar{C}_{(p)k} - \bar{\bar{C}}_k \right)^2 \qquad \text{(EQ 74)}$$

It is easily shown that $S = S_B + S_W$ (see [13] or [14]), so that scatter is decomposed

into between-cluster and within-cluster scatter. Thus if the distance measure employed in a clustering algorithm is simple Euclidean, there is no difference between the two clustering criteria ‘minimise distances to centroids within groups’ and ‘maximise distances between group centroids’. We say that S_B/S is the *proportion of scatter accounted for by clustering* $\tilde{C}$.

8.4 One-Pass Clustering

A computationally very efficient but otherwise inferior clustering algorithm can be used to generate a clustering having examined each pattern only once.

Firstly the patterns are arranged into a fixed order. This order may be arbitrary, decided by application heuristics, or determined systematically (say, in order of proximity of the patterns to the pooled centroid). Pattern number one is assigned to cluster C_1. Pattern number two is assigned to cluster C_1 also, if its ‘distance to cluster’ C_1 is less than a certain threshold value, or otherwise assigned to C_2. Subsequent patterns are assigned to existing clusters if their distance to the nearest existing cluster is less than the threshold, or otherwise to new clusters, until every pattern is assigned. ‘Distance to a cluster’ may be variously defined (e.g. Euclidean distance to cluster centroid; Euclidean distance to nearest member of cluster; etc.) resulting in a variety of one pass methods. One-pass clustering is primarily used to obtain an initial clustering for a subsequently employed iterative clustering algorithm. The threshold can be varied to obtain a hierarchy.

8.5 Graph Theoretic Hierarchical Clustering

A variety of graph theoretic clustering algorithms are in use which only employ the $N \times N$ proximity matrix $[\mathrm{d}_{ij}]$ (calculated from the N patterns), and not the pattern data themselves, to determine clusters. They may in some cases be somewhat restricted by their inability to access the original pattern data.

Note that, assuming $\mathrm{d}_{ii} = 0; (1 \le i \le N)$ and $\mathrm{d}_{ij} = \mathrm{d}_{ji}; (1 \le i, j \le n)$, there are $(N/2)\ (N-1)$ distances to consider, which is prohibitive when N is very large. Before clustering, the distances are replaced by their rankings, so that the two closest patterns in the database have rank distance 1, and the two most distant patterns rank distance $(N/2)\ (N-1)$ (ties in proximity are usually assumed absent, or can be broken arbitrarily).

The two best known graph theoretic algorithms for clustering are *single link cluster-*

ing and *complete link clustering*. The algorithms may be applied in either top-down (divisive) or bottom-up (agglomerative) versions, with no difference to the resultant hierarchical clusterings. Here consider the bottom-up versions, which begin with each pattern in its own singleton cluster. A series of threshold graphs is constructed on the nodes, where every node represents a pattern. The first threshold graph has only one edge, which connects the two patterns with rank distance 1 (i.e. the closest two patterns). The *k*th threshold graph has *k* edges, connecting each pair of patterns with rank distance less than *k*+1. In single link clustering, the *connected* subgraphs of a threshold graph are interpreted as the clusters. In complete link clustering, the *complete* subgraphs of a threshold graph are interpreted as the clusters. The nodes in a complete subgraph are such that every node is connected to every other node. Where two maximally complete subgraphs overlap, the earliest one formed takes precedence. (FIGURE 6.) gives a few threshold graphs for a hypothetical data set.

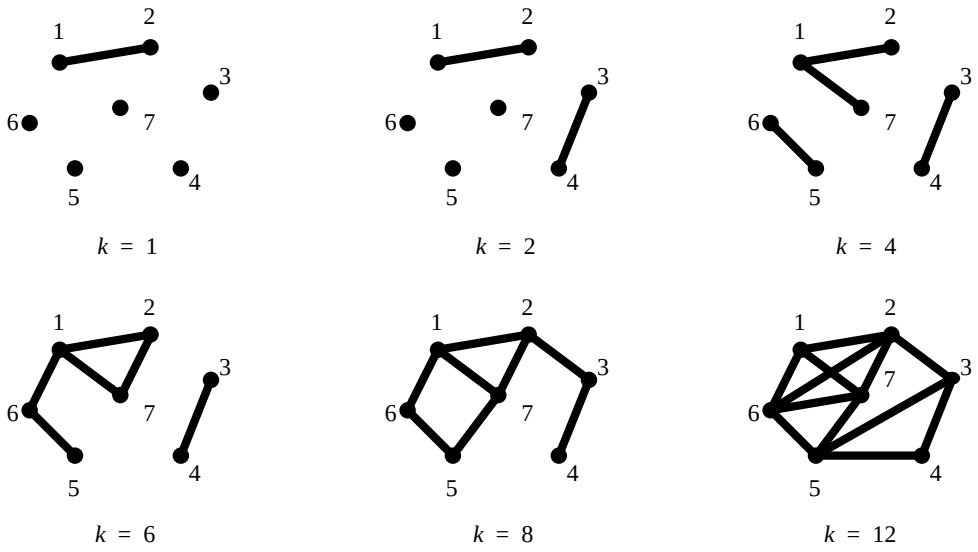


FIGURE 6.

For *k*=1, (1,2) forms a single-link cluster and a complete-link cluster (since nodes 1 and 2 are connected, *and* form a complete subgraph). Every other node is in a singleton cluster. By *k*=4, (1,2,7) forms a single link cluster, but not a complete link cluster (since 2 and 7 are not connected). (1,7) is not a complete link cluster since node 1 is in (1,2) which takes precedence. At *k*=6, there are just two single link clusters, and (1,2,7) is a complete link cluster. By *k*=8, there is only one (universal) single link cluster; (1,2,7) is still the largest complete link cluster. By *k*=12, (1,2,6,7) is a complete link cluster, as is (3,4,5), but not (5,6,7), since (1,2,6,7) takes precedence.

Since if any pair of patterns are in the same cluster in the *k*-th threshold graph they will still be so in the (*k*+1)-th, the sequence of clusterings is hierarchical. Single link clustering tends to form ‘stringy’ drawn-out clusters; complete link clusters are usually compact and near hyperelipsoidal.

Other graph theoretic methods employ a cluster criteria (on subgraphs of threshold graphs) that is intermediate between connectedness and completeness. Hierarchical clusterings can be visualised in a dendrogram, a type of tree in which the height on the page at which two nodes are joined indicates the relative distance of the nodes. A dendrogram corresponding to the single link clustering based on the threshold graphs of (FIGURE 6.) is shown in (FIGURE 7.).

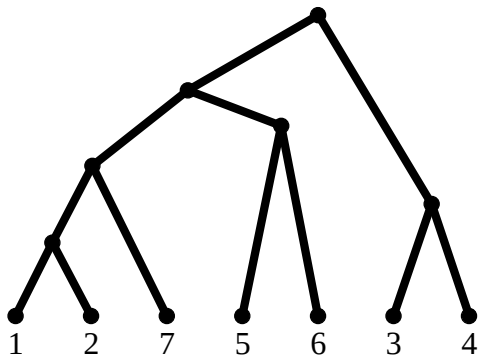


FIGURE 7.

8.6 Non-Graph Theoretic Hierarchical Clustering

If pattern variables are discrete, it is possible to form a hierarchical partition by starting with the universal cluster (which contains every pattern) and, at each stage, selecting a pattern variable on which to partition; each node in the decision tree has one child for each value of the variable selected for partitioning. The variable selected is that which ‘best predicts’ the other variables: choosing that variable which has least sum of mutual informations (see section 3.3) with the other variables is one option.

If the pattern variables are continuous, the variable selected for partitioning might be that with the greatest sum of correlations with the other variables. Another option is to partition not on an explicit pattern variable, but on the principal components of the pattern variables. The first variable selected for partitioning is the first PC, the next the second PC and so on. Such algorithms can be termed *direct splitting* algorithms, or *intrinsic decision tree clustering.*

The *Filtering Algorithm* is a hierarchical version of *K*-means clustering (see 8.7). An initial binary tree forms the first clustering. Then each pattern is ‘filtered down’ the hi-

erarchy, by assigning it to the closest of the two cluster centroids available at each binary split. An initial binary tree might be obtained from the methods of the preceding two paragraphs.

The *Binary Splitting Algorithm* divides a set of patterns into two clusters, and applied recursively generates a hierarchical clustering. If the clusters are denoted $C_{(1)}, C_{(2)}$, then the weighted averages of the *k*th variable over the two clusters are

$$\mu_{(p)k} = \frac{\sum_{C_i \in C_{(p)}} w_{ik} C_{ik}}{\sum_{C_i \in C_{(p)}} w_{ik}} \qquad p = 1, 2 \qquad \text{(EQ 75)}$$

and the error of a binary partition is based on the sum (over *K* variables) of the squared differences between these averages,

$$\sum_{k=1}^{K} (\mu_{(1)k} - \mu_{(2)k})^2 \qquad \text{(EQ 76)}$$

The weights w_{ik} are usually the number of observations combined into the pattern C_{ik}. Initially $C_{(2)}$ is empty. The pattern which most decreases or least increases the error given in (EQ 76) is transferred to $C_{(2)}$ at each step until $C_{(1)}$ is empty. The best of the N partitions thus examined is selected.

The *Two-Way Splitting Algorithm* (Hartigan, [13]) does not distinguish between patterns and variables: the objective is to partition the pattern matrix $[C_{ik}]$ by simultaneously building a hierarchical partition of variables and a hierarchical partition of cases, with the objective that the responses within any cell (cluster) have a variance less than some predetermined threshold *T*. Transposing the pattern matrix (so cases become pattern variables and vice versa) does not affect the resulting two-way clustering. Initially there is one universal cluster. At each stage, either that column or that row which has the greatest variance is partitioned according to the binary splitting algorithm (preceding paragraph), excluding from consideration those rows or columns whose variance is less than the threshold *T*. As well as the marginal hierarchies on variables and cases, a joint hierarchical partition is generated on both.

Hartigan [13] also presents an analogous technique based on joining (variables/cases) rather than splitting.

8.7 Partitional Clustering

If there are only a handful of patterns *N* and a prefixed number of clusters *P*, it may be reasonable to exhaustively calculate the within-cluster scatters for all possible clusterings, and select the clustering with the least within-cluster scatter. However unless both *P* and *N* are small, exhaustive search is computationally infeasible, and local optimisation of some form must be used to estimate the 'best' *P* clusters. The most widely used partitional clustering algorithm is known as *K-means clustering* (*P*-means in our notation). Here it is presented in a form which allows the number of clusters *P* to vary (slowly) from an initial preset.

1. Select initial *P* clusters (forming a partition) and compute cluster centroids.
2. Generate a new partition by in turn assigning every pattern to the cluster whose centroid it is nearest to.
3. Recompute centroids.
4. Repeat 2 and 3 until no patterns are reassigned in 2.
5. Adjust *P* by splitting large clusters, and/or merging small ones, and/or removing small outlying clusters. Repeat from 2.

Various methods can be used to compute the initial clusters (step 1), and various criteria used to decide when to stop. The details of the algorithm can be varied substantially to produce many different clustering algorithms. An important variation is to introduce some random perturbation of the clusters after stage 3, allowing the algorithm to buck local optima—a simulated annealing approach. These algorithms are good at finding compact hyperelipsoidal clusters.

Mixture Decomposition (a.k.a. statistical pattern recognition) is a statistical approach to clustering, in which patterns are assumed to have been drawn from one of *P* underlying populations of known distribution. Hill climbing is used to obtain a locally maximal likelihood estimate for the memberships of patterns to populations—the principal problem is formulating the underlying model.

Density Estimation approaches to clustering operate by choosing as cluster centres the local maxima of some density estimate in pattern space. Clusters in such a scheme are essentially regions of high density in pattern space. Simple histograms as well as non-parametric estimates of pattern density have been used.

Fuzzy Clustering does not impose that any pattern should belong to just one cluster, but instead assigns degrees of membership $f_p, \ldots, f_P$ for each pattern to each cluster. The objective is to iteratively recluster so as to minimise some measure of 'fuzziness' whilst maximising between cluster distance.

8.8 Using Variables Extrinsic to Pattern

8.8.1 Using Extrinsic Ordering

Thus far we have examined only intrinsic clustering methods. One helpful piece of extrinsic data we might employ when clustering patterns is a non-strict ordering on the patterns (an ordered discrete extrinsic variable). Most commonly this will be a temporal variable measured on each case. If we restrict the form of the clustering so that each cluster must represent a contiguous time interval, the number of possible clusterings is very dramatically reduced. Fisher (see e.g. [13]) uses such a restriction to facilitate an exhaustive search of partitions, practical for moderate N and P. If a hierarchical partition is sought, Fisher gives a particularly simple and rapid algorithm which repeatedly splits the time interval, and is practical for even very large N, P (though does not guarantee a globally optimal partition).

8.8.2 Using Extrinsic Categories

As well as simple temporal orderings there may be categorical information which can be used to determine cluster structure (but which is not part of the pattern). If there are continuous variables we wish to exclude from the pattern (i.e. continuous predictors), we can discretise them (partition them into ordered discrete variables).

Categorical predictors can be used in various ways to guide cluster formation. Assuming all predictors are discrete, first let us compress the total number of cases (patterns) into the reduced set of *atomic patterns* or *atoms*; each atomic pattern is the centroid of all those patterns whose categorical predictor values are identical. If $C_1, \ldots, C_{i*}, \ldots, C_{N*}$ are the initial cases, let $A_1, \ldots, A_i, \ldots, A_N$ be the induced atomic patterns with kth response

$$A_{ik} = \frac{\sum_{C_{i*} \in \underset{\sim}{X}_{(i)}} C_{i*k}}{\left|\underset{\sim}{X}_{(i)}\right|} \qquad 1 \le i \le N; 1 \le k \le K \qquad \text{(EQ 77)}$$

where $\underset{\sim}{X}_{(i)}$ is the *i*th non-empty cell in the maximal partition $\tilde{\underset{\sim}{X}}$ of predictor space. Naturally, atom A_i has associated predictor values equal to the common predictor values of its constituents. The maximal partition of predictor space is that partition in which every cell represents a unique combination of attribute values, $\{X_1 = x_{1v_1}, \ldots, X_J = x_{Jv_J}\}$. Such a cell is empty if there is no pattern matching this specification.

We can now perform any intrinsic clustering algorithm thus far described on the atomic patterns $A_i; (1 \leq i \leq N)$, rather than the original cases. Essentially we thus restrict our clustering technique so that it is forced to put cases with identical predictor values in the same cluster. We will call this methodology *atomic clustering*.

It may be prudent to modify any atomic clustering algorithm so that the *weight* of (i.e. number of initial patterns represented by) each atom is taken into account when calculating centroids, scatter measures, and so on. Moreover, information about the within-atom spread or scatter can be retained and utilised in calculating the within-cluster scatters.

Note that atomic clustering does not make any restrictions about the nature of the predictor variables represented within a cluster. A related approach, which we will call *extrinsic decision tree clustering*, allows less freedom for predictor values in a cluster. The idea is to cluster by recursively partitioning the predictor space using one predictor at a time, so as to locally maximise between-cluster variation *of patterns* (or minimise within-cluster variation). At each stage, the domain of one predictor is to be partitioned; the centroidal patterns corresponding to each predictor value can be treated as the patterns in any intrinsic clustering algorithm. The ‘best’ predictor and ‘best’ partition of that predictor’s domain should be used at each stage.

A great advantage of this methodology is that each constituent clustering problem has at most $\max_j(|\underset{\sim}{X}_j|)$ patterns to cluster (in the usual notation); a disadvantage is the inflexibility and local greediness of the approach.

Neither of the above approaches allows the ‘splitting’ of atoms, a restriction which may need to be addressed when an atom is built from constituent patterns which are markedly dissimilar—i.e. when cases with identical predictor values have diverse patterns.

Chapter 9 — The Load Profiling Task

9.1 Task Selection

After consultations with the PhD project's Sponsor organisation regarding their databases, two particularly promising data sets were identified for investigation using data mining techniques: fault data for high voltage (HV) lines, and half-hourly electrical load readings for customers of known and various type.

The fault database comprised several thousand abnormality reports on HV power lines. For every interruption, attributes recorded included data and time of interruption; method(s) of fault clearing; direct and contributory causes where known; main equipment/components involved; age and manufacturer of equipment; and information concerning progress of restoration (major faults are restored in several stages, and various data recorded for each stage); and more. Further, some records had plain language entries describing aspects of the incident.

The faults database was characterised by a great number of missing data (especially regarding the cause of faults), and of subjective entries (particularly the plain language entries). The rather non-specific objectives of the data mining task were to seek out interesting generalisations about the fault data (for example, regional and seasonal trends); and to identify any structure underlying the data which was not previously known.

The half-hourly load databases comprised the meter readings every half hour, over some years, for various customers, together with information on each of the customers, and hourly weather variables (temperature, wind speed, humidity, cloud cover and rainfall) and daily sunset times, for the period. The first such database comprised load data for business customers on monthly billed tariffs; their tariff codes, two-digit Standard Industrial Classification (SIC) codes, total consumptions, maximum demands, and load factors were supplied. A second database of interest comprised quarterly billed business customers, and a third comprised residential customers; customer information for these databases derived from customer questionnaires. The data mining task was to analyse daily load/time-of-day profiles with regard to their dependencies on the weather, day type, month, and customer information. This type of task is known as load profiling.

The half-hourly load profiling task was selected, primarily because the data structure is simpler (each record the same size and complexity) and the data contains less ambiguity (less missing data; less subjective data). There is a greater depth of literature on

load profiling and forecasting than on analysis of line fault data.

At the Sponsors' behest, only the monthly billed customer database was studied for this PhD project, rather than also the quarterly billed business customer or the domestic customer databases.

It is worth noting that line fault databases are thought to contain much hidden and valuable information, and that data mining techniques are not necessarily less applicable to this problem than to the load profiling task.

9.2 Importance of Load Profiling Tasks

The relationships between daily load profiles and knowable factors such as weather conditions, temporal factors, and customer characteristics, are known to be complex, non-linear, and have many mutual interactions. A better understanding of the behaviour of daily load profiles is desirable since the load shape, as well as the daily peak load, are vital factors in planning the production and pricing of electricity. In the short term, it is essential to know with as much accuracy as possible what the total and local system demand will be in the next minutes, hours, and days, so that generators with various start-up times and start-up costs can be switched on or off or vary their output levels so as to optimise the total efficiency of generation. This is the domain of *Short-Term Load Forecasting* (STLF), and STLF methodologies often utilise various daily load profiles to this end.

Medium- and long-term forecasting seeks to predict daily peak load and load shape weeks, months, and years hence. Load shape and peak load are affected by demographic and socioeconomic trends as well as seasonal and weather factors, and so load profiles disaggregated according to socioeconomic/demographic factors are of particular importance in longer term planning.

Most of the research until now into the behaviour of load profiles has been for the purposes of short/medium-term and (particularly) long-term forecasting. However, load profiling is of particular interest at current in the United Kingdom because of the unique competition structure which is legally imposed on the UK electricity industry. All electricity customers in England and Wales, irrespective of their location or their peak or annual demand, can choose from which licensed UK supplier they purchase their electricity.

Whilst the purchase costs of electricity to the supplier are subject to daily and sea-

sonal fluctuation, the supplier is unable to pass on the price variation and subsequent risk to the customer because most of these customers are not equipped with intelligent two-way metering. In order to minimise their risks and maximise profits, suppliers need flexible tariff structures tailored to the load profile of the customer being charged.

On the other hand, there is so much diversity in customer profiles that it would hardly be viable to negotiate every customer's tariff on the basis of their load profile history. Rather, a manageable set of profiles intended to represent as much of that diversity as possible can be used to design a manageable set of tariff structures.

In the case of potential rather than existing customers, the supplier has effectively no available detailed profile data at all. In order to offer tariffs that are competitive without incurring undue risk on the part of the supplier, the variations between load profiles from customer to customer and from day to day must be better understood. Such is the variety of load profiles in the population that the deliberate 'poaching' of customers having unusual (but advantageous) load shape with the offer of a tailored tariff may be viable, where such customers can be identified.

Hence there is a need for reliable models of load shape which attempt to explain the half-hourly and the daily and seasonal variations in load as responses to the time of day, the time of year, weather conditions and known or observable customer information.

Nor is tariff determination the only reason to study models of load shape. As well as the well documented importance of profile models in load forecasting, concise sets of 'standard profiles' that nevertheless maximally capture load shape variability are desirable in load flow analysis.

9.3 Objectives of the Load Profiling Task

Customers of differing types and characteristics have differing daily load shapes. Not only does load shape differ according to the weather, the day of the week, and the time of year, but it does so in a different manner for customers of different characteristics. Given a set of customers, their recorded attributes, and their half-hourly demands over a period of some months, the primary objective of our load profiling task, agreed upon with the Sponsors, is: to build models which estimate, for certain subsets of customers, their load shapes *and* confidence estimates for those load shapes, for different weather conditions, times of year, and days of the week. The generation of sets of 'standard profiles', which are not too large but still capture most of the variability in the database, is

part of that task. This will be formally formulated in Chapter 10.

Secondary objectives include producing models and visualisations which help to promote understanding of whatever underlying structures there may be in the load data; and identification of the relative importance of, and interactions between, the various predictor variables (such as weather, temporal variables and customer attributes).

To give some idea of the huge variety of load shapes existing in the business customer databases to be studied, some 3 dimensional overviews of seasonal variation in load shape are presented in (COLOUR FIGURE 1.) to (COLOUR FIGURE 4.), which are drawn from a 1994/5 database of business customers' loads. The y-axis shows time of day, the x-axis shows time of year, and the z-axis (which is represented on a smoothly changing scale of colour) shows the corresponding half-hourly load. A full explanation of this type of colour load diagram is given in section 12.5. These diagrams demonstrate that not only from customer to customer but from day to day, load shape can vary in many different ways; they go some way towards showing what a difficult task load shape modelling can be, and why data mining methods which can adapt to such extreme variation are of importance in load profiling problems.

9.4 A Review of Literature on Short-Term Load Forecasting

9.4.1 Overview

Short-Term Load Forecasting is a field of study with certain relevance to load profiling (although load profiling does not aim to predict hourly or half hourly total system load in the immediate future—the aim of STLF. The principal common ground between these fields of study is that practitioners from either domain must find ways of accounting for effects of weather, day-of-week, time of year, and daylight saving clock changes in their models. The identification and modelling of special days and periods (bank holidays, Christmas and Easter vacations, school holidays, periods of industrial action, and so on) is also of interest to either practitioner. However in STLF considerable attention is paid to modelling the hour-to-hour variations in load, and accounting for hour-to-hour errors in the forecast, using time series analysis. In load profiling, modelling detailed hour-to-hour variation in load as a time series is not the intention; rather we aim to accurately model the variation in *daily load shape* due to customer and temporal variables. There is somewhat more literature regarding STLF than load profiling; [15] provides a good overview and thorough bibliography. A review of some salient examples of load

forecasting procedures (next) precedes a review of load profiling literature (9.5).

9.4.2 United Kingdom Short-Term Load Forecasting Methodology

The UK STLF regime is described in the report [16]. An adaptive regression technique with many levels is employed, and human judgement is required at certain stages. Separate models are derived for GMT and BST[1], since the effect of daylight saving on load is quite dramatic; and for weekdays, Saturdays and Sundays, as load shape varies substantially between these day types—six models in all.

A number of *cardinal points* of the daily load shape are selected, usually at maxima, minima and inflections of the load shape. Using historical data, loads at each cardinal point are regression modelled by a low order polynomial in three weather statistics: effective temperature *TE*— an exponentially weighted average of recent temperatures; effective illumination *EI*; and the cooling power of the wind *CP*. These are derived from basic weather data. A slowly changing quartic polynomial in *n*, the day number, accounts for seasonal changes.

From these cardinal point models, historical data is reconstructed into *basic demand* (hypothetical loads at the cardinal points affected by neither weather nor day of the week) by subtracting the weather/day-type models from total demand. To produce target day load forecasts (usually just one day ahead) basic demand is reconstructed into predicted demand by reintroducing the weather/day-type model using the target day's weather forecast.

The final stage is to construct a continuous profile for the target day from the cardinal point forecasts. This is done by selecting a profile from the historical database which is likely to be similar to the profile for the day to be forecast, and stretching and compressing it so as to force it to interpolate the forecasts at the cardinal points. It is important that the historical profile selected is from the same time of year (and ideally day of week) as the target day, possibly from a previous year, so as to capture seasonal variation in load shape. Bank holidays are treated as either Saturdays or Sundays (according to experience) in the model building phase; days adjacent to bank holidays (*window days*, which are known to have perturbed load shapes) are treated normally in the model building phase. However in the interpolation phase, the historical profile selected is usually taken

1. Greenwich Mean Time; British Summer Time.

to be that from the same holiday/window-day from the previous year.

The UK forecasting method nicely illustrates many of the complicating factors in load profiling tasks, and some common ways to deal with them. Of particular interest is the manner in which the effect of weather on historical loads is modelled using multiple regression, and then the modelled effects 'filtered out' to recover hypothetical 'weather-free' loads. We shall refer to this as *deweathering*, and to the subsequent reintroduction of weather effects given certain weather conditions, *reweathering*.

9.4.3 Load Forecasting by Identification of Similar Previous Days

Moghram & Rahman [17] describe a Knowledge Based Expert Systems (KBES) approach to forecasting. Based on a large historical database of daily profiles, the KBES selects a past reference day expected to be similar in profile to the target day. This is an automation of a task performed using human judgement in the method of section 9.4.2. The KBES also has rules which reshape the profile of the reference day based on weather and any other factors expected to cause variation.

Another KBES is described by Jabbour et al. [18]. A historical database of profiles and weather conditions is searched to find the eight days whose weather conditions are the eight nearest neighbours to the weather forecast for the target day (nearest by weighted Euclidean distance). The eight corresponding load profiles are averaged to produce the basic forecast, which is modified by various means.

Rahman & Bhatnagar [19] also describe a system which averages profiles from days with weather similar to that forecast for the target day. They advocate the use of four separate models for the different seasons, but note that load behaviour changes more rapidly at the boundaries between the seasons; at seasonal boundaries, both the appropriate models are implemented and monitored for accuracy.

In a related approach, Dehdashti et al. [20] perform an initial hierarchical clustering of weather patterns. Variation within each cluster is assumed inevitable and due to unknown factors, and consequently it is the most *recent* days in the appropriate cluster (rather than the most similar) which are used as reference days. The data library used is different for different times of the year, the transitional periods between seasons having more data since these are the most unpredictable periods. Different weather variables were used for each time-of-year period and each time-of-day period.

9.4.4 Forecasting Procedures Using Spectral Analyses

Lijesen & Rosing [21] propose some unusual ideas to model weather effects when the weather is changing rapidly. *Nominal* weather conditions are averaged from either the seasonal historical weather, or the recent past weather. They then calculate the deviation between the weather forecast for the target day and the nominal weather, and use this to calculate the *total* amount of energy which should be added or subtracted from a weather-independent profile forecast for the target day. How this energy should be distributed over the target day is determined from a *spectral analysis* of the weather dependent load over the recent past.

Many STLF techniques involve spectral analysis of recent or historical loads. Spectral analysis is a form of linear regression analysis for multiple responses problems, in which the fitting functions are selected especially so as to represent certain features known or postulated to affect the multiple responses additively.

The idea is to identify a small set of functions of time which account for the majority of the variation in the profiles under consideration. If the load profiles $Y_1(t), \ldots Y_N(t); (1 \le t \le T)$ each take T values (T=48 for half-hourly profiles), we seek a set of M fitting functions $\phi_1(t), \ldots, \phi_M(t); (1 \le t \le T)$ (where M is much less than T) with the objective that least-square-error regressions

$$Y_i(t) = \sum_{m=1}^{M} \beta_{im}\phi_m(t) + \varepsilon_i(t) \qquad \text{(EQ 78)}$$

for each of the $Y_1, \ldots, Y_N$ over the T time values yield low square errors.

The M functions ϕ_m may consist of preordained functions selected by an expert; the principal components of the profiles (see section 5.2); sinusoids (whose frequencies can be determined from Fourier theory). Preordained choices for the fitting functions might include certain load profiles to which the Y_i are suspected to bear resemblance.

Once a set of fitting functions has been identified and the regressions of (EQ 78) been performed, the values of the coefficients β_{im} can be seen as a reduced set of data which represent the profile Y_i. This invites a hierarchical regression (section 6.3) of the β_{im} on the other predictors.

Another example of spectral analysis in STLF is from Christiaanse [22]. The fitting functions used are sinusoids of periods 168hrs, 24hrs, and 12hrs, as well as the second, third, fourth, fifth, sixth and eighth harmonics (84hrs, 56hrs, 42hrs etc.).These periods

correspond to spikes on the autocorrelation function of the load time series. The regression uses weighted least-square-error with weights $\alpha^j;\ j = 0, 1, 2, \ldots; (0 \leq \alpha \leq 1)$ where j is the age in hours of the load measurement under consideration, to produce a constantly moving exponentially smoothed spectral decomposition of load as a function of time.

9.5 A Review of Literature on Load Profiling

9.5.1 Overview

A selection of techniques which model daily load shape as a function of predictor variables including weather conditions, temporal factors, customer characteristics and end use types and appliances, are presented. Such models are sometimes referred to as load shape models.

Note that end use types (e.g. cooking, lighting, air conditioning) and actual end use appliances (e.g. televisions, cookers) are not recorded in the Sponsors' database being studied. Therefore the techniques modelling the effects of weather, temporal predictors, and customer characteristics are of greatest interest to our study.

9.5.2 Thermodynamic End Use Models

A number of methods build thermodynamic models for load shape contributed by certain end uses— particularly air conditioning and storage heating. Predictors include weather variables, thermal characteristics of buildings, thermostat settings, and tariff information (in particular the time of the onset of cheap-rate tariff, where storage heating loads peak [23]).

Such physical models for load shape (see e.g. [23], [24], [25]) rely heavily on factors not recorded in the Sponsor's data, such as thermal properties of buildings, though other methods have obtained end use profiles empirically, without using a physical model.

9.5.3 Mixed and Non-Thermodynamic Models for End Use Load Shape

Gellings & Taylor [26] disaggregate load shape into around 160 components. Loads are decomposed into a weather-dependent and a basic (weather independent) component using the notion of *no-weather days*. A no-weather day is one in which weather-dependent loads are assumed absent (say, when Fahrenheit temperature is between 55 and 70,

and humidity is less than 60%). No-weather days provide base curves, which are subtracted from total curves to obtain weather-dependent curves for those days where weather-dependent loads are considered present.

Storage heating and storage air conditioning have complex thermodynamic models; other weather sensitive loads (e.g. refrigeration) have piece-wise models as functions of temperature, including seasonal and weekly response factors.

Domestic loads are broken down into thirty-five end uses (e.g. washing machine), and industrial and commercial loads disaggregated by tariff code (SIC codes were tried initially put proved somewhat inconclusive). In order to disaggregate load into end uses, data must be present for each customer regarding ownership of appliances. The method requires much data absent from the Sponsor's databases.

Broehl [27] also disaggregates load shape according to end use. For residential customers, experimental data concerning the use patterns of various end use appliances was available (a matrix with dimensions 24 hours by 4 day types by 12 months) which was used in conjunction with socioeconomic and demographic factors such as appliance saturation (what percentage of households have a certain appliance), and experimentally obtained appliance wattages. Weather dependent appliances have a partitioned temperature variable in place of month. Industrial customers are disaggregated by SIC code, and commercial loads divided into heating, air conditioning, and other. Again, the method requires data unavailable in the Sponsor's databases.

The papers [28] and [29] consider the estimation of residential end use load shapes from whole-house metered data. The study comprises a longitudinal study (modelling the behaviour of load over time) and a cross-sectional study (modelling load behaviour in terms of household characteristics) combined in a hierarchical regression.

Weather dependency is first removed—the weather-dependent component is modelled as a non-linear thermodynamic system. The longitudinal analysis uses a spectral analysis (decomposition into sinusoids and other simple functions). This compresses longitudinal load data for each household into a few parameters. In the cross-sectional analysis, the coefficients of the fitting functions are regressed on household characteristics.

Appliance end use load shapes are generated by aggregating all household level profiles for those households *with* an appliance and subtracting the aggregate profile for households *without* the appliance. Willis & Brooks [30] attempt to reconstruct approximate 8760^1 hour load profiles for various end uses, from a number of data sources. They

note that most utilities have total system loads hourly, billing-class loads monthly, loads on various feeders and substations etc. hourly, and unreliable data on the relative breakdown of end use categories of load and appliance saturation. The idea is to fill in as many as possible of the gaps to reconstruct yearly end use profiles, utilising the facts that some feeders and substations supply a single class of customer; that all hourly loads must sum to the total system load; and that all hourly loads must integrate to the appropriate monthly billing total. The recovered 8760 hour profiles are approximated using either 77 or just 48 numbers.

9.5.4 Load Profiling Using Cluster Analysis

Papers [31] through [33] use cluster analysis (chapter 8) to obtain groupings of similar load profiles.

Müller [31] performs extrinsic clustering using various sets of predictors; the patterns to be clustered are daily total system profiles for a number of years. The daily profiles are first normalised by dividing through by the daily mean.

In a weather based clustering, nine weather variables recorded for each day (temperature at various times; minimum and maximum temperature; previous days average temperature; illumination means for various periods) are used as disaggregating predictors. In a seasonal clustering, a seasonal index, plus day of the week (including a ‘bank holiday’ day type) were the predictors. A special days clustering used a summer/winter variable and a day type variable (with values such as workday, Sunday, holiday, pre-holiday day, etc.) predicted the clusters. *K*-means clustering and a top-down decision tree clustering were investigated.

In [32], a hierarchical clustering of daily profiles is used as the basis of a method for load forecasting. In [33], a rapid iterative statistical clustering technique for clustering profiles within a spreadsheet is described. Extrinsic (i.e. non-profile) data is not used in papers [32] or [33].

1. Number of hours in a year.

Chapter 10 — Task Formulation for the Monthly Billed Business Customer Database

10.1 Data For the Monthly Billed Business Customer Database

Unfortunately, the data provided by the Sponsor is incomplete, inconsistent, and inaccurate in various ways. This is not an unusual state of affairs (in data mining nor in the electricity supply industry). The data was delivered in flat file format, which had been cobbled together from various sources, this being the only data that was available.

The half-hourly load profile data hails from three load databases (1994/5, 1995/6, 1996/7), covering different time periods (with some slight overlap) but not forming a continuous record. Certain days, and entire months, of data are missing for various customers, and it is not the same days and months which are missing for each customer. An overview of the extent of the missing data in the 1995/6 and 1996/7 databases is given by the visualisation in (COLOUR FIGURE 9.) (though a handful of missing dates illustrated in that diagram were deliberately removed because they were bank holidays or holiday periods). Colour figures are to be found in the Appendix to this Thesis. The 1994/5 database covers October 1994 through April 1995 inclusive (and is complete); the 1995/6 and 1996/7 databases each cover April through March inclusive (and have omissions).

The customer information data covers 500 customers; these match perfectly with the customers present in the 1994/5 load database, but the 1995/6 and 1996/7 load databases contain many customers not among those 500, and do not contain data for all the 500 customers for which customer information is recorded. The customer information comprises the customers' two-digit SIC codes (12.4% of which are missing, and a few of which are not valid SIC codes); their tariff codes; their maximum demands; their load factors; and their maximum measurement errors. Not only are some SIC codes missing or invalid, but it is suspected by the Sponsor that some of the recorded SIC codes may be inaccurate. Load factor (the ratio of average load to maximum demand, expressed as a percentage) is a measure of the extent to which the peak demand is maintained throughout the year. The tariff codes each correspond to a different tariff schedule used by customers, i.e. a different way of pricing their electricity through the day. SIC (Standard Industrial Classification) codes carry information about the type of business a customer carries out; some two digit codes represent rather broad business descriptions (e.g. the

retail and repair industries are, with few exceptions, represented within the same two digit SIC code).

There are certain questionable data within all three of these loads databases. Particularly, there are instances where load increases by a very conspicuous amount for just one half hour reading, and then drops back to about the same level as before; while it is not possible to be certain that these data are actually erroneous, it seems rather likely in many cases that they are. There are also conspicuous cases where the reverse happens (i.e. a dramatic but short-lived fall in load). These are much more common, and are likely on the whole to be genuine artifacts, caused by total or near-total loss of power, resulting from an electrical fault of some type, or a power outage. Even so, these must be classed as noise since there is no discernable pattern to the instances and no variable capable of explaining them in any model. This type of data feature will be called a *solecism*.

The weather data for the study period comprises the following variables, recorded hourly: ambient temperature, windspeed, humidity, cloud cover (in *Oktas*, i.e. eighths, convertible to a percentage), and hourly rainfall; but the hourly rainfall only covers up until the end of March 1995 (i.e. six out of seven months of the 1994/5 loads database) and is unavailable thereafter. Additionally the daily sunset times were supplied (for one year only, though they differ very little from year to year; and not for the exact region where the customers are actually situated).

A serious problem in the data is that where the 1994/5 loads overlap the 1995/6 loads, there are discrepancies between the two databases. The discrepancies take one of two forms: in the first form, profiles of a given customer in the second database are translated by a small fixed constant β from the corresponding profile in the first database; in the second form, profiles of a given customer in the second database are scaled by a scalar α from the corresponding profile in the first database. Whilst for any particular customer, β or α was fixed, α ranged from 0.46 up to 1.28. The full predictor variable set including these two derived variables is described in (TABLE 1.)-(TABLE 3.).

TABLE 1.

Predictors (Supplied, Non-Weather)

Notation	Description	Type	Domain
X_1 or t	time of day	ordered	[00.00h, 00.30h,..., 23.30h], equivalently, [1, 2,..., 48]
X_2 or d*	day index	ordered	[Oct 1 1994,..., Mar 31 1997], equivalently, [1, 2,..., 913]

TABLE 1.

Predictors (Supplied, Non-Weather)

Notation	Description	Type	Domain
X_5	SIC code	categorical	various two figure integers
X_6	tariff code	categorical	various alphanumerical codes
X_7	maximum demand	continuous	non-negative real numbers
X_8	load factor	continuous	positive real numbers

TABLE 2.

Predictors (Derived, Non-Weather)

Notation	Description	Type	Domain
X_3 or d	day type	categorical	[Mon, Tue, Wed,..., Sun], equivalently, [1, 2,..., 7]
X_4 or m	month index	ordered	[Jan, Feb,..., Dec], equivalently, [1, 2,..., 12]

TABLE 3.

Predictors (Supplied, Weather)

Notation	Description	Type	Domain
X_9 or θ	temperature	continuous	real numbers
X_{10} or w_w	windspeed	continuous	non-negative real numbers
X_{11} or w_h	humidity	continuous	non-negative real numbers
X_{12} or w_c	cloud cover	continuous	[0, 100]
X_{13} or w_c	rainfall	continuous	non-negative real numbers
X_{14} or w_s	daily sunset time	continuous	positive real numbers

A variety of derived weather variables are of interest also; these will be introduced in section 10.5.3. The maximum measurement errors are of possible interest in determining model goodness-of-fit, but are of no use as predictors, and hence were not considered.

10.2 Normalisation of Load Profiles

Consider the load data for two customers c_1, c_2.[1] Suppose the total energies consumed (in the study period) by the two customers are denoted E_1, E_2 respectively. Total

energy consumed is perhaps the best measure of the customers' sizes (although maximum demand X_7 is a viable alternative). The database contains customers of significantly different sizes, and thus load profiles from the same day for two different customers are not directly comparable; in order to compare the load profiles it is necessary to normalise them (convert them to comparable scales), and there are many candidate normalisations available. At one extreme, (*coarse* normalisation) we normalise each customer c_l's load readings by dividing each reading by the customer's total energy consumption E_l; at the other extreme (*fine* normalisation), we divide customer c_l's load readings for a particular day by the total energy consumed *on that day only*. Intermediate normalisation schemes might divide daily load readings by the monthly energy total, the seasonal total energy, the total energy for all days of the same *day type* over the seven months, the total energy corresponding to a certain month *and* day type, and so on. There is a danger associated with too fine a normalisation, in that two customers whose annual load shapes vary from each other significantly might appear to have similar annual load shapes after normalisation, and so distinction between dissimilar customers may be blurred. On the other hand, the fact that two load profiles from different two customers have the same basic *shape*, but different daily energy *totals*, can sometimes be lost when a coarse normalisation is used.

Now consider the effects of normalisation when *aggregating* (rather than comparing) load profiles (for example when computing centroids for clustering algorithms; when aggregating profiles in the same partition cell in a regression tree; and so on). If load profiles have been normalised (according, say, to annual energy consumption) then all customers have the same weight in the aggregated profile. It is important to decide whether or not this is desirable. There is an obvious case that 'larger' customers should make a greater contribution in aggregate profiles (as occurs when no normalisation of the above type is in place)—since their contribution to total system load is greater. However there is an argument for aggregating normalised profiles; viz that data is often sparse in certain regions of attribute space, and we cannot afford the loss of information contained in the profiles of smaller customers incurred by down-weighting them (by failing to normalise).

In all that follows, it should be assumed that all the raw data for each customer has

1. Recall that C_i denotes the *i*th *case* in the database, not the same as c_l, the *l*th customer.

been normalised by dividing through by the total estimated energy for each particular customer. Thus if Y_i^* is a raw load datum for customer l (estimated total energy consumption over the study period $\tilde{E}_l$), then

$$Y_i = Y_i^* / \tilde{E}_l \qquad \text{(EQ 79)}$$

The reason $\tilde{E}_l$ is only an estimate of total energy consumption is that customers have load data missing on some dates; $\tilde{E}_l$ was calculated by assuming the energy consumed on any day missing from a customer's data was equal to the mean daily energy observed for that customer on non-missing days.

10.3 Uniresponse Model and Multi-response Model for Monthly Billed Business Customers

For notational convenience, define $\{B_1, \ldots, B_{Bmax}\}$ to be a hypothetical set of unknown (and unknowable) predictors which affect load but whose nature and values are unattainable, and imagine that the union of the recorded predictors with the unknowable predictors is a variable set capable of accounting for 100% of variance in the load.

The response can be thought of in two ways: either as a single response variable *load*, denoted Y; or a multivariate response (or response vector), *load profile*, denoted $\underline{Y} = (Y_1, \ldots, Y_t, \ldots, Y_{48})$; time of day t can be dispensed of as a variable in the latter model. Thus we can describe two general models for the load data, the *Uniresponse Model* (UM), and the *Multi-response Model* (MM).

The most general uniresponse model we can propose for the data is

$$Y_i = f_t(t, d^*; X_5, \ldots, X_{14}; B_1, \ldots, B_{Bmax}) + \varepsilon_i^{(m)} \qquad 1 \le i \le N_U \qquad \text{(EQ 80)}$$

where the $\varepsilon_i^{(m)}$ are the additive errors due to measurement error in the data (note X_3 and X_4 can be derived from d^*). N_U, the total number of data points (i.e. cases), is given by the product of the number of half-hours in a day (48), the number of days in the study N_{days} (maximum 913 in our study), and the number of customers N_{custs} (maximum 500 in our study, $48 \times 913 \times 500 = 21\ 912\ 000$ being the maximum N_U for our study). f_t is a hypothetical 'true' model which could recover the exact load for a given customer. When we exclude the unknown attributes $B_1, \ldots, B_{Bmax}$, we can recast this as

$$Y_i = f_t(t, d^*; X_5, \ldots, X_{14}) + \varepsilon_i^{(m)} + \varepsilon_i^{(u)} + \varepsilon_i^{(s)} \qquad 1 \le i \le N_U \qquad \text{(EQ 81)}$$

The errors $\varepsilon_i^{(u)}$ are errors due to absence of unknown attributes $B_1, \ldots, B_{Bmax}$; the errors $\varepsilon_i^{(s)}$ are errors due to limited sample size and missing data. As in (EQ 2), we assume that the errors cannot be separated, and combine them into $\varepsilon_i^{(mus)}$. Finally we acknowledge that any practical model (involving regression, classification and so on) we may build will be far from perfect, thus introducing modelling errors $\varepsilon_i^{(mod)}$:

$$Y_i = \hat{f}_U(t, d^*; X_5, \ldots, X_{14}) + \varepsilon_i^{(mus)} + \varepsilon_i^{(mod)} \qquad 1 \le i \le N_U \qquad \text{(EQ 82 : UM)}$$

The multi-response model MM is a model for load profiles, rather than loads. $\underline{Y}$ is a 48 dimensional vector of half-hourly load responses (time of day is no longer a predictor) and the multi-response model corresponding to (EQ 82 : UM) is

$$\underline{Y}_i = \hat{f}_M(d^*; X_5, \ldots, X_{14}) + \underline{\varepsilon}_i^{(mus)} + \underline{\varepsilon}_i^{(mod)} \qquad 1 \le i \le N_M \qquad \text{(EQ 83 : MM)}$$

where the error terms are now vectors of 48 half-hourly errors. In (EQ 83 : MM), the number of data points (i.e. cases) N_M is limited to only $913 \times 500 = 456\ 500$, though each data point has 48 associated responses. Note that we can obtain local estimates for $\varepsilon_i^{(mus)}$ (or $\underline{\varepsilon}_i^{(mus)}$) independent of the modelling errors $\varepsilon_i^{(mod)}$ (or $\underline{\varepsilon}_i^{(mod)}$), and so do not always combine them into a single error term.

10.4 Atomic Model for Monthly Billed Business Customers

At this point it is convenient to temporarily drop maximum demand (X_7) from consideration as it was not eventually used as a predictor, and to discretise load factor (X_8) into a categorical variable X_L, with a handful of values. A maximum entropy partition (i.e. one which retains the most information about the variable partitioned) has cells which are as nearly as possible equal in size (see end of section 3.2). Hence, though replacing continuous load factors with discrete categories (such as low, medium, and high) does result in some loss of information, we lose the least information when there are approximately equal numbers of customers in each category. In practice, seven categories were always used (denoted $L_1, \ldots, L_7$).

This move allows us to treat load factor in the same manner in which we treat SIC codes and tariff codes, which is more convenient because load factor is, like SIC and tariff codes, a customer property (rather than temporal or meteorological).

With this simplification, the multi-response model becomes

$$\underline{Y}_i = \hat{f}_M(d^*, X_5, X_6, X_L; X_9, \ldots, X_{14}) + \underline{\varepsilon}_i^{(mus)} + \underline{\varepsilon}_i^{(mod)} \qquad 1 \le i \le N_M \qquad \text{(EQ 84)}$$

Note that any two cases which have identical values for d^*, X_5, X_6 and X_L are indistinguishable in this model[1]. We can aggregate the profiles of indistinguishable cases by simply averaging them, thereby reducing the number of cases. In this way we obtain the *atomic profiles* or *atomic cases* or *atoms* A_i for the load data. This is identical with the notion of an atomic pattern (section 8.8.2). With $\underset{\sim}{X} = \underset{\sim}{d}^* \times \underset{\sim}{X}_5 \times \underset{\sim}{X}_6 \times X_L$ the domain of the discrete predictors and $\underset{\sim}{X}_{(i)}$ the *i*th non-empty cell in its maximal partition, (EQ 77) defines the atoms (it is repeated as (EQ 85) for convenience).

$$A_{ik} = \frac{\sum_{C_{i^*} \in \underset{\sim}{X}_{(i)}} C_{i^*k}}{\left|\underset{\sim}{X}_{(i)}\right|} \qquad 1 \le i \le N; 1 \le k \le 48 \qquad \text{(EQ 85)}$$

$A_{ik}; (1 \le k \le 48)$ are the 48 response values for the *i*th atom, *i* the index for the atoms, and i^* the index for the original (non-aggregated) profiles. Let $\underline{A}_i$ be the 48 dimensional vector of responses for atom A_i. Denote by X_{ji} the value taken by predictor X_j for atom *i*.

We can modify (EQ 84) to the Multi-response Atomic Model (MAM)

$$\underline{Y}_i = \underline{A}_i = \hat{f}_{MA}(\underline{X}^*(i); \underline{w}(t, d^*)) + \underline{\varepsilon}_i^{(mus)} + \underline{\varepsilon}_i^{(mod)} \qquad 1 \le i \le N \qquad \text{(EQ 86 : MAM)}$$

$\underline{X}^*$ is the vector of discrete predictors (d^*, X_5, X_6, X_L), and $\underline{w}$ is the vector of supplied weather variables $(X_9, \ldots, X_{14})$. N is the number of distinct values for $\underline{X}$, i.e. the number of non-empty cells in a maximal partition of $\underset{\sim}{X} = \underset{\sim}{d}^* \times \underset{\sim}{X}_5 \times \underset{\sim}{X}_6 \times \underset{\sim}{X}_L$. The upper bound for N is $913 \times \left|\underset{\sim}{X}_5\right| \times \left|\underset{\sim}{X}_6\right| \times \left|\underset{\sim}{X}_L\right|$, but N is far less since there are many, many combinations of SIC, tariff code and load factor group for which there are no matching customers.

This is easily recast as a Uniresponse Atomic Model (UAM):

$$Y_i = \hat{f}_{UA}(t; \underline{X}^*(i); \underline{w}(t, d^*)) + \varepsilon_i^{(mus)} + \varepsilon_i^{(mod)} \qquad 1 \le i \le 48N \qquad \text{(EQ 87 : UAM)}$$

Note that there is a loss of information involved in aggregating each set of profiles with identical discrete predictors into one atomic (centroidal) profile; we retain information (viz the mean) about the central tendency of loads at each half hour, but lose infor-

1. Since if they have identical day index d^* then they have identical weather data $X_9, \ldots, X_{14}$.

mation about the spread of loads at each half hour. However, by extending the notion of a profile from ‘a vector of half-hourly loads’ to ‘a vector of half-hourly loads *and* a vector of half-hourly *squared* loads’, and extend the notion of adding two profiles so that the sum profile records the sums of the 48 constituent *squared* loads as well as the sums of the 48 constituent loads, important information about the spread of loads at each half hour is retained (since this is enough data to reconstruct the initial half-hourly standard deviations).

10.5 A General Model for Weather-Dependent Loads

10.5.1 The Effects of Weather on Load

The six weather variables $X_9, \ldots, X_{14}$ available (alternatively $\theta, w_w, w_h, w_c, w_r, w_s$, respectively temperature, windspeed, humidity, cloud cover, rainfall, and sunset time, collectively designated $\underline{w}$) are an adequate variable set to explain all effects of weather on load. However, the *spot* weather variables (the values of the supplied weather variables for a particular day and time of day) are not necessarily an adequate set of predictors for weather dependent loads observed at that time of day on that day. This is because the values of weather variables prior to that time point also have a lag effect on load. The discourse of this section would be applicable to residential customers as well as business customers (although the load-weather relationships of business and residential customers may vary substantially). Further reading concerning the ideas discussed in this section can be found in [15], [17], [21], [23], [24].

The most influential weather factor on load is temperature. Suppose we are interested in the load-weather relationship at a particular half-hourly time point $t = T$. Call T the *current* time (or ‘now’). Thermostatic heating (/cooling) devices like water heaters, space heaters (/space air conditioners, refrigerators) etc. are more likely to be drawing power when the current *indoor* temperature $\theta_T^{(I)}$ is low (/high). When load at time T is averaged over several days or customers, the particular on/off statuses of the various devices will be ‘blurred’ (smoothed), so that the mean load is expected to vary quite smoothly with $\theta_T^{(I)}$. Non-thermostatic heating or cooling devices (which are switched on or off by time switches or by hand) have aggregate loads expected to follow $\theta_T^{(I)}$ somewhat less closely, but nevertheless to follow it.

However the indoor temperature half an hour ago, an hour ago, or several hours ago, also have effects on the loads of heating and cooling devices: thermostatic devices may

take some time before reaching cut-off point (i.e. before they have heated or cooled to such a degree that their thermostat switches them off); they may take half an hour or longer before they switch back on again (this period is known as the dead band). Loads for non-thermostatic heating or cooling devices may be dependent on even older temperature readings (for example, a time-switched heating device generally changes its consumption pattern only when its on/off times are manually reset, which might be only once every few days or weeks; the sudden onset of a cold snap may well not manifest itself for a day or two).

However this is only half the story, since the indoor temperature $\theta^{(I)}$ (which is different for every customer) is not known: it is a function not only of the status of the various heating and cooling devices present, but of the current *ambient* (outdoor) temperature, and of the ambient temperature over the recent past. It is ambient temperature (θ) which is recorded in the databases. The thermal properties of buildings (which vary from customer to customer and are not known) are such that the changes in ambient temperature take some while to manifest themselves indoors. Thus the short-term *lag* temperatures (the ambient temperatures from half an hour ago to several hours ago) are important factors affecting load. Longer term lag temperatures may also have an effect: the temperatures a few feet under the ground, and the temperature of large bodies of water, are dependent on the ambient temperatures several days (even weeks) ago. Thus heat losses and gains through the floor may depend on the ambient temperature some time ago; water-heating devices draw cold water through underground pipes from reservoirs, and the temperature of the cold water can lag the ambient temperature by days or weeks.

The importance of these factors requires us to introduce lag temperature variables (which can be derived) with various lag periods, and/or exponential smooths of the ambient temperature with various exponents. An exponential smooth of θ_t has the form

$$\theta_t^{(\alpha)} = (1-\alpha)\,\theta_t + \alpha\theta_{t-1}^{(\alpha)} \qquad \text{(EQ 88)}$$

α is fixed between 0 and 1. When α is closer to 0, the exponentially smoothed temperature is ‘forgetful’ and represents average temperature over the recent past; when α is closer to 1, the smoothed temperature is more affected by older temperatures. Longer term exponential smooths could use daily average temperatures or daily peak temperatures instead of half-hourly temperatures.

Other weather variables affect loads more subtly. Humidity is a factor in predicting

air conditioning and cooling fan loads. High indoor temperatures may be unpleasant, but are considerably more so when indoor humidity is high. Indoor humidity may lag ambient humidity slightly, and lag or smoothed humidities could be employed. A derived variable combining temperature and humidity is also of possible interest. Whilst air conditioning is only a small part of UK domestic load, is can be a significant factor in UK commercial and business loads.

Windspeed may have a minor effect on air conditioning/cooling fan loads (opening a window on a breezy day is cheaper, if less effective, than air conditioning). However its principal load-weather effect is on heating loads. Firstly, cold draughts may affect indoor temperatures on cold, windy days. Secondly, the effect of wind on the exterior walls of buildings is to cool the walls (particularly so if the walls are wet). A derived variable combining temperature and windspeed could be of interest. Load is unlikely to lag windspeed by very much; a slightly lagged windspeed variable could be of use.

Cloud cover has an effect on lighting loads. In fact, the amount of natural illumination would be a more useful predictor of lighting loads, but was unfortunately unavailable in the data supplied. Natural illumination is principally a function of the time of day, the time of year, rainfall, the opacity of cloud cover, and the percentage cloud cover (of which only cloud opacity was unavailable in our data; rainfall is largely missing from the data supplied, but has only slight effect on natural illumination). A derived variable combining time of day, time of year and cloud cover, might be of interest in predicting lighting loads. Since people frequently leave light switches on when they ought not, some lag effect may be present.

Rainfall, as well as its (minor) effect on illumination, has (as noted above) an interaction effect with windspeed: wind cools wetter buildings more. The effect of rainfall on illumination is instantaneous, but the effect of rainfall on the wetness of building walls may last several hours, prompting the introduction of lag or smoothed rainfall variables. Rainfall can affect domestic electric clothes drying loads, unlikely to be a factor for businesses.

10.5.2 Temporal asymmetry of the Load-Weather Relationship

The effects of temperature on load are different when the temperature is rising than when the temperature is falling. For example: a rapid fall in ambient temperature from 13° C to 10° C is likely to place a large strain on (especially thermostatic) heating loads.

However a rapid rise in ambient temperature from $10°$ C to $13°$ C can result in a somewhat smaller relaxation on heating loads (see [23], [24]). If ΔY is the change in load for some profile over a small time period of Δt, and $\Delta\theta$ the change in ambient temperature over that period, we can term

$$\Delta Y(\Delta\theta) = -\Delta Y(-\Delta\theta) \quad \text{(EQ 89)}$$

the *temporal symmetry assumption*, a statement that the change in load resulting from a rise in the temperature is equal in magnitude (opposite in sign) to the change in load resulting from an equal temperature drop.

A static load-weather model (one which models current temperature in terms of only instantaneous weather variables) cannot account for temporal asymmetry effects. (FIGURE 8.(a)) illustrates a particularly simple static load-weather model (in which load is a function of θ only).

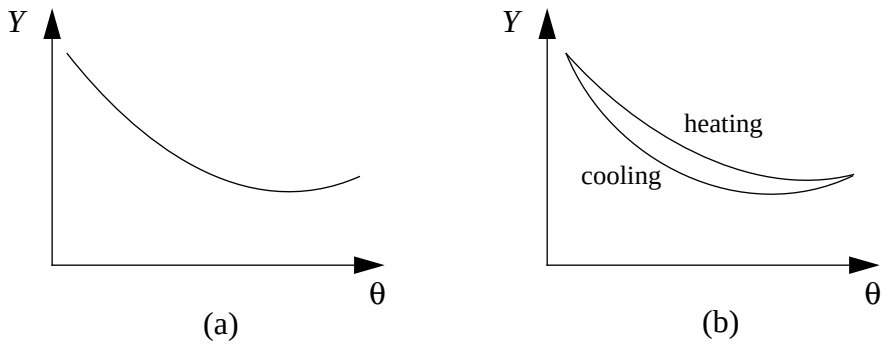


FIGURE 8.

A heating/cooling curve load-weather model derives a different model for when the temperature is rising (heating model) than when the temperature is falling (cooling model)—see (FIGURE 8.(b)). Such a model requires a categorical derived weather variable which decides which sub model is appropriate depending upon whether temperature is rising or falling.

A continuous derived variable which we might consider introducing is $\Delta_\tau\theta$, the change in temperature between the current time *t* and time $t - \tau$. We might consider various values for τ, since temperatures from various times in the past might have effects on load. Such variables enable a weather model to model temporal asymmetry (although the lag temperature variables suggested in section 10.5.1 are already capable of conveying this information indirectly).

10.5.3 Derived Weather Variables

Due to the various lag times of lag and Δ weather variables, and the various exponents which can be used in exponentially smoothed weather variables, there are practically limitless numbers of weather variables (derived from the spot weather variables) available for a load-weather model. Many of these are highly correlated, and it is usually necessary to select a parsimonious subset of weather variables (one which tries to avoid redundancy of variables). This might be performed using principal components analysis or related techniques; by heuristic choices; by trial and error; by use of an adaptive modelling technique which automatically tends to produce parsimonious models; or by some combination of these methods. (TABLE 4.) summarises some of the derived weather variables which are candidates for inclusion in a load-weather model.

TABLE 4.

Predictors (Derived, Weather)

Notation	Description	Comment
$\theta^{[\tau]}$	lag ambient temperature, lag τ half-hours	short, medium and long lags
$\theta^{(\alpha)}$	exponentially smoothed temperature, exponent α	short, medium and long term smooths
$\Delta_\tau\theta$	delta-temperature (temperature change), lag τ	short, medium and long term lags
$\theta^{\max}$	yesterday's peak temperature	
$\bar{\theta}$	yesterday's mean temperature	
$\theta^{\min}$	yesterday's minimum temperature	
$w_w^{[\tau]}, w_w^{(\alpha)}$	lag windspeed, smoothed windspeed	short term
$w_h^{[\tau]}, w_h^{(\alpha)}$	lag humidity, smoothed humidity	short term
$w_c^{[\tau]}, w_c^{(\alpha)}$	lag cloud cover, smoothed cloud cover	short term
$w_r^{(\alpha)}$	smoothed rainfall	medium term smooth; estimate of building/ground wetness
$w_s{}^*$	squared minutes from 18:00hrs GMT to sunset	peaks at midsummer and at midwinter; seasonality

$w_s{}^*$, the square of the time difference between sunset and 18:00hrs GMT, was suggested as a useful predictor of load by a researcher at the project's sponsoring organisation. It can be thought of as a measure of the extremeness of season (as it peaks midsummer and midwinter), and is dubbed 'seasonality'.

Compound variables suggested in section 10.5.1 (combinations of temperature and windspeed; temperature, rainfall and windspeed; temperature and humidity; cloud cover, time of day, day of year) representing particular concepts (respectively presence of draughts; wind chill; discomfort; natural illumination) might be simple low order polynomials in the constituent variables or non-linear terms suggested by theory. However, a non-parametric method like MARS (section 6.5) would have the advantage that such concepts can be modelled automatically (though not necessarily explicitly), since interactions terms between any number of predictors are considered for inclusion in the model.

10.5.4 Derived Temporal Variables

Reconsider the task specification: to analyse load profiles with regard to their dependencies on weather, day type, month, and known customer details. The weather data is recorded at hourly intervals: it is a simple task to recover reasonable interpolations so that estimated weather variables are available for each half-hour (see section 11.2.1). Since the combined weather variables vary from hour to hour and day to day in a chaotic (i.e. dynamic non-linear) fashion, and have no weekly period, it is not viable to replace d^*, the day index, with (d, m), the day type and month index, in any part of the model which predicts the effect of weather on load. If day index were removed from the model, then daily weather statistics would have to be aggregated (for example, all January Mondays clumped together), despite the fact that weather has no weekly periodicity. A day-of-year variable (which records how many days have passed since January 1st, rather than from the start of the study period like d^*) may be a more suitable variable for the weather model (see 11.3.2).

On the other hand, the non-weather dependent component of load is highly dependent on the day of the week, but tends to change only slowly from week to week. Thus it is highly appropriate to scrap d^* in favour of (d, m) for a weather-free load model. This suggests a compound model comprising one model for weather-dependent load and another for weather-independent load.

There is also a case for considering the inclusion of other derived temporal variables in the weather model. For example, a sinusoid with period 24 hours peaking at noon (with its trough at midnight), and a sinusoid with period 1 year peaking on midsummer's day (with its trough at midwinter's day), are derivable from (respectively) t and d^*, but

provide the time-of-day/time-of-year information in a way which may be more useful to a particular regression model. Note that sunset time w_s and seasonality w_s* can be viewed as derived temporal variables themselves, since they are determinable accurately from the time of day, time of year, and the latitude/longitude (which are fixed in this model).

10.5.5 Weather Decomposition

To derive a compound weather/non-weather model, it is desirable to find some initial means of decomposing load into *weather-dependent* and *weather-independent* components. Once the weather-dependent loads have been modelled, we can recover *deweathered load profiles* (hypothetical *weather-free* load profiles) by subtracting (or otherwise removing) the loads predicted by the weather model from the original data. Deweathered profiles can be thought of as our expectation of what the profiles would have looked like *did weather not exist*. Depending on the deweathering methodology, weather 'not existing' can be interpreted as weather being always exactly 'typical' (and static), or weather being always such that no weather-dependent electrical loads are present. The latter is rather idealistic, since loads such as lighting, water heating (other than for central heating) and refrigeration are present every day of the year, but are mildly weather dependent (not just seasonally dependent).

The initial decomposition of loads into weather-dependent and -independent components can be performed variously. These components are somewhat nominal since no perfect separation of these two components is possible. Two basic approaches to weather composition have been seen in chapter 9:

1. Identification of No-Weather Days. Load profiles for days on which weather has very little effect on load are aggregated to obtain estimates for weather-independent loads. Subtraction of the estimated weather-independent load from the total load produces an estimate for the weather-dependent load (see section 9.5.3).
2. Smoothing and Averaging. Profiles are smoothed or averaged over a variety of weather conditions. The smoothed or averaged loads are assumed weather-independent; again, these are subtracted from total load to obtain a weather-dependent load.

Many researchers have additionally sought a *trend* component (prior to deweathering), which models daily peak load or daily total energy as a smooth function of month index,

week index, or day index. This is to distinguish changes due to annual weather periodicity (and long term load growth), from changes due to hourly and daily weather fluctuation.

Note that whilst the weather-independent load is assumed to have no dependency on the weather variables, the weather-dependent load is certainly *not* assumed to have no dependency on *non*-weather variables. Thus once a weather-dependent component has been modelled, the *residuals* from that model are assumed dependent on the non-weather variables; the weather model residuals are combined with the weather-independent component to estimate the weather-free (deweathered) load; it is this weather-free load (and not the weather-independent load component) which is subsequently modelled using the non-weather variables.

10.5.6 Atomic Weather Model

Introduce the derived predictor d (day type), in addition to (rather than in place of) the day index d^*. Although weather does not have a weekly period, we may anticipate that the *effect* of weather on daily load has some dependency on the day of the week, and that day type may be relevant to any model for weather-dependent load. Suppose that Y_i^W are the weather dependent components of the $1 \le i \le 48N$ loads, produced by one of the decomposition techniques in section 10.5.5. Further, suppose that the data have been aggregated into atoms as described in section 10.4. Note that the atoms arising from discrete predictor space $\underset{\sim}{X}^W = \underset{\sim}{d}^* \times \underset{\sim}{d} \times \underset{\sim}{X}_5 \times \underset{\sim}{X}_6 \times \underset{\sim}{X}_L$ do not differ from those arising from discrete predictor space $\underset{\sim}{X}^* = \underset{\sim}{d}^* \times \underset{\sim}{X}_5 \times \underset{\sim}{X}_6 \times \underset{\sim}{X}_L$, since the value of d^* dictates the value of d.

Incorporating some appropriate subset of (*i*) the spot weather variables $\underline{w}$ and (*ii*) the various lag, delta and smoothed weather variables and lag-load variables introduced in section 10.5.3, into a weather vector $\underline{W}$; incorporating the day type d into $\underline{X}^*$; and denoting weather-dependent loads $Y_i^W; (1 \le i \le 48 \times N_W)$, we can adapt the uniresponse atomic model of (EQ 87 : UAM) into an Atomic Weather Model (EQ 90 : AWM)

$$Y^W{}_i = \hat{f}_{AW}(t;\underline{X}^W(i);\underline{W}(t, d^*)) + \varepsilon_i^{(mus)} + \varepsilon_i^{(mod)} \qquad 1 \le i \le 48N_W \quad \text{(EQ 90 : AWM)}$$

Here N_w (the number of profiles presented to the weather model) is the number of days in the study times the number of atoms arising from predictors $\underset{\sim}{X}_5 \times \underset{\sim}{X}_6 \times \underset{\sim}{X}_L$.

A point arises concerning the proper separation of the load-weather model AWM and

the subsequent model for deweathered data (which will be described in section 11.1). We would prefer that the load-weather model models precisely just that, and does not model the effects of the non-weather variables $\underline{X}^W$. The presence of the day index d^* in the load-weather model need not particularly worry us provided that model terms including d^* vary only slowly with d^*, since this just facilitates the computation of gradual trends. However, the presence of d, X_5, X_6, X_L facilitate the modelling of non-weather effects on load which are better left to the weather-free model. The reason why we might wish to retain d, X_5, X_6, X_L in the weather model is that the *way* in which weather affects load is liable to be different for different day-types, SIC code classes, and tariff code classes, and may be different for customers with significantly different load factors. The weather modelling function $\hat{f}_{AW}$ should be determined in a manner which circumvents this danger (see further discussion in 11.2.4).

10.6 A General Model for Weather-Free Loads

Now suppose that $\hat{f}_{AW}(t;\underline{X}^W(i);\underline{W}(t, d^*))$, for $1 \le i \le 48N_W$, an estimated function for weather-dependent load in terms of time of day, weather variables and derived weather variables, and discrete predictors, has been obtained. Call it $\hat{f}_{AW}$ for brevity. Recall that an initial estimate of *weather-independent* load, obtained by smoothing/averaging, was required to build this model. If $Y(t)$ are the initial loads (the raw data), $Y^W(t)$ are the weather-dependent loads, and $Y^I(t)$ are the weather independent loads, then $Y^W(t) - \hat{f}_{AW}$, the weather residuals, are added to the weather-independent loads to obtain the weather free loads

$$Y^F(t) = Y^I(t) + [Y^W(t) - \hat{f}_{AW}] = Y(t) - \hat{f}_{AW} \qquad \text{(EQ 91)}$$

Section 10.5.4 suggests that in modelling weather-free loads it is appropriate to replace d^* by d and m (i.e. day index with day type and month). Since we can now drop the weather variables $\underline{W}$, we can adapt the multi-response atomic model of (EQ 86 : MAM) to the Atomic weather-Free Model (AFM):

$$\underline{Y}_i^F = \hat{f}_{AF}(\underline{X}^F(i)) + \underline{\varepsilon}_i^{(mus)} + \underline{\varepsilon}_i^{(mod)} \qquad 1 \le i \le N_F \qquad \text{(EQ 92 : AFM)}$$

Now $\underline{X}^F = (d, m, X_5, X_6, X_L)$, and N_F is no more than 7 (day types) times 12 (months) times $|\underset{\sim}{X}_5| \times |\underset{\sim}{X}_6| \times |\underset{\sim}{X}_L|$ (and is much less, since most SIC/tariff/load factor combinations are empty).

Chapter 11 — Weather Model for Monthly Billed Customer Database

11.1 Weather Decomposition

The no-weather day approach to weather decomposition (section 10.5.5) was not deemed particularly appropriate for UK business loads; it would be hard to determine conditions under which UK businesses are unlikely to be drawing any weather dependent load, and there appears no major advantage of a no-weather day approach over a smoothing/aggregation approach.

A simple aggregation approach was selected. The weather models were calculated on the 1995/6 and 1996/7 databases, i.e. 24 months of data (with occasional days missing). For a given model, some non-empty subset of the categorical predictors $\{d, X_5, X_6, X_L\}$ is selected. Denote it X^I, the set of categorical predictors used to determine the weather-independent loads. Certainly, the atoms arising from the predictor subspace $\underset{\sim}{d} \times \underset{\sim}{X}_5 \times \underset{\sim}{X}_6 \times \underset{\sim}{X}_L$ (i.e. the centroidal profiles for a given set of values for day of the week, SIC code, tariff code and load factor group, the *weather-independent atoms*) are reasonably weather-independent, since they cover all types of weather (24 months). They are only weather dependent in so far as the 24 months in question may have been atypical meteorologically, but we can do no better with the data available. This is equally true if we only use some smaller subset of the categorical predictors $\{\underset{\sim}{d}, \underset{\sim}{X}_5, \underset{\sim}{X}_6, \underset{\sim}{X}_L\}$. If we discount special day types, so that day-type can take only seven values, then each atom is guaranteed to represent at least around 100 constituent profiles, so we might hope that the 48 load values in an atomic profile are reasonable estimates of the underlying 'true' weather-independent loads.

Denoting these weather-independent atoms $\underline{Y}^I_{i'}$, or recast as a single response, $Y^I_{i'}(t)$, the *weather-dependent loads* are recovered from the equation

$$Y^W_i(t) \;=\; Y_i(t) - Y^I_{i'}(t) \qquad 1 \le i \le 48N_W \qquad \text{(EQ 93)}$$

where $Y^I_{i'}(t)$ is the weather-independent atom appropriate to the ith cell in a maximal partition of the domain of $\underset{\sim}{X}^W$ (see atomic weather model, section 10.5.6). $Y_i(t)$ are the atoms for the raw data, using the same partition.

Note that the weather dependent loads take both positive and negative values. Where weather has a locally above-average effect on consumption, $Y^W_i(t)$ will be positive; be-

low-average effects on consumption give negative $Y_i^W(t)$.

11.2 MARS for Load/Weather Modelling

11.2.1 Calculating Derived Weather Variables and Interpolating Weather Data on Half Hours

Note that the weather data provided (section 10.1) is recorded hourly, but the load profile data is recorded half-hourly. It would be possible to throw away every second recorded load, retaining only those recorded on the hour, so that there is a set of weather predictors available for every load recorded. However, it is possible to interpolate for the missing half-hour weather data, in order not to waste half of the recorded load data. It was decided to use cubic interpolations for the half-hour weather data, interpolating at the two closest recorded data points on either side, as in (FIGURE 9.).

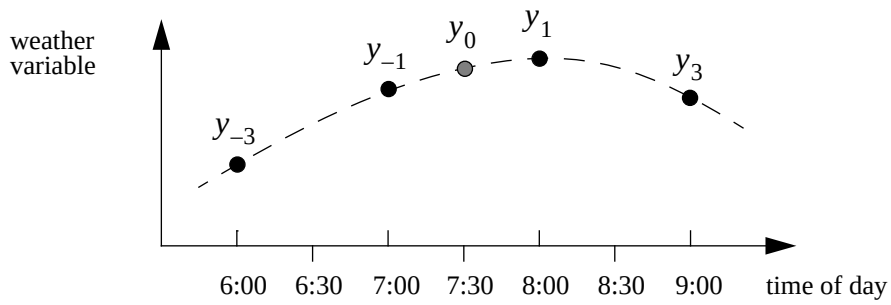


FIGURE 9.

It is easily determined by solution of simultaneous equations that the unique cubic polynomial which interpolates known responses y_{-3}, y_{-1}, y_1, y_3 at ordinates $t = -3, -1, 1, 3$ respectively, takes a value $y_0 = \frac{9}{16}(y_{-1} + y_1) - \frac{1}{16}(y_{-3} + y_3)$ at $t = 0$ (t measures half hours either side of the missing value, e.g. 7:30 in (FIGURE 9.)). At the beginning and end of the time period under study, where one or more of y_{-3}, y_{-1}, y_1, y_3 fell outside the recorded data they were replaced by the nearest recorded values. The interpolations are performed with each of θ, w_h, w_w, w_c, and w_r replacing y in the above. Where, on occasion, the weather statistic produced by this interpolation falls outside of the meaningful range (say, more than 8 oktas of cloud cover, or less than 0% humidity) it is automatically replaced by the nearest meaningful value.

The implementation of the selected weather model also calculates exponential smooths, and lag and delta variables (with various exponents and lags supplied by the

user), as described in section 10.5.3, using the above interpolations at the half hour ordinates. Again, where weather data are required which fall before the beginning of the study, they are replaced with the first value in the study period, which is only a poor estimate when the lags are very long or the smooths very long term.

11.2.2 Continuous Part of the Weather-Dependent Model

Prior to any analysis, it is assumed that profiles have first been normalised so that each customer has the same total energy consumption over the study period, as per section 10.2.

Recall the atomic weather model of (EQ 90 : AWM) again:

$$Y^W{}_i = \hat{f}_{AW}(t;\underline{X}^W(i);\underline{W}(t, d^*)) + \varepsilon_i^{(mus)} + \varepsilon_i^{(mod)} \qquad 1 \le i \le 48N_W \qquad \text{(EQ 94 : AWM)}$$

The predictors t and d^* are ordered discrete variables, but will be treated as continuous predictors. Except for the presence of d, X_5, $X_6$and X_L, all predictors are (considered as) continuous, as is the response $Y(t)$. Thus we have a regular multivariate regression task with four additional discrete predictors.

Temporarily ignoring these discrete predictors, the task is to model the response in terms of t, d^* and $\underline{W}$, the latter comprising several, possibly dozens of, weather, derived weather, and derived temporal variables. Most load studies have greatly simplified the task, by only using a handful of the candidate predictors, and by assuming a simple regression function, usually linear in most predictors, sometimes with a few higher order polynomial terms, and sometimes with some cross-terms (such as $\theta \cdot w_w$, $\theta \cdot w_h$). However the best models of load-weather dependence may not be low-order polynomial models, and such models may miss features in the data; moreover, derived variables of the types in section 10.5.3 are known to be important, but because there are so many candidates, and so many mutual correlations between them, inclusion of more than a few of them can lead to sparse data (due to the high dimensionality of predictor space) and ill-conditioned regression equations (due to correlated predictors), hence poor fit.

It was decided to perform a multidimensional non-parametric regression of load on a large number of weather variables $\underline{W}$ using Friedman's Multivariate Adaptive Regression Spline (MARS) data mining regression algorithm (outlined in section 6.5, see [9],[10]). The MARS model has many advantages both over parametric models and over non-adaptive non-parametric models in this regression task.

Parametric models require a preordained form for the fitting function; in order to capture all aspects of load-weather relationship, such a form would be highly complex, and require some protocol for its design. Furthermore the high dimensionality of the predictor space $\underset{\sim}{W} \times \underset{\sim}{t} \times \underset{\sim}{d}^*$ inevitably dictates that data will be sparse. Non-adaptive non-parametric models (for instance, a $(|W| + 2)$ -variate smoothing or product spline model) suffer from the same problem of data sparsity, and also from the problem of correlated predictors.

An interaction spline model (see section 6.5.1) could avoid the problem of data sparsity and partly avoid the problem of correlated attributes: each additive component of the model can be limited in dimension (say, allow components of dimension one, two, three and perhaps four); and strongly correlated predictors can be prohibited from appearing in the same additive term. We need only include terms involving more than one predictor (*interaction terms*) where the two or more predictors are known or suspected to interactively affect response.

The chief obstacle to using interaction splines is the initial choice of additive terms—the problem of selecting the sets of interacting predictors. One might simply propose including a term for every set of two, three or four predictors suspected to interactively affect response. However, there are vary many such sets, many of which will be redundant in the presence of the others. Consequently such a scheme is liable to overfit the data (fit noise not trend); also, the presence of similar additive terms could lead to ill conditioning of the regression equations.

The MARS model has most of the advantages of the interaction spline model: no preordained fitting function is required; and (referring to the ANOVA decomposition of the model (EQ 57)), additive terms with a certain number of predictors can be prohibited from appearing in the model. Moreover, the total number of additive terms appearing in the final model can also be limited, and smoothness parameters varied, so that the danger of fitting noise rather than trend can be kept at bay.

The crucial advantage of MARS over interaction splines is that the selection of interaction terms is performed automatically in MARS models. Certain predictors can be excluded a priori from appearing in the same interaction term (usually those which are known to be correlated), but otherwise any set of predictors can appear in the same interaction term (subject to the specified maximum number of interacting predictors). Moreover the pruning part of the MARS algorithm removes from the model those discovered terms which contribute least to goodness of model fit, so that interaction terms

in the final model are liable to represent real interactions rather than noise.

The interpretability of MARS models is good. Firstly, constraints on the dimensionality of interactions and final number of model terms check the model complexity. Secondly, many terms in the final model have only one or two predictors, admitting curve and surface plots for those components. Thirdly, higher level interactions can easily be *sliced* to admit curve and surface plots. Slicing an interaction term involves fixing the values of all but one or two predictors (often so as to take their mean or median value), and plotting a curve/surface using the remaining one/two predictor(s) (and the response). The MARS code (in Fortran) is freely available and includes routines to perform slicing.

Note that the data (EQ 94 : AWM) are in atomic form; thus each data point presented to MARS has a response which is aggregated from several underlying profiles. Accordingly it is necessary to normalise and weight each datum. Each response Y^W_i is divided by the number of profiles summed thereat before presentation to MARS. In this way, if one tariff code matches twice as many customers as another tariff code, for instance, it will not result in twice as high a response being presented to the MARS model for observations matching that tariff code. However, each datum is also *weighted* by the number of profiles aggregated into that datum. MARS allows weights for each data point in its lack-of-fit function; with such weights w_i, the GCV equation of (EQ 59) is modified to

$$E(\hat{f}_p) = \frac{\frac{1}{N}\sum_{i=1}^{N} w_i \left[Y_i - f_p(\underline{X}_i) \right]^2}{\left[1 - \frac{C(P)}{N} \right]^2 \sum_{i=1}^{N} w_i} \qquad \text{(EQ 95)}$$

Thus atom/day combinations with fewer representatives in the database are accorded proportionately less importance.

11.2.3 Discrete Part of the Weather-Dependent Model

Now consider how best to employ the categorical predictors into the load-weather model. Since the weather dependent loads $Y^W_i(t)$ are obtained from the raw data by subtracting the average profiles (over the whole study period) for each distinct combination of values of d, X_5, X_6, and X_L, we can assume that the principal effects of these predictors on the weather dependent loads has already been removed. Thus the danger that the

load-weather model might attempt to model the effects of non-weather variables on load (rather than the *interaction effects* of weather *and* non-weather variables on load) is averted.

To derive a separate MARS model for each weather-dependent atom of $\underset{\sim}{d} \times \underset{\sim}{X}_5 \times \underset{\sim}{X}_6 \times \underset{\sim}{X}_L$ is not a viable option: not only are an unwieldy number of separate load-weather models produced, but each would be built using only a fraction of the data. Constructing separate models for work-day (Monday to Friday), Saturday, and Sunday loads would, however, be viable.

Otherwise, the binarising of categorical variables, subsequently treated as continuous variables (see section 6.6), is an option, though it introduces many more variables. In order to limit the number of new binary predictors in the model, tariff codes (or SIC codes) deemed sufficiently similar could be lumped together into compound tariff (or SIC) codes prior to binarisation. This could be achieved by prior clustering, as in Chapter 13.

However, as noted in section 6.6.2, MARS includes its own mechanism for the inclusion of categorical variables, in which they are treated analogously to and simultaneously with the continuous variables, yet avoiding (to an extent) the problems of these other schemes, and this was the method used.

Finally, a database of 'weather-free' loads is constructed, by subtracting the profiles arising from the weather model from the original profiles (see section 11.2.5). Note that the weighting scheme described in 11.2.2 ensures that where a tariff code or an SIC code has fewer representatives in the database, it is down-weighted proportionally in the MARS model. As before, it is simply necessary to weight each datum presented to the MARS model proportionally to the number of profiles aggregated into that datum.

11.2.4 Limitations on MARS Categorical Variables

It was decided to limit the categorical variables that are allowed categorical predictors in the MARS model in two ways. Firstly, only categorical variables in $\underset{\sim}{X}^I$ (the variable set used to determine the weather independent atoms, see section 11.1) are acceptable as categorical predictors. Thus if the weather independent profiles are determined by summing profiles with identical values for d and X_6 (only), for instance, then neither X_5 nor X_L are acceptable as categorical predictors in the weather model. The justification for this is that (as noted in 10.5.6) the weather dependent model is intended to model only the effects of weather, not customer attributes/day of the week; the categor-

ical variables (customer attributes and day of the week) are included as predictors because load may have a different dependency on weather for different values of these categorical variables. However, if we allow only those variables in $\underset{\sim}{X}^I$ as predictors in MARS, then their combined effect on load has already been removed from the loads presented to MARS (since the weather dependent loads were created by subtracting the relevant atomic profiles arising from all variables in $\underset{\sim}{X}^I$ from the original loads, and are statistically independent of any variable or product of variables in $\underset{\sim}{X}^I$). The effect of these variables' *interactions* with the continuous (weather) variables, however, are not removed by the weather separation process, and so we allow MARS to attempt to model such interaction terms.

The second restriction on categorical variables allowed in MARS is practical rather than theoretical, namely that possibly d, and at most one other variable from $\underset{\sim}{X}^I$, are used as predictors in any one model. Whilst allowing d (day of the week) into the model does not increase the total number of observations (data points) presented to the MARS model (since there are already 48 half-hourly data points for every day in the model whether we include d or not), for each variable $X \in \{X_5, X_6, X_L\}$ we allow into the model we significantly increase the number of data points; for each day and half hour in the model, we need a separate observation for each combination of values of X_5, X_6 and X_L occurring in the data. In the final implementation, both computation times and memory requirements proved impractical when this restriction was not made.

11.2.5 Deweathering Whole Load Using a Computed Mars Model

For each original (normalised) profile in the loads database, we wish to construct a 'weather-free' profile, to arrive at a set of profiles which (as far as possible) has had the effects of weather removed. This can be achieved by subtracting, from the original profile, the profile predicted by the weather model (EQ 91). More exactly, for each day index d^* and customer c, look up or calculate the values for the weather variables and derived weather variables, and the temporal variables and derived temporal variables, for that day, and look up the non-weather variables X_5, X_6, X_L. Feed these into the computed weather model $\hat{f}_{AW}(t;\underline{X}^W(i);\underline{W}(t, d^*))$ (i is the index of the weather-dependent atom that the original profile $\bar{Y}$ belongs to) for each of 48 values of t to determine the half-hourly profile of the effects of weather on load $\underline{f_{AW}}$. Then by subtracting this profile from the original, we obtain a weather-free profile

$$\underline{Y^{WF}} = \underline{Y} - \underline{f_{AW}} \quad \text{(EQ 96)}$$

Note that due to the way in which the weather-dependent profiles are normalised prior to being presented to the MARS model (see end of section 11.2.2), the weather model profile $\underline{f_{AW}}$ is on the same scale as the original profile $\underline{Y}$.

The piecewise cubic version of the MARS model is subtracted in the deweathering scheme, rather than the piecewise linear model; the piecewise cubic model is obviously smoother, and generally has lower cross-validation errors.

11.3 Results and Interpretation of the MARS Load/Weather Models

11.3.1 Data Selection and Cleansing

Firstly, recall that rainfall w_r was recorded only for October 1994 through March 1995 in the available weather data, none of which covers the 1995/6 and 1996/7 profiles databases on which the principal weather study was performed. However, a number of exploratory weather models were built using the 1994/5 load data, for which (excepting April) rainfall data was supplied. In all these exploratory studies, neither rainfall nor any lag/delta/smoothed variables derived from rainfall were ever selected for inclusion by the MARS modelling algorithm. It is therefore reasonable to conclude that the availability of rainfall data in the 1995/6/7 study period would probably not have made a significant impact on the resulting weather models, and rainfall is not considered in the results.

In fact, due to the discrepancies where the 1994/5 data and 1995/6 data overlap, it was decided to omit the 1994/5 data from consideration completely in the weather modelling, since it would appear that customers in the former database have undergone a different normalisation procedure from those in the latter two. Note that including the six months of non-overlapping data from the 1994/5 database would have introduced a pro-winter bias into the weather-independent profiles (which, recall, are supposed to be averaged over all types of weather) which would have to have been countered by some system of weighting.

The 1995/6 and 1996/7 provide two years of data, which is not really enough data on which to build a definitive weather model (since two years' worth of data might not contain all possible types of weather and weather changes for the geographical area modelled; and even so may not contain enough profiles, for any given set of prevailing weather conditions, to be representative of those conditions); but it is certainly enough

data with which to build an interesting and informative weather model, and with which to test and fine tune the presented methodology.

Accordingly, a loads database for 1995/6/7 (running 24 months from April 1995) was built from the raw data supplied, having first excluded from consideration various special days and holidays. These were determined by visual inspection of coloured 3 dimensional plots of the 1995/6 and 1996/7 databases, as described in section 12.5 (see (COLOUR FIGURE 10.), (COLOUR FIGURE 11.)). In fact, all bank holidays and extended holidays had already been removed from the former database, but not from the 1996/7 database. All UK bank holidays were removed, and also the whole of the Easter long weekend (Good Friday through Easter Monday) and Christmas/New Year period. When to start and end the Christmas/New Year period is a matter of judgement (as it depends on whereabouts weekends fall), which is why the visual inspection of the coloured plots is helpful. In fact, December 18th 1995 to January 1st 1996 inclusive, and December 22nd 1995 to January 3rd 1996 inclusive, were excluded (the corresponding 1994/5 figures being December 24th 1994 to January 2nd 1995 inclusive).

There were a total of 431 customers present in this data, and a total of 248 614 load profiles (11 933 472 weather-dependent load readings).

A mechanism for the automatic removal of *solecisms* (suspicious spikes in the data sustained for only one half-hourly reading, see section 10.1) was built into the code which builds the binary databases from the raw (flat-file) load data. Whilst the mechanism used is rather crude, and may in some cases class genuine data as erroneous (or wrongly class a short term dip in load as an outage or blown circuit), it was deemed necessary to remove at least some solecisms in order to reduce noise (and also to calculate accurate load factors), and it seems likely that most of the spikes removed were indeed genuinely erroneous data (and most of the dips genuinely due to special circumstances). A datum l_0 was classified as a *peak solecism* if the raw load reading immediately preceding it l_{-1} and the raw load reading immediately following it l_1 were such that

$$min\left(\frac{l_0}{l_{-1}}, \frac{l_0}{l_1}\right) > 4.0 \qquad \text{(EQ 97)}$$

and $l_0 > 18.0$, whilst it was considered a *trough solecism* if the data immediately preceding and following it were such that

$$min\left(\frac{l_{-1}}{l_0}, \frac{l_1}{l_0}\right) > 8.0 \qquad \text{(EQ 98)}$$

and $l_0 > 15.0$. These various thresholds were chosen intuitively after inspection of the raw data, with the aim of discriminating best between spikes which were genuine and spikes which were likely not to be genuine, or due to brief loss of power. The thresholds $l_0 > 18.0$ and $l_0 > 15.0$ are intended to account for the fact that the ratios in (EQ 97) and (EQ 98) are more likely to occur as genuine artifacts when load levels are low. In the event that either type of solecism was detected, l_0 was replaced by $(l_{-1} + l_1)/2$.

The final stage in creating the database was to normalise each customer so as to make their average half-hourly load equal to 100 units (see section 10.2). In what follows, ‘original database’ means one with special days removed, solecisms smoothed over, and then normalised, as above, not the original flat file data.

11.3.2 Exploratory Runs and Conclusions Drawn Therefrom

During the software’s development, a vast number of runs of the MARS algorithm, and of the routines necessary to present data to and extract data from it, were performed, usually on small subsets of the data; while results from these debugging and exploratory runs are not recorded here, various observations were made which informed the format of the experiments which are presented.

As has been noted, rainfall was excluded from further consideration (though there is hardly any rainfall data recorded for the 1995/6/7 database in any case). It was noted that cross-validation runs, to determine the MARS smoothing parameter d (section 6.5.3) (using the inbuilt MARS cross-validation scheme, see section 11.3.8) took far too long for many to be performed often, except when the problem size is greatly reduced. The total complexity can be reduced primarily by restricting the number of predictor variables, particularly categorical customer variables (see section 11.2.4). (Of course, the complexity is also reduced by presenting less data, i.e. less dates and less customers, but it was decided to perform all the final runs here presented on the full 1995/6/7 date and customer sets, to allow for fair comparisons.) Another way to lower complexity is to limit the number of initial basis functions selected prior to pruning (section 6.5.3), though this often results in candidate variables being excluded from the model which are found to be of importance when the initial number of basis functions is increased.

Even when cross-validation is not used, runs can still take prohibitively long, and it

is necessary to restrict the number of candidate predictor variables (and be prudent with the number of pre-pruning basis functions) as above. MARS also provides a ‘speed’ parameter. Larger values progressively sacrifice optimization thoroughness in the selection of basis functions, to gain computational speed [11]. This usually results in a marked decrease in computing time with little or no effect on the resulting approximation’s accuracy. All the presented results use the highest (fastest) value for this parameter, due to the high size of the problem (many observations, many variables).

Upon observing the long computation times on exploratory runs, it was also decided to restrict the types of interactions allowable between predictor variables, which considerably reduces complexity in some cases. The scheme chosen was based partially on the exploratory runs, and partially on heuristic choices informed by the known interaction effects of weather variables on load. The following types of interaction were allowed/ prohibited (and these apply to all lag/delta/smoothed versions of the variables referred to, not only the spot variables):

- cloud can only interact with “insolation”, rainfall and temporal variables[1]
- “insolation” can only interact with cloud, rainfall and temporal variables
- “darkness” can only interact with rainfall and temporal variables
- windspeed can only interact with temperature-based variables[2], humidity, rainfall, “discomfort” and temporal variables
- “discomfort” can interact with rainfall, windspeed and temporal variables
- “chill” can interact with rainfall and temporal variables

The variables insolation, darkness, chill and discomfort are derived variables introduced in sections 11.3.3 and 11.3.10.

Additionally, MARS was prohibited from considering interactions involving any lags, deltas and smooths of the same basic (spot) variable (with each other, or with the spot variable).

Note that a day-of-year variable (days since January 1st), was used rather than day index d^* (days since beginning of study period). This is because day index allows the

1. The temporal variables are time of day, sunset time, seasonality (see 10.5.3), and the derived variables summer, spring, noon and evening (see 11.3.4).

2. The temperature based variables are θ, θ^{max}, $\bar{\theta}$ and θ^{min} (temperature now, peak yesterday, mean yesterday and trough yesterday).

weather model to differentiate between a particular time of year in one year and the same time of year in another year, regardless of whether the weather differs between the two days in question, and so is not as relevant to a weather model as day-of-year.

A pure random noise variable (with a uniform distribution on [0, 1]) was also included in all the experiments that follow; this was used as a safeguard against overfitting: if at any stage in the MARS algorithm pure noise is selected in preference to the meaningful variables, it is a very strong indication that the model is being overfitted. Whilst in some exploratory runs where smaller customer and date sets were used and where few meaningful variables were available for selection the noise variable was indeed selected, in the full size trials presented here noise was never selected.

Any other MARS parameters were left at their default values (see [9],[10]), the MARS algorithm being deemed too expensive to determine their optimal values by various series of cross-validation runs.

11.3.3 Derived Variables Describing Illumination

An important observation made during the exploratory runs is that day-of-year was found to be a very important variable, more so than had been anticipated. Since day-of-year is not a weather variable, its chief justification for inclusion in the weather model is that the amount of natural illumination at any moment is dependent on the time of year. Of course, the time of year affects load very strongly, because the time of year affects the weather so strongly, but since we have the principal weather variables recorded in their own right, that does not in itself justify the inclusion of time-of-year variables in a weather model. By including time of year in the weather model, any effects that the time of year has on load shape *that are unrelated to weather* are also liable to be modelled, whereas this ideally ought be left to the weather-free (clustering) model.

If time-of-year variables were to be excluded, it would be especially desirable to include a natural illumination variable (which, as discussed, was not available in the weather data). In fact, the amount of natural illumination available *above the atmosphere* at any moment and any place, can be calculated from the date, the time of day, and the (fixed for our model) latitude and longitude. The SOLPOS library of functions, freely available from the (United States) National Renewable Energy Laboratory (NREL) [34], were used to calculate this quantity, known as *extraterrestrial insolation*. However, it is the surface insolation (i.e. below the clouds) which we would really like to know, and

this is affected by such factors as atmospheric pressure, pollutants, cloud cover and cloud density. A very crude estimate for this quantity was calculated based on the only one of these factors available, cloud cover. The simplifying model assumes that all of the extra-terrestrial insolation will reach the surface, except for that which hits a cloud; and that a fixed fraction (the *cloud transparency*) of the latter will penetrate to the surface regardless. Thus our estimate for surface insolation $\hat{I}_s$ is

$$\hat{I}_s = I_e ([1 - CF] + [CF \times CT]) \quad \text{(EQ 99)}$$

where I_e is the extraterrestrial insolation, $0 \le CF \le 1$ is the fraction of the sky covered by cloud and $0 \le CT \le 1$ is the cloud transparency parameter. In fact, the parameter used was named *darkness* D, where

$$D = I_{max} - \hat{I}_s \quad \text{(EQ 100)}$$

and I_{max} is the greatest value that can be taken by I_e. This is an estimate of effective darkness, which avoids confusion with extraterrestrial insolation I_e, which we will just call *insolation*.

The SOLPOS library was also used to calculate daily sunset times; these calculated sunset times differ (though only very slightly) from true sunset times for a particular location, as they do not take elevation, refraction, and other complicating factors into account; however the true sunset times for the geographical location in question were not available.

11.3.4 More Derived Temporal Variables

Four derived temporal variables were also thought to be worth adding to the model after some experimentation. These are sinusoids with various phases and periods (and with amplitudes arbitrarily equal to 1). The variables *summer* and *spring* have period 1 year (reckoned as 365.25 days), and peak respectively on midsummer's day and on the vernal equinox. The variables *noon* and *evening* have period 24 hours and peak respectively at 12:00 hours GMT and at 18:00 hours GMT. They represent closeness to summer, spring, noon and evening respectively.

As noted in section 10.5.4, such variables represent a different way of presenting time of day and time of year information to MARS which may be more helpful than the current hour or day of year. They also allow the periodicity of days and year to be rep-

resented in the model (whereas time of day and day of year each have a discontinuity that does not correspond to a similar discontinuity in nature). Note that linear combinations of *spring* and *summer* will be sinusoids with period 1 year, but with various phases. Thus MARS is able to indirectly model closeness to any particular time of year by using these variables in separate additive terms. An analogous point holds for *noon* and *evening* and times of day.

11.3.5 Notes on MARS Output

Recall that lack-of-fit function used by mars is GCV, defined in (EQ 59) (or modified by weights, (EQ 95)). MARS produces a measure of relative variable importance for each variable based on GCV; it is calculated for each variable in turn by removing all terms in the MARS model involving that variable, and recording by how much the lack-of-fit score GCV increases as a result. These figures are normalised into percentage figures (by dividing by the greatest calculated GCV increase, taking square the root, and multiplying by 100%; square roots are taken since GCV is measured in squared units). Thus the variable whose removal most increases lack of fit is awarded 100% relative importance, variables which do not appear in the model at all are adjudged 0% important, and a variable whose removal increases lack of fit only 1/4 as much as the 100%-important variable is awarded 50% relative importance.

Whilst these figures may be a useful guide to relative variable importance, the figures should not be taken as any definitive ranking of variable importance. Particularly, it should be noted that the amount of GCV added by removing all model terms featuring a particular variable v may be wildly different from the GCV that would be added by removing v from the original list of candidate variables and building the model afresh. This is especially the case where there are two or more related variables. For example, a temperature smooth with a half-life of 8 hours might feature heavily in a model, yet a temperature smooth with half-life of 7 hours feature much more lightly or be overlooked completely. However, were the former variable removed and the model built again, we would expect the latter variable to gain significantly in importance, and the overall model fit to remain largely unchanged.

Unfortunately it is not practical to judge the importance of each variable by rebuilding the model without the variable, whereas the relative importance figures produced by MARS are cheaply calculated.

11.3.6 Experiment 1: Preliminary Run, No Cross-Validation

A very large MARS model was run using a very wide selection of variables. The maximum number of predictor variables in any interaction term was restricted to 3 (as was the case in all following experiments unless noted otherwise), the number of basis functions selected (prior to pruning) was set at 90, and 55 variables were made available for selection, as follows:

- 26 temperature-based variables: θ, $\theta^{\max}$, $\bar{\theta}$ and $\theta^{\min}$, plus temperature lags of 2, 4, 8 and 48 half-hours, delta temperatures for 2, 4, 24, 48 and 336 half-hours and exponentially smoothed temperatures with half-lives of 1, 2, 4, 6, 12, 48, 96, 336 and 1344 half-hours[1].
- 6 humidity-based variables: w_h (current humidity), plus lags of 2 and 4 half-hours and smooths of 1, 2 and 4 half-hours.
- 4 windspeed-based variables: w_w (current windspeed), plus a lag of 2 half-hours and smooths of 1 and 3 half-hours.
- 4 cloud cover-based variables: w_c (cloud cover), plus, again, a lag of 2 half-hours and smooths of 1 and 3 half-hours.
- 8 temporal variables: time of day t, day of year, *summer*, *spring*, *noon*, *evening*, plus sunset w_s and seasonality w_s^*.
- 10 illumination-based variables: (extraterrestrial) *insolation*, plus lags of 1 and 2 half-hours and smooths of 2 and 4 half-hours; plus (estimated) *darkness*, with the same lags and smooths. In this instance, darkness was calculated using a cloud transparency parameter, somewhat arbitrarily, of $CT = 0.5$ (see 11.3.3).
- 1 noise variable (see 11.3.2).

The MARS smoothing parameter d of (EQ 60) was set at 3.0, this being the recommended typical optimal value [9]. Parameter d was kept at 3.0 in all the following ex-

1. In the results, exponentially smoothed variables are denoted by their half-lives (in half-hours), rather than their smoothing exponent α (EQ 88), for ease of comparison with lag and delta variables. The half-life of an exponentially smoothed series is the time it will take for the current value v_t's contribution to the series to decay to half that level. Half-life H and exponent α are related by the formula $\ln \alpha = \left(\ln \frac{1}{2}\right) / H$.

periments, except where otherwise noted.

Weather dependent loads were calculated from the (normalised, special days-removed) 1995/6/7 database using a weather separation as described in section 11.1, using only day of the week (d) as a separating variable (i.e. $X^I = \{d\}$, so there are just seven weather independent profiles). Except where otherwise noted, all the weather separations for the results presented were generated thus. This separation results in a total of 32112 observations being presented to MARS (=48×669, 48 half-hours and 669 dates, after 62 special days out of a possible 731 days have been excluded).

The intention of this run was to narrow somewhat the large pool of candidate variables, and particularly to identify some suitable values for the time period of various lag, delta and smoothed variables. Even this broad set of variables needed to be picked judiciously (based on the experiences of many exploratory runs) to keep computation times acceptable.

The *GCV score for the unmodelled data* is defined as that arising from a constant MARS model (one with a single constant term, denoted a_0 in (EQ 57)); this is a measure of the total variation in the data. The unmodelled GCV for the data is 63.14. The model generated for the above described experiment resulted in 38 ANOVA functions (see section 6.5.3) after the pruning phase, and the resulting piecewise-linear MARS model has a GCV score of 14.09. This yields a percentage of variation (as measured by root percentage of GCV) accounted for by MARS of 88.14% (=$\sqrt{\frac{(63.14 - 14.09)}{63.14}} \times 100\,\%$). The square root takes into account that GCV is measured in squared units.

The relative variable importances determined by MARS appear in (TABLE 5.)[1], which also features the correlation coefficients of each variable with the response (in the

1. Tables presented in this chapter are partially automatically generated by the data mining software, and feature abbreviations of some variable names as used in the software. Most are just a four letter abbreviation of the variable's full name, and are easily deciphered. The only obscure abbreviations are 'avey' for average temperature yesterday, 'maxy' and 'miny' for peak and trough temperatures yesterday, and dofy for day of year. Asterices are used for 'not applicable'. 'RESP' is the response variable, which, of course, is not actually used as a predictor variable.

Y-Correlation column). The first column shows the variable number, and the second,

TABLE 5.

No	Type	Subtype	1/2-hrs	Y-Correlation	Importance: Experiment 1	Importance: Experiment 2	Importance: Experiment 3
1	RESP	*	*	1	*	*	*
2	avey	*	*	-0.4606	9.453	41.93	16.99
3	clou	smooth	1	0.1177		*	
4	clou	smooth	3	0.1192	14.26	*	19.19
5	clou	lag	-2	0.1129		*	
6	clou	*	*	0.1159		*	
7	dark	smooth	2	0.2613		*	
8	dark	smooth	4	0.332		*	
9	dark	lag	-1	0.2179		*	
10	dark	lag	-2	0.2434		*	
11	dark	*	*	0.1913	7.870	*	6.811
12	dofy	*	*	-0.1248	44.85	*	44.97
13	even	*	*	-0.02852	100.0	*	100.0
14	humi	smooth	1	0.3127	12.25	*	17.47
15	humi	smooth	2	0.324		*	
16	humi	smooth	4	0.3361	11.98	*	12.51
17	humi	lag	-2	0.3159		*	
18	humi	lag	-4	0.3206		*	
19	humi	*	*	0.3019		*	
20	inso	smooth	2	-0.2584	17.58	*	11.50
21	inso	smooth	4	-0.3293	23.34	*	19.66
22	inso	lag	-1	-0.2163	30.73	*	29.25
23	inso	lag	-2	-0.2419		*	
24	inso	*	*	-0.19	31.77	*	35.06
25	maxy	*	*	-0.4594	13.35	100.0	16.94
26	miny	*	*	-0.4413		52.77	12.41
27	nois	*	*	0.01252			
28	noon	*	*	0.007577	36.87	*	45.04
29	seas	*	*	0.09116	22.21	*	26.56
30	spri	*	*	0.1871	26.90	*	35.96
31	summ	*	*	-0.4968	9.751	*	27.97
32	suns	*	*	-0.4757	15.56	*	20.77
33	temp	smooth	1	-0.4851			
34	temp	smooth	2	-0.4893			
35	temp	smooth	4	-0.4916		18.03	
36	temp	smooth	6	-0.4896			
37	temp	smooth	12	-0.4827	17.03	38.08	23.46

TABLE 5.

38	temp	smooth	48	-0.4811	13.60	33.04	16.86
39	temp	smooth	96	-0.4855	31.96	43.11	16.01
40	temp	smooth	336	-0.4804		34.59	14.61
41	temp	smooth	1344	-0.3759	25.11	42.77	33.29
42	temp	delta	-2	0.06831		13.04	
43	temp	delta	-4	0.05303		12.45	
44	temp	delta	-24	-0.1334		52.93	
45	temp	delta	-48	-0.0249		43.86	
46	temp	delta	-336	-0.01002			
47	temp	lag	-2	-0.4883			
48	temp	lag	-4	-0.4914			
49	temp	lag	-8	-0.4848		20.44	
50	temp	lag	-48	-0.473		33.68	
51	temp	*	*	-0.4804	34.64	32.37	42.19
52	time	*	*	-0.01999	14.77	*	18.38
53	wind	smooth	1	0.07475		*	
54	wind	smooth	3	0.07743	7.514	*	7.617
55	wind	lag	-2	0.06949		*	
56	wind	*	*	0.07604		*	

third and fourth give the abbreviated basic variable name, and where applicable the type of derivation used to generate the variable (lag, exponential smooth, delta) and the time period (in half hours) used in that derivation. Blank cells represent zero importance (variables not used).

Firstly, it should be noted that almost all of the variable types included are represented in some form in the model; temporal variables (both daily and yearly, as well as sunset-derived variables), derivatives of temperature, cloud cover, humidity, windspeed, and derived illumination variables (insolation and darkness) all feature. Maximum and mean temperatures from the previous day (but not the minimum temperature) are featured; only noise (as we would expect) plays no part at all.

Next to be noted is the great importance of the temporal variables, as reckoned by MARS. Particularly, closeness-to-18:00 GMT (i.e. *evening*), and also closeness-to-noon and the day of the year, are ranked as very important; hour of day (*time*), closeness-to-spring, and seasonality also figure highly, whilst closeness-to-summer and sunset time figure as less important (at least partially because the two variables share the same burden, being strongly correlated and representing the same essential concept).

Extraterrestrial insolation, and variables derived from it, are temporal variables

strictly speaking (though our estimated darkness variable must be regarded as a weather variable, since it depends on cloud cover). Spot insolation and 1/2-hour lagged insolation both appear to be influential variables, and short-term smooths of insolation also featured. (Spot) estimated darkness featured in the model, though not heavily.

Spot temperature and smoothed temperature variables appear to be easily the most important of the non-temporal variables; particularly the spot temperature and some medium and long term smooths (notably those with half-lives of 2 days and 28 days, i.e. 96 and 1344 half-hours); shorter term smooths (half-lives 12 and 48 half-hours) also play their part.

The delta temperature variables were not selected at all by the model; various exploratory runs had found delta temperature variables to be of use, however. At least in this experiment, lag variables were also found to be of very little importance (only one variable - the half hour lagged insolation - being utilised), though again this had not always been the case in prior exploratory runs. These observations prompted Experiment 2 (section 11.3.7).

Of the remaining weather variables, humidity and cloud cover, and to a lesser extent windspeed, all influenced the model, all in the form of short-term smooths.

Of the 38 ANOVA functions, 7 were functions of one predictor, 10 were functions of two predictors, and the remainder (21) were functions of three predictors (recall the maximum interaction level is set at three). There is no need to detail them all for this preliminary experiment, but it is worth noting that no humidity variables interacted with temperature (or any other weather) variables, that windspeed interacted with a medium-term (half-life two days) temperature smooth, and that cloud cover-based variables only appeared in interaction with insolation-based variables. Every interaction term included at least one temporal variable - there were no weather-only interaction terms (unless insolation is counted as a weather variable).

11.3.7 Experiment 2: Temperature-Based Variables Only

The same data and parameters were used as in experiment 1, but only temperature-based variables (plus a noise variable) were available for selection (23 variables). As would be expected, much less GCV was accounted for by the resulting model. The piecewise linear version of the model has a GCV score of 26.68 (28.86 for the piecewise cubic version), compared to 14.09 for the piecewise linear model of experiment 1. Thus the

(piecewise linear) model accounts for a root percentage of GCV of 75.99% (compared to 88.14% in experiment 1).

There is no surprise that a model which is unaware of time-of-day variables can account for much less of the variation in weather dependent load (never mind the absence of the other variables), since the effect of weather on load is strongly time-of-day dependent. However it is interesting to note the changes brought about in the relative importance of the temperature-based variables, shown in (TABLE 5.). Perhaps the most important change is that delta temperature variables now play a role, and a reasonably important one; the temperature change from 12 and from 24 hours ago appear particularly influential. Lag temperature variables now also play a part, though are still of lesser influence than exponentially smoothed temperatures. The previous day's statistics (peak, mean, trough), especially the previous day's peak, assume a greater significance than most of the other variables, whereas in experiment one they were less important than other temperature variables.

It is only to be expected that when a large selection of variables known to be useful in load/weather modelling are removed, the importance of some remaining variables will increase. Exactly why the previous day's statistics, and the delta temperature change from 24 hours ago, are rated as so important in the absence of the non-temperature-based variables is unclear. In the case of the delta temperature with lag 8 half-hours, it is possible that it is rated as important partly because the variable conveys indirect time-of-day information, in that the temperature tends to rise during the morning and fall through the evening (no matter what the season); however the results of experiment 2 indicate that delta variables are at least worth considering as candidate variables in load/weather models.

11.3.8 Experiment 3: Cross-Validation Results

n -fold cross-validation is a common technique for assessing the accuracy of predictive mathematical models such as MARS. Whilst a goodness of fit statistic is generally easy to calculate for a model, it only measures the goodness of the fit of the built model to the actual data presented, not its goodness of fit to data of that type per se. However since we generally assume that our data are drawn from a much larger, usually infinite population, it is necessary to perform some kind of cross-validation in order to verify to what extent a model's claim of good fit applies to the population at large. This is espe-

cially relevant when the data is noisy, since an overfitted model may claim an excellent goodness of fit when it is in fact fitting to the noise; when that model is applied to fresh data (which has different noise) the goodness of fit may be far lower.

n-fold cross-validation errors are calculated by subdividing the sample population into n sub-populations, as equal in size as possible, in a completely random manner. n different models are then built, each using $n-1$ of the sub-populations (combined) as training data (on which the model is fitted) and the remaining sub-population as test data. The cross-validation error for each model is the lack of fit of the fitted surface to the *test* data. The n cross-validation errors are then usually averaged to obtain a single cross-validation error figure.

The main smoothness parameter in MARS, degrees-of-freedom d charged per basis function added (EQ 60) is intended as a parameter with which to guard against overfitting in MARS. Each basis function added adds a knot to the fitting function, and fewer knots result in a smoother fitted surface. Addition of too many knots to the fitted surface increases the likelihood of overfitting the model, so it is necessary to use a cross-validated lack of fit measure to determine a suitable value for d. MARS has a built-in routine to estimate the optimal value for d based on n-fold cross-validation errors.

Using this routine, based on a ten-fold cross-validation scheme, experiment 1 was repeated. Tenfold cross-validation greatly increases the algorithm's computational complexity (by a factor of around 10); furthermore, the number of basis functions (prepruning) was increased from 90 to 120, imposing even greater complexity. It was not feasible to compute cross-validation on very many of the experiments presented here because of the complexity, but the results in this instance are enlightening.

(TABLE 6.) shows a snippet of the output for the tenfold cross-validation run. The

TABLE 6.

Number of basis functions	Corresponding d	Cross-Validation Errors (CV)
120	-1.95	12.82
119	-1.88	12.82
118	-1.78	12.82
117	-1.70	12.82
116	-1.59	12.82
115	-1.30	12.81
114	-.80	12.81
113	-.42	12.82
112	-.02	12.82

TABLE 6.

111	.29	12.81
110	.98	12.81
109	1.64	12.82
108	1.97	12.82
107	3.14	12.82
106	4.11	12.82
105	4.95	12.83
...	...	...
91	21.03	12.91
90	22.19	12.92
...	...	...

cross-validation errors (CV) are almost constant for any final (post-pruning) number of basis functions from 105 to 120 (and in fact, are little changed for as few as 90 basis functions). Though Friedman recommends values between 2.0 and 4.0 for d [9], it would appear that for the problem presented here, a much wider range of values for d can be used with minimal effect on goodness of cross-validated model fit.

The cross-validation routine determined the optimal d parameter at 0.98, corresponding to 110 basis functions, with a predictive square error (averaged over 10 cross-validation models) of 12.81. Running with d=0.98 and 120 candidate basis functions, the MARS model achieves an actual GCV score of 12.98 (for the piecewise linear version; 14.86 for the piecewise cubic version), meaning it accounts for a root percentage 89.13% of GCV.

Comparing the relative importances in variables between experiments 1 and 3 (TABLE 5.) there is very little difference to speak of, suggesting that although seeking 120 (rather than 90) candidate basis functions does not noticeably degrade cross-validated goodness of fit (i.e. does not overfit), it does little to radically alter the final model.

Since the estimated optimal value for smoothness parameter d was shown to result in very similar cross-validated goodness of fit for such a wide range of values, it was not deemed necessary to alter its value from the suggested default value of 3.0 (see [9]) in any of the subsequent experiments, though the suggested value (approximately) was used in the final model (α) of section 11.3.12.

It must be noted that a cross-validation scheme which selected n random sub-populations of *customers* with which to build/test separate models, rather than of observations, might actually be more appropriate for our purposes; under the MARS cross-validation scheme described, every sub-population is certain to contain observations

from every customer, and so customers with extremely unusual weather dependent loads are unlikely to contribute quite as much lack-of-fit when appearing in test data as they would under a cross-validation based on customer sub-populations. Unfortunately the cross-validation scheme is programmed into the MARS code in a way which would be very difficult to change, and time did not permit for this type of cross-validation to be implemented. However, there are enough customers in the database that any difference between these cross-validation schemes would most likely be slight.

11.3.9 Experiment 4: Illumination Variables

The variables available governing the amount of natural illumination present, and hence affecting lighting loads, are principally cloud cover, (extraterrestrial) insolation, (estimated) darkness and sunset time (and, excluding sunset, their lagged and smoothed versions). Other temporal variables also influence the amount of natural illumination, most notably time of day and closeness to summer; however they are not considered here, since extraterrestrial insolation is of more direct relevance to lighting loads than any of the time of day-based variables, and because sunset time is presumably at least as relevant to lighting loads as any of the time of year-based variables.

The principal problem in trying to determine the importance of the above variables on lighting loads is that most of them, directly or indirectly, strongly influence other loads, especially heating and cooling loads. Sunset time is strongly correlated with temperature variables, as is insolation (and inversely, darkness), and even cloud cover is co-dependent with other weather variables (having a positive correlation with relative humidity and a negative correlation with temperature). Thus it cannot be supposed that a model featuring only the above variables will model mostly lighting loads rather than other loads. Even if the actual recorded natural illumination figures were available to us, there would be no way to isolate its effects on lighting loads from its indirect effects on other loads, natural illumination having such strong correlations with other meteorological and temporal variables.

The purpose of the experiments in this section was only to try and determine whether our estimated darkness variable is of much utility relative to cloud cover and insolation, and to try and determine a suitable value for the cloud transparency parameter CT of (EQ 99).

To this avail, models were built using the same model parameters as experiment 1,

except that the number of basis functions to be determined was only 50. The candidate variables consisted of 2 temperature smooths (with half-lives of 4 hours and of one week); yesterday's mean temperature; closeness-to-evening (18:00 GMT) and closeness-to-noon; sunset time; plus a small selection of lagged and smoothed versions (and a spot version) of (a) cloud cover, (b) extraterrestrial insolation and (c-g) estimated darkness with a variety of cloud transparency parameters. The temperature variables were included so that when variables deriving from cloud cover, insolation or darkness were selected it was not merely due to their correlations with temperature (temperature being the most influential non-temporal variable); the temporal variables for similar reasons (since insolation/darkness have strong correlations with temporal variables). The results appear in (TABLE 7.).

TABLE 7.

Experiment Number:				4a	4b	4c	4d	4e	4f	4g
Illumination Variable Type:				clou	inso	dark				
Cloud Transparency Parameter:				*	*	0.20	0.40	0.60	0.70	0.80
No	Type	Subtype	1/2-hrs	Relative Variable Importance						
1	RESP	*	*							
2	avey	*	*	27.10	70.95	73.91	48.02	59.93	60.00	57.11
8	even	*	*	100.0	100.0	100.0	100.0	100.0	100.0	100.0
9	nois	*	*							
10	noon	*	*	37.53	41.63	31.04	31.86	35.36	43.78	33.44
11	suns	*	*	67.67	51.11	68.19	62.92	53.45	47.93	42.46
12	temp	smooth	4	31.76	36.61	30.86	38.41	40.15	42.21	28.97
13	temp	smooth	336	22.08	23.87	28.43	25.67	24.84	21.67	24.86
3	c/i/d[a]	smooth	2		28.16	36.69	24.66	34.93	33.07	18.29
4	c/i/d	smooth	4	9.542	32.69	34.86	17.15	6.332	6.793	10.36
5	c/i/d	lag	-1							13.78
6	c/i/d	lag	-2							
7	c/i/d	*	*			12.56	7.535	10.10	11.91	
GCV score (piecewise linear):				17.79	17.85	18.37	17.73	17.59	17.38	17.84
GCV score (piecewise cubic):				18.53	18.32	19.24	18.71	18.50	17.84	18.31
Correlation between observed and (piecewise cubic) fitted:				0.8394	0.8409	0.8329	0.8381	0.8391	0.8457	0.8420

a. c/i/d means cloud cover in experiment 4a, insolation in experiment 4b, and darkness in experiments 4c-4g.

The GCV scores for experiment 4a (using cloud cover) and experiment 4d (using extraterrestrial insolation) are quite similar (although which model is judged better depends on whether the piecewise linear or piecewise cubic version is used). Note that the

cloud variables are not rated as particularly important in experiment 4a.

In experiments 4c-4g, using estimated darkness, cloud transparency parameters of CT = 0.2, 0.4, 0.6 and 0.8 were used, and subsequently, a cloud transparency parameter of 0.7 was also tried. For the lower values of CT, fit was worse than, or barely better than, either cloud alone or insolation alone. However for CT values of 0.7 and 0.8, fit was better than in experiments 4a and 4b, particularly so for $CT = 0.7$.

In any of the following experiments in which estimated darkness appears, a cloud transparency figure of $CT = 0.75$ has been adopted. It should not be inferred, however, that the actual average transparency figure for clouds above the geographical area under study is necessarily very near this figure.

11.3.10 Experiments 5, 6 & 7: Humidity and Windspeed; Discomfort and Wind Chill Factors

There are a variety of measures used by meteorologists which attempt to describe the combined effect of temperature and (relative) humidity on the level of comfort felt by humans experiencing high temperatures. Air conditioning loads would be expected to increase as human discomfort increases, and might be expected to follow a discomfort index more closely than either temperature or humidity individually, prompting the introduction of a discomfort index to the weather model.

The most common of these measures is the *Heat Index* (also known as Apparent Temperature, also known as Temperature-Humidity Index) [35]. Another measure of the discomfort arising from high temperature and high humidity is the *Summer Simmer Index* [36]. Both measures combine temperature and humidity into a figure reckoned to represent how hot it actually feels when it is both hot and humid (rather than hot and dry), measured in degrees Fahrenheit. The figures do not differ too greatly from one another for most of the temperature/humidity range for which they are valid; however both formulas become completely meaningless for low temperatures. There being little reason to prefer one to the other, the Summer Simmer Index was deemed more useful for our purposes because it is easier to extend the formula to be meaningful for lower temperatures; there is a single temperature, 58 degrees Fahrenheit, at which the index is constant whatever the relative humidity, a temperature at which there is deemed to be no discomfort due to heat. Therefore by using a modified form of the Summer Simmer Index (which we will just call *discomfort*) which remains constant at or below 58 degrees Fahrenheit, an index meaningful for all (reasonable) temperatures is recovered (there is no

such cutoff point inherent in the Heat Index formula). The modified Summer Simmer Index, then, is defined for Fahrenheit temperatures as

$$discomfort = \begin{cases} 58 & \theta_F \leq 58 \\ 1.98\,(\theta_F - (0.55 - 0.0055 w_h)\,(\theta_F - 58)) - 56.83 & \theta_F > 58 \end{cases} \quad \text{(EQ 101)}$$

where θ_F is the Fahrenheit temperature, and w_h the relative humidity (a percentage figure), and by converting to and from degrees Fahrenheit and Celsius, a version is obtained which can be used in our MARS model. Lagged and smoothed version of this spot variable are also allowed as candidate variables; in fact, it could be the case that a different lag/half-life is appropriate for temperature than for relative humidity in the calculated discomfort index, but allowing such flexibility would lead to an explosion in the number of candidate variables and was deemed counter-productive.

Whilst windspeed may also have an effect on air-conditioning loads (see 10.5.1) it is more closely associated by modellers of the load/weather relationship with its effect on heating loads, due to the cooling effects of the wind (again, see 10.5.1). Like discomfort due to heat and humidity, the combined effect of wind and temperature are often combined into one statistic. The most common of these is the Wind Chill Factor, which models the cooling power of the wind on skin, usually in watts per meter squared. This figure is actually intended to model the apparent coldness for a human outdoors, rather than indoors, where except as draughts and through opened doors, the effect of wind will not usually be felt directly. Since heating loads obviously depend chiefly on the indoor conditions, Wind Chill Factor might not be an entirely appropriate candidate variable for a load/weather model. However, since windspeed was demonstrated in experiment 1 to be a non-trivial factor in the load/weather relationship, presumably mostly due to its interaction with the cold, it was felt that some measure of the combined effect of coldness and windspeed might be useful, and Wind Chill Factor is the most obvious candidate.

Various versions of Wind Chill Factor are in use; the version that was used derives from the Meteorological Service of Canada [37], and we will call it just *chill*:

$$chill = 1.1626\left(5.2735\sqrt{w_{wm}} + 10.45 - 0.2778 w_{wm}\right)(33.0 - \theta) \quad \text{(EQ 102)}$$

Here, θ is the temperature (degrees Celsius) and w_{wm} is the windspeed in miles per hour; conversion of windspeed from knots to miles per hour was necessary before calculating chill.

Three experiments were conducted to evaluate the effectiveness of the derived variables discomfort and chill. Firstly, in experiment 5 a moderate selection of variables, which did not include windspeed, humidity, chill or discomfort, were presented to MARS (with the same model parameters as in experiment 1). Secondly, in experiment 6 these variables were used once more, together with windspeed and humidity (and some lagged and smoothed variables derived from windspeed and humidity). Finally, in experiment 7 the original variable set was used in conjunction with chill and discomfort variables (again with some lags and smooths), without windspeed and humidity. The results are presented in (TABLE 8.).

TABLE 8.

						Experiment 5	Experiment 6	Experiment 7
No	Type	Subtype	1/2 hours	Y-Correlation		Importance	Importance	Importance
1	RESP	*	*	1		*	*	*
2	dofy	*	*	-0.1248		22.89	25.02	16.15
3	even	*	*	-0.02852		100.0	100.0	100.0
10	nois	*	*	0.01252				
11	noon	*	*	0.007577		43.48	38.70	43.41
12	spri	*	*	0.1871		44.46	31.14	34.45
13	summ	*	*	-0.4968		91.21	69.48	70.33
14	temp	smooth	1	-0.4851				
15	temp	smooth	2	-0.4893		25.84	16.80	12.99
16	temp	smooth	4	-0.4916		13.92		6.421
17	temp	smooth	6	-0.4896				
18	temp	smooth	12	-0.4827		30.26	22.69	15.05
19	temp	smooth	48	-0.4811		32.84	28.25	20.83
20	temp	smooth	96	-0.4855		40.03	32.96	29.62
21	temp	smooth	336	-0.4804				12.52
22	temp	smooth	1344	-0.3759		32.66	23.66	15.44
23	temp	lag	-2	-0.4883				10.93
24	temp	lag	-4	-0.4914				
25	temp	*	*	-0.4804		21.24	6.479	10.93
26	time	*	*	-0.01999		23.80	13.04	15.21
				humi	disc			
4	humi/disc[a]	smooth	1	0.3127	-0.2336	*		
5	humi/disc	smooth	2	0.324	-0.2396	*		
6	humi/disc	smooth	4	0.3361	-0.2391	*	7.191	
7	humi/disc	lag	-2	0.3159	-0.2402	*		
8	humi/disc	lag	-4	0.3206	-0.2467	*		
9	humi/disc	*	*	0.3019	-0.226	*	20.79	

TABLE 8.

				wind	chil			
27	wind/ chil[b]	smooth	1	0.07475	0.4739	*		5.612
28	wind/ chil	smooth	2	0.07501	0.4781	*		1.262
29	wind/ chil	smooth	4	0.08147	0.4797	*	17.45	8.565
30	wind/ chil	lag	-2	0.06949	0.4736	*		
31	wind/ chil	lag	-4	0.0672	0.4755	*		
32	wind/ chil	*	*	0.07604	0.4681	*		

a. Humidity in experiment 6, discomfort in experiment 7.

b. Windspeed in experiment 6, chill in experiment 7.

In each case the GCV score for the unmodelled data is (again) 63.14. The GCV scores for the piecewise linear versions of the constructed MARS models are respectively, 15.18, 14.39 (humidity and windspeed included) and 14.93 (discomfort and chill included).

The first thing to note is that humidity is preferred by MARS to our discomfort measure: indeed discomfort and its derivatives are not selected at all, whereas plain relative humidity (spot humidity and humidity smoothed with a half-life of 4 half-hours) is selected. This is perhaps surprising, but less surprising in light of the fact that response is, in fact, *inversely* correlated with discomfort in the data (see TABLE 8.). This is perhaps counter-intuitive, since discomfort is constant when temperatures are low (i.e. less than 58 degrees Fahrenheit, or 14.5 degrees Celsius). It may well be the case that the cutoff point used in the summer simmer index is much too low for our load/weather model (at least for UK business loads), hence the negative load/discomfort correlation. It would appear that even as discomfort (as measured by (EQ 101)) increases, the loads of the Sponsor’s customers are still being relaxed, possibly due to lower water heating costs, and possibly due to lower summer lighting costs. There might be some mileage in replacing the discomfort measure of (EQ 101) with some other measure, one with a higher cutoff point. However, there are four interaction terms involving humidity in the ANOVA decomposition of experiment 6. One involves spot humidity and closeness-to-summer; two involve spot humidity and temperature smoothed with a half-life of 28 days (once with closeness-to summer, once with the day of the year); and the last involves smoothed humidity (2 hour half-life), smoothed temperature (2 day half-life) and smoothed windspeed (2 hour half-life). Thus it would appear MARS is capable of constructing terms which take account of the effect of humidity on human discomfort, and even of windspeed on human discomfort, without needing the assistance of an explicit

discomfort variable.

The wind chill variable of (EQ 102), in contrast, is selected by MARS in various forms, though it does not appear to assume markedly more importance than windspeed. Smoothed windspeed (2 hours half-life) appears in ANOVA interaction terms with medium-term temperature smooths, and closeness to summer (as well as with humidity/ temperature as above) in experiment 6, suggesting that wind chill can be modelled implicitly by MARS just as well as by including an explicit Wind Chill Factor.

Indeed, in another experiment on the same data (TABLE 9.) featuring windspeed, humidity, chill *and* discomfort, discomfort again went unselected, whilst chill and windspeed were both selected; however windspeed appeared to have greater importance than chill. No temporal variables were present in this model, in order to better highlight the effect of weather variables (compare with experiment 2, which featured only temperature-based variables).

TABLE 9.

No	**Type**	**Subtype**	**1/2-hrs**	**importance**	**No**	**Type**	**Subtype**	**1/2-hrs**	**importance**
1	**RESP**	*****	*****	*****	**26**	**temp**	**smooth**	**2**	
2	avey	*	*	100.0	27	temp	smooth	4	
3	chil	smooth	1		28	temp	smooth	6	37.71
4	chil	smooth	2		29	temp	smooth	12	65.77
5	chil	smooth	4		30	temp	smooth	48	39.32
6	chil	smooth	6	16.41	31	temp	smooth	96	35.59
7	chil	*	*		32	temp	smooth	336	35.20
8	clou	smooth	1		33	temp	smooth	672	66.07
9	clou	smooth	3		34	temp	smooth	1344	69.74
10	clou	smooth	5		35	temp	smooth	2688	63.64
11	clou	*	*		36	temp	delta	-2	
12	disc	smooth	1		37	temp	delta	-4	
13	disc	smooth	2		38	temp	delta	-24	31.05
14	disc	smooth	4		39	temp	delta	-48	17.63
15	disc	smooth	6		40	temp	delta	-336	20.52
16	disc	*	*		41	temp	lag	-2	
17	humi	smooth	1		42	temp	lag	-4	
18	humi	smooth	2	21.06	43	temp	lag	-8	47.52
19	humi	smooth	4	94.08	44	temp	lag	-48	51.88
20	humi	smooth	6	47.08	45	temp	*	*	
21	humi	*	*	33.62	46	wind	smooth	1	
22	maxy	*	*	24.31	47	wind	smooth	3	16.45
23	miny	*	*	40.58	48	wind	smooth	5	39.26

TABLE 9.

No	Type	Subtype	1/2-hrs	importance	No	Type	Subtype	1/2-hrs	importance
1	**RESP**	*****	*****	*****	**26**	**temp**	**smooth**	**2**	
24	nois	*	*		49	wind	*	*	
25	temp	smooth	1						

Aside from the observations about the relative merits of chill and discomfort compared to windspeed and humidity, it is interesting to note the huge relative importance attributed by MARS to humidity (and its lags and smooths); indeed it appears to be attributed around as much importance as temperature and its lags and smooths, though not as much as the previous day's mean temperature. Since humidity is not found to have an influence comparable with that of temperature in the presence of temporal variables, it is possible that the way in which MARS estimates variable importance is giving undue importance to humidity here, especially as (relative) humidity has a fairly strong (negative) correlation with temperature (humidity smoothed with a half life of 2 hours has a correlation coefficient of -0.4569 with a temperature smooth having the same half-life, and a correlation coefficient of -0.4700 with closeness-to-summer). It is possible therefore that humidity is taking some of the role that would otherwise be taken by closeness-to-summer, or that could be modelled by temperature variables, in its absence.

Another explanation for the high relative importance of humidity is that cold weather heating loads may be elevated when humidity is high. This could arise because cold temperatures are experienced as more unpleasant when the air is damp than when the air is dry.

For whatever reason, there seems little doubt that humidity is a useful predictor in the load/weather models presented, and of more importance than windspeed.

The ANOVA decomposition again features terms in which humidity variables interact with one or more of: temperature variables; closeness-to-summer; and windspeed variables. Humidity smoothed with half-life 4 half-hours also appears in interaction terms with delta temperature variables (with 24, 48 and 336 half-hour lags).

11.3.11 Experiments with Mixed Categorical and Continuous Variables

Using the mixed MARS model of section 6.6.2 (as detailed in sections 11.2.3 and 11.2.4), models were built featuring a moderate selection of temporal and weather variables together with various categorical variables from $\{d, X_5, X_6, X_L\}$ (day of the week, SIC code, tariff code and load factor category). Seven (maximum entropy) load factor

categories were employed, based on recalculated 1-percentile load factors (see section 13.5.2 for an explanation of percentile load factors). There are 7 tariff codes and 41 SIC codes represented in the data.

In the previous experiments, weather decomposition was performed by subtracting one of just 7 weather independent profiles (corresponding to the day of the week), see section 10.5.5 for details. In the following experiments, a full weather decomposition using all four categorical variables (day-of-the-week, load factor category, tariff code and SIC code) was performed to arrive at the weather-dependent loads used for the mars models. There were in fact 1491 weather-independent atoms; these were generated using recalculated 1-percentile load factors and 7 load factor categories, and featured 7 distinct tariff codes and 41 distinct SIC codes. Thus of the $7 \times 7 \times 7 \times 41 = 14063$ possible combinations of day of the week/load factor category/SIC/tariff code, only 10.6% have any actual representatives in the 1995/6/7 database.

The experiments performed involved (i) no categorical variables, (ii) day of the week only, (iii) load factor category only, (iv) day of the week and load factor category, (v) tariff code only, (vi) day of the week and tariff code and finally (vii) SIC code only. For (i) to (vi), 90 basis functions were sought. However, due to the massive size of the problem when SIC code is used as a categorical variable, it was not possible to seek as many as 90 basis functions, and the number had to be limited to 40 in order to fit into machine memory; even with this restriction, the program required over 1600 megabytes of machine memory. Thus the results of (vii) are thus somewhat compromised, and this should be borne in mind when comparing with (i)-(vi).

It is important to note that the data presented is not the same in all these experiments; even the number of observations varies (see TABLE 10.) because of the way in which the response values are calculated when categorical customer variables are employed (see sections 11.2.3 and 11.2.4). Whilst (i) and (ii) have the same data (since introducing day of the week does not require each date's data to be split into categories), experiment (iii) has seven times as much data (since on any date we need separate weather-dependent load readings for each load factor category). Experiment (iv) uses the same data as experiment (iii) since only day of the week is added. Experiments (v) and (vi) use nearly seven times as much data as experiment (i) (not exactly 7, because one or more tariff code must represent customers all of whom have data missing on a certain date or dates). Similarly, experiment (vii) uses not far off 41 times as much data as experiment 1, hence the excessive computational demands when using SIC codes as a categorical predictor.

Relative variable importances from the experiment, and some other model statistics, are shown in (TABLE 10.). The results of (i) do not show any major departures from a previous similar experiment (experiment 1), and are given for comparison purposes.

TABLE 10.

Experiment:			(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)
Type	Subtype	1/2-hrs	Relative Variable Importance						
Categ'l Variables Used:			none	day	ld fac	day & ld fac	tariff	day & tariff	SIC
avey	*	*	28.13	13.43					
chil	smooth	1							
chil	smooth	2							
chil	smooth	4							15.85
chil	smooth	6	22.22	20.70			28.80	42.06	
dark	smooth	2	13.45	29.76	31.66	19.51	19.54	50.05	20.92
dark	smooth	4	43.73	35.93	53.68	54.07	41.40	42.80	23.93
disc	smooth	1							
disc	smooth	4							
disc	*	*							
dofy	*	*	30.87	39.34	29.71	28.36	36.44	34.76	
even	*	*	100.0	100.0	100.0	100.0	100.0	100.0	57.51
humi	smooth	1	9.701	24.94					
humi	smooth	4	9.075						
humi	*	*	12.22		15.45	11.92	21.78		
maxy	*	*							
miny	*	*	10.16	7.448					
nois	*	*							
noon	*	*	31.17	32.85	42.31	36.28	41.74	33.93	17.55
spri	*	*	43.08	34.73	43.71	41.02	37.86	32.91	15.88
summ	*	*	46.99	38.48	44.23	32.43	48.46	39.77	22.18
temp	smooth	1							
temp	smooth	2							
temp	smooth	4							
temp	smooth	8	19.54	23.61	31.74	28.08	23.19	23.35	
temp	smooth	12					9.122		
temp	smooth	48	16.79	17.30	17.92	32.72	11.74	11.48	
temp	smooth	96	47.40	42.23	47.99	43.44	31.88	43.88	100.0
temp	smooth	336					9.608		
temp	smooth	1344	30.97	14.07	43.80	41.22	30.89	15.53	
temp	delta	-2							
temp	delta	-4							
temp	lag	-2							

TABLE 10.

temp	lag	-4							
temp	lag	-48							
temp	*	*			16.26	18.54			
time	*	*	15.97	43.98	17.57	37.13		17.89	14.07
wind	smooth	2	10.10						
wind	smooth	4	8.725		14.18				
day of wk.	*	*	*	35.27	*	40.73	*	46.02	*
other categ.[a]	*	*	*	*	95.76	91.33	67.96	67.32	90.95
			none	day	ld fac	day & ld fac	tariff	day & tariff	SIC
Number of observations:			32 112		224 784		223 344		1 223 808
GCV in unmodelled data:			62.52		99.76		100.1		222.4
GCV score (piecewise linear):			14.50	13.24	42.54	41.26	46.52	45.69	153.3
GCV score (piecewise cubic):			15.64	14.54	43.29	42.87	47.37	47.26	154.1
Root percentage GCV accounted for (piecewise linear model):			87.64	88.78	75.73	76.58	73.16	73.73	55.74

a. 'Other categorical variable': load factor in (iii) and (iv), tariff in (v) and (vi), SIC in (vii).

The introduction of day of the week in experiment (ii) yields a notably better model fit (comparing to (i)). Recall that MARS adds basis functions in pairs. The first appearance of the categorical variable d in the procedure is as the 27th and 28th basis functions are selected (out of 90 non-constant basis functions in total), and unsurprisingly the categorical basis functions represent {Mon, Tue, Wed, Thu, Fri} and {Sat, Sun}, i.e. weekday and weekend. The next selection of d is upon the addition of the 35th and 36th basis functions. More surprisingly, those basis functions represent {Mon, Tue, Wed} and {Thu, Fri, Sat, Sun}. The next four times d is selected, the split is again into weekday and weekend, and the next (and final) selection of d is for the 81st and 82nd basis functions, the split (knot) being between {Thu, Fri} and the other days.

The grouping of {Thu, Fri, Sat, Sun} at the 35th/36th selection does not infer that Thursday and Friday load/weather behaviour is more like Saturday and Sunday load/weather behaviour than Monday to Wednesday behaviour: since the {Sat, Sun} basis function is already part of the model, the new {Mon, Tue, Wed} term could be seen as differentiating between {Thu, Fri} and {Mon, Tue, Wed, Thu, Fri}. However, there is no obvious reason why Thursday and Friday are grouped together. Friday load shape is

known to be slightly different from Tuesday to Thursday load shape (as is Monday load shape) as a result of proximity to the weekend, but Thursday holds no such distinction (except in the rare case when Friday is a bank holiday). In fact, it was eventually discovered that there is some highly questionable data in the original whole loads database for one particular month (January 1997), in which Thursday and Friday (and probably Saturday and Sunday) loads are almost certainly misrecorded. This is discussed in 13.5.1, and entirely accounts for the unusual grouping of Thursday and Friday.

Variables which appear in interaction terms together with d (all of order 3) are closeness to evening and smoothed temperature (48 hour half-life); day of year and smoothed darkness (1 hour half-life); time of day and smoothed darkness (half-life 2 hours); closeness to evening and smoothed darkness (half-life 1 hour); and day of year and humidity (smoothed with half-life 1 hour).

In (iii) and (iv) there is (understandably) more variation per observation (GCV), since each observation in (i) is now replaced by several generally differing observations. The modelling task is accordingly harder, and it would not be expected that the models built on this data could account for as much variation as the models of (i) and (ii). Indeed, model (iii) (featuring load-factor as a categorical variable) accounts for 75.73% root-percentage of GCV in the piecewise-linear model, as compared to 87.64% in (i) and 88.78% in (ii). When day-of-the-week is also admitted in (iv), the figure rises to 76.58%. However, while we cannot fairly compare the importance of load factor to the importance of day-of-the-week, i.e. models (ii) and (iii), using GCV scores, it should be noted that load factor was featured in a basis function more often, and at an earlier stage, in (iii) than day-of-the-week was in (ii). In fact it was picked 8 times (rather than 7) and was first used in the 9th and 10th basis functions. The first time load factor is used, it is load factor category 1 that is separated from categories 2 to 7 (category 1 comprising those customers with the lowest load factors). The second time, it is categories 5, 6 and 7 that are separated from the others.

Much later, some rather more unusual splits occur: the 75th and 76th basis functions introduce a split between categories {2, 3, 7} and {1, 4, 5, 6}. We would expect load factor categories nearby each other to exhibit similar load/weather behaviours, given enough data, since load factor is a continuous quantity. Since various basis functions featuring splits separating only contingent categories are already in the model, terms which group together non-contiguous categories could in fact represent a valid modification to the earlier terms. However, terms grouping non-contiguous categories could also arise

due to some coincidental similarities between the customers in non-contiguous load factor categories, indicating possible overfitting.

When day-of-the-week and load factor are allowed to compete in the same model, in (iv), load factor is selected 8 times (again) and day-of-the-week 4 times. Day-of-the-week is selected rather later than load factor, and groups together Saturday and Sunday 3 times, and Thursday and Friday once. Load factor shows even less tendency to group together non-contingent categories than in (iii). Load factor is rated by MARS's relative variable importance criterion as more important than day-of-the-week, across experiments (ii)-(iv).

In (v) and (vi), using tariff code as a categorical predictor, there is only slightly more GCV in the unmodelled data than in (iii) and (iv), but the variation in this data would appear to be rather harder to model (allowing the use of tariff code) than that of (iii) and (iv) (allowing the use of load factor), given that the root percentage of GCV accounted for by these models is rather lower than that accounted for by the corresponding load factor models. Also, tariff code is rated by MARS as having lower relative variable importance than when compared to load factor category (though that comparison is across different models). However, tariff code is actually selected for inclusion in new basis functions more frequently than was load factor category. In (v), tariff code was selected 12 times, 5 of these occurring quite early in the forward selection algorithm (between the selection of the 15th/16th and 31st/32nd basis functions, inclusive). These basis functions split the tariff codes into a wide variety of binary partitions. In (vi), tariff code was selected 8 times, and day-of-the-week was selected 5 times, for inclusion in new basis functions. Day-of-the-week is split along weekday/weekend lines in four of these instances, and into {Thu, Fri} and {Mon, Tue, Wed, Sat, Sun} in the other instance (on the addition of the 77th and 78th basis functions). Day-of-the-week is first selected for the addition of the 33rd and 34th basis functions, at which stage tariff code has already been selected for inclusion on five occasions.

Tariff code is rated by MARS's relative variable importance criterion as more important than day-of-the-week. However, comparing across experiments, tariff is rated as less important than load factor; day-of-the-week is rated as more important when used in conjunction with tariff code data than in any of the other categorical variable experiments, indicating that there is some interaction between tariff code and day type in their effect on the weather.

Because computational complexity considerations constrained experiment (vii) to

only 40 candidate basis functions, what we can say about it's effectiveness as a categorical predictor in load/weather models is limited. Since the data presented to the model featured (understandably) much greater variation (GCV) than that presented to the other models, we could not fairly compare the amount of GCV accounted for with GCV accounted for in the other models even if as many as 90 basis functions had been selected. However it would appear that SIC code is quite a useful predictor in the load/weather model presented; it is rated as having high relative importance by MARS, and is selected for inclusion in basis functions 10 times even though only 20 pairs of non-constant basis functions were generated. It appears in the ANOVA decomposition in three terms, interacting with darkness (smoothed with a one hour half-life), with temperature (smoothed with a 48-hour half-life), and again with that temperature smooth in an order 3 interaction with closeness-to-noon. At least one basis function from each pair of basis functions featuring SIC survived pruning, and varied greatly in which SIC codes were grouped together, as can be seen in (TABLE 11.). Here, the 41-character strings feature a zero in

TABLE 11.

00001100110011001001011100100100101110010
00001100010011100001011100100100101010010
00000000000010000000000000000000000010000
11110011001100110010100100110110110010000
00001100010010000001001000000000001010010
00000010101011000101010110000110010000011101
10010001011110111011111111100100100110010
10111000100111100110101111101111101011001
01100101001001100000001100100110111010000
11111111010110110111111011011011010011111

the *n* th position wherever the *n* th SIC category was in one half of the partition, and a 1 where it was in the other half, for the ten different partitions used. It appears that there is a complex relationship between SIC code, the weather and temporal variables, and weather-dependent load.

Due to computational complexity considerations, no MARS model was built featuring SIC code and day-of-the-week as categorical predictors together. Whilst it appears that SIC code might be a very powerful predictor in load/weather models, in order to perform more thorough experiments (with more candidate basis functions), a server with more random access memory and/or swap space, and ideally with a faster processor,

would be required than was available. Another approach might be to use a decision tree clustering technique as outlined in 13.3 and 13.4 on weather-dependent profiles, using SIC code as the only extrinsic variable; this would cluster together SIC codes for which the weather dependent profiles are similar. If, say, a dozen or so clusters of SIC codes were used in place of the original 41 SIC codes, the computational complexity and memory requirements would become more manageable, though it can only be speculated as to how useful a predictor such a clustered SIC variable could be.

It should be noted that in the 1995/6/7 data (and, indeed, the 1994/5 data) some SIC codes only feature one customer, or very few customers. Whilst the weighting scheme described in sections 11.2.2 and 11.2.3 ensures that such SIC codes do not assume undue importance in the model, the predictive power of a weather model using SIC as a predictor may be relatively poor when the previously unseen customer has an SIC code for which there was little data in the original (training) database.

Comparing relative variable importances of the continuous weather and temporal variables across all of experiments (i)-(vi) (comparisons with experiment (vii) are not really valid because of the smaller number of basis functions), there are few patterns to be noted. Generally, it would seem that yesterday's statistics, and that humidity and windspeed (though not wind chill), assume somewhat less importance as more categorical variables are introduced.

There are some weather variables that seem to assume more importance in the presence of one particular categorical customer variable; notably, wind chill gains in importance when tariff code is present as a predictor; and spot temperature is only included in the model when load factor is present as a predictor. There are no obvious explanations for these two observations.

11.3.12 Final Models Used for Deweathering Data

Based on insights drawn from the results of the experiments in sections 11.3.6 to 11.3.11, three final load/weather models were built; all were built using data weather-separated according to all four categorical variables (day-of-the-week, load factor category, tariff code and SIC code), as in section 11.3.11; two were used to create nominally weather-free databases (with and without the use of categorical variables), whilst one model (without categorical variables) was restricted to use interaction terms of order no higher than two, for the purposes of generating 3 dimensional surface plots of various

ANOVA terms. (MARS provides a slicing option to generate lower dimensional terms from higher dimensional terms by fixing one or more variables - see section 11.2.2 - to enable plotting of high dimensional interaction terms. The plots so produced were not found to be particularly revealing, and it was felt that restricting the maximum interaction order produced a set of plots with better interpretive power.)

The first model, *model* α, uses 49 weather and temporal variables, generally those which had been rated as most important in the previous experiments. One exception to this is that time-of-day and day-of-year were excluded from the final models, despite usually being rated as fairly important in earlier experiments. The reasoning behind this was that time-of-day and day-of-year are suspected of being more prone to overfitting the data than the sinusoidal temporal variables (closeness-to-noon, -evening, -spring and -summer); this is because time-of-day is capable of singling out one particular half-hour, and day-of-year is capable of singling out one particular day, whereas a particular value of one of the sinusoidal variables corresponds very closely to two non-consecutive half-hours or days (except at its peak or trough). Thus the sinusoidal variables are less likely to overfit to an outlying load value on a particular day and half-hour. Also, sunset time was omitted from the final models, as it is so strongly correlated with closeness-to-summer as to serve no real additional value (their correlation coefficient is 0.9952).

No discomfort variable was used, though wind chill was included, as were relative humidity and windspeed, each with a small selection of short-term smoothed versions. Illumination-related variables consisted of cloud and estimated darkness (transparency parameter CT=0.75), with short-term smooths. Finally a wide range of temperature-based variables (including yesterdays mean/minimum/maximum, and various lags, smooths and deltas that had proved of use in other experiments), and as always a noise variable, were included. These are all detailed in (TABLE 12.). Parameters to the MARS model were changed only slightly from experiment 1; the smoothness parameter d was set to $d = 1.00$ (in close accordance with the value determined by cross-validation in experiment 3) and the number of candidate basis functions was increased from 90 to 120 (as in experiment 3). It was not thought necessary to perform a new cross-validation experiment for this model, the set of candidate variables being very close to that of cross-

validation experiment 3.

TABLE 12.

				Model Version:	α	β	γ
				Maximum Interaction Order:	3	3	2
No.	Type	Subtype	1/2-hrs	Y-Correlation	Relative Variable Importance		
2	avey	*	*	-0.4663	19.94		10.41
3	chil	smooth	1	0.4808			
4	chil	smooth	2	0.4846			
5	chil	smooth	4	0.4857			
6	chil	smooth	6	0.484		19.27	19.43
7	chil	*	*	0.4753			
8	clou	smooth	1	0.1168			
9	clou	smooth	3	0.1187			
10	clou	smooth	5	0.1213			7.414
11	clou	*	*	0.1149			
12	dark	smooth	2	0.2718	8.377	31.19	24.11
13	dark	smooth	4	0.3404	23.48	50.81	29.14
14	dark	*	*	0.2044	22.96	5.463	
15	even	*	*	-0.02271	100.0	100.0	100.0
16	humi	smooth	1	0.3152			33.99
17	humi	smooth	2	0.3257			
18	humi	smooth	4	0.3366	23.85		
19	humi	smooth	6	0.3392	8.998		
20	humi	*	*	0.3051	6.715	17.45	
21	maxy	*	*	-0.4648	12.12		16.93
22	miny	*	*	-0.4469	12.07		
23	nois	*	*	0.01321			
24	noon	*	*	0.003568	44.81	47.31	39.19
25	seas	*	*	0.09136	23.27	28.95	12.02
26	spri	*	*	0.1899	49.89	63.91	49.99
27	summ	*	*	-0.5024	49.59	44.97	31.55
28	temp	smooth	1	-0.4917			
29	temp	smooth	2	-0.4954			
30	temp	smooth	4	-0.4968			17.45
31	temp	smooth	6	-0.4945	28.90		
32	temp	smooth	12	-0.4876		27.87	24.00
33	temp	smooth	48	-0.4867	20.81	19.80	17.46
34	temp	smooth	96	-0.4914	38.10	50.42	26.19
35	temp	smooth	336	-0.4864	11.52		9.117
36	temp	smooth	672	-0.4584	19.75		
37	temp	smooth	1344	-0.3805	30.96		29.64
38	temp	smooth	2688	-0.2534	29.55	48.05	21.47

TABLE 12.

39	temp	delta	-2	0.05979			
40	temp	delta	-8	0.001169			
41	temp	delta	-24	-0.1372			
42	temp	delta	-48	-0.02625			
43	temp	delta	-336	-0.009455	8.745		
44	temp	lag	-8	-0.4882			
45	temp	lag	-48	-0.4796	5.920		
46	temp	*	*	-0.4875	20.87	14.85	
47	wind	smooth	1	0.07613			
48	wind	smooth	3	0.08026			9.967
49	wind	smooth	5	0.08996	23.65		
50	wind	*	*	0.07682			
51	day of week	*	*	*	*	37.20	*
52	load factor	*	*	*	*	97.73	*
Number of observations:					32 112	224 784	32 112
GCV in unmodelled data:					62.52	99.76	62.52
GCV score (piecewise linear):					12.72	39.80	13.75
GCV score (piecewise cubic):					14.37	41.45	15.17
Root percentage GCV accounted for (piecewise linear model):					89.25	77.53	88.32

It would have been viable to increase the maximum interaction order from 3 for the final model α, but this was decided against. Whilst MARS weather models which allow unrestricted interaction orders were tested (not presented), and did sometimes select interaction terms of order 4 and even 5, there is no compelling evidence that terms of order greater than 3 play any particularly significant role in the load/weather relationship. Allowing a model additional freedom when there is not a compelling reason to do so increases the chances that the model will overfit the data, thus the maximum interaction order was left at 3.

Model α does not select cloud cover or wind chill in any form. Estimated darkness, relative humidity and windspeed all assume moderate importance, as do the previous day's temperature statistics. Whilst seasonality's role is modest, the other temporal variables, especially closeness-to-evening, are all rated as highly important.

A one-day lagging temperature variable played only a tiny role, and the only delta variable selected, with a lag of one week, played almost as small a role. It appears that (at least in the presence of temporal variables) temperature lags and deltas are of little to no importance, as compared with exponentially smoothed temperatures. As previously, a wide range of temperature smooths were found to be important, as well as the spot tem-

perature.

Of 53 ANOVA functions in the final model α, 8 were functions of one variable, 18 were interactions of order two, and the remaining 27 were interactions of order 3. There are many interactions between the various temporal variables and the various temperature variables; humidity appears in order 3 interactions with windspeed and a long term temperature smooth (half-life two weeks); with windspeed and a short term temperature smooth (half-life three hours); with closeness-to-evening and a long term temperature smooth (half-life 1 week); and interestingly, with seasonality and a long-term temperature smooth. Since seasonality peaks at both midwinter and midsummer, this may be indicative of humidity having an effect on winter heating loads as well as summer cooling loads. Aside from interactions with humidity already mentioned, wind speed occurs in only one other interaction term (of order 2, with a short term temperature smooth). Estimated darkness appears in several interactions, always with only temporal variables. The previous day's temperature statistics appear in interactions with a variety of temporal variables and with temperature (spot temperature, and short, medium and long term temperature smooths).

In *model* β, the same candidate variables were used, plus two categorical variables (load factor category and day-of-the-week). The parameters were unchanged, except for MARS smoothness parameter d, which was determined by a new cross-validation experiment. Because of the higher computational cost of the weather model in the presence of categorical customer variables, tenfold cross-validation (as in experiment 3) was rejected in favour of single-fold validation, which is less accurate but much faster. This involved partitioning the population randomly into two sets (a training subset and a test subset), with $\frac{4}{5}$ of the data in the training subset and $\frac{1}{5}$ of the data in the test subset. Part of the output from this cross-validation run appears in (TABLE 13.).

TABLE 13.

Number of basis functions	Corresponding d	Cross-Validation Errors (CV)
120	-1.70	39.26
...	...	...
102	-1.68	39.26
101	-1.66	39.26
100	-1.36	39.26
99	-0.73	39.26
98	-0.31	39.26

TABLE 13.

97	0.79	39.27
96	2.17	39.27
95	5.36	39.27
...	...	...
80	62.36	39.37
...	...	...
50	513.36	41.36

As in experiment 3, the cross validation errors are very insensitive to changes in the number of basis functions retained in the final (pruned) model over quite a wide range. The optimal number of basis functions was determined by MARS to be 101, with a corresponding smoothness parameter $d = -1.66$, representing only a tiny penalty for the addition of new basis functions in the lack-of-fit function (EQ 59), (EQ 60). Using this value for d, the model β (rebuilt on the full dataset, not just the training data) is described in (TABLE 12.). Any differences from the results of experiment (iv) of section 11.3.11 (which used a similar variable set) are very slight.

Finally a model (*model* γ) was built with the same variables and parameters as in model α, except that the maximum interaction level was restricted to 2, for easier visual interpretation. There are some differences in relative variable importances when the maximum interaction level is set at two (compared to model α), and the overall model fit is somewhat poorer, but the simplified model is basically similar and allows much more comprehensible visualisation by avoiding the need to slice order 3 terms. (COLOUR FIGURE 5.) to (COLOUR FIGURE 6.) show various surface plots for ANOVA terms from the simplified order 2 model. The x- and y-axes are always scaled so that the minimum value of a predictor is 0.0 and the maximum value is 1.0. The z-axis is scaled by adding or subtracting a constant so that the smallest value of the fitted surface is 0.0; however the highest value shown on the z-axis is indicative of the actual magnitude of the illustrated ANOVA function in the final model. To make interpretation easier, points are plotted in a colour with varies smoothly with the fitting function (i.e. z-axis). The same colour gradient is used as in (COLOUR FIGURE 1.) and similar figures, with deep purple being used for the minimum of the fitted function, through to bright red for the maximum of the fitted function. Regions of the x-y plane in which no data points occur do not have a valid fitting function in a MARS model, and so remain blank.

It must be pointed out that in the presence of so many correlated predictors, no single ANOVA term for any two predictors can be properly interpreted in isolation from all the

other ANOVA terms featuring related predictors. Nevertheless the presented plots are instructive.

(COLOUR FIGURE 5.) shows an ANOVA term featuring a 48 hour half-life temperature smooth with closeness to evening. Notice that the term increases for both low temperatures and high temperatures, more so near evening. (COLOUR FIGURE 6.) features the same temperature smooth together with closeness to summer. The two main peaks occur where there are high temperatures in summer (top of diagram), and where there are low temperatures distant from summer (centre right of diagram).

(COLOUR FIGURE 7.) features an interaction term involving estimated darkness (smoothed with half-life 2 hours) and closeness to evening. The fitted surface peaks when it is very dark near evening, and smoothly falls as the darkness level or the closeness to evening falls.

(COLOUR FIGURE 8.) features wind chill and closeness to summer. It is interesting that as well as a peak associated with high wind chill factors, which is higher near winter, there is a lesser peak associated with low wind chill factors, but only near to summer. Since in summer a low wind chill factor is associated with warm, still, days this is evidence that the model accounts for the increased air conditioning loads on hot days that result from stiller air (since opening a window will provide little relief from discomfort when the air is still).

Chapter 12 — Visualisation of Load Profiles

12.1 Introduction

Discrete predictors collectively have complex interactional effects on loads. By averaging only those atomic profiles for which a specified predictor takes some specified value, we obtain a marginal profile (for example the Sunday profile). By subtracting the average profile from a marginal profile (and scaling) we obtain an effect profile, which indicates how a marginal profile differs from the typical profile. These ideas are extended to conditional profiles (for example, the January-Thursday profile, and the effect profile of Thursdays on the January profile), which indicate interactional effects of discrete predictors.

These objects are useful as explanatory devices; plots of these profiles provide a handy visualisation of the local or global nature of the relationship between load shape and discrete predictors.

A simple means for the visualisation of half-hourly load profile over several months at one glance is suggested.

12.2 Basic Marginal, Effect and Difference Profiles

During this chapter one can assume that weather modelling has already been performed, producing weather-free loads $Y_i^F(t)$ (and corresponding profiles $\underline{Y}_i^F$), though all the concepts could be applied equally to whole (i.e. not deweathered) loads. Recall the weather-free atomic model AFM, here repeated:

$$\underline{Y}_i^F = \hat{f}_{AF}(\underline{X}^F(i)) + \underline{\varepsilon}_i^{(mus)} + \underline{\varepsilon}_i^{(mod)} \qquad 1 \le i \le N \qquad \text{(EQ 103 : AFM)}$$

whose atoms derive from a maximal partition on $\underline{X}^F = (d, m, X_5, X_6)$.

The *marginal profiles* for a predictor X_j in $\underline{X}^F$ are simply the centroids of the atoms for which $X_j = x_{jl}$ (for each $x_{jl} \in \underset{\sim}{X}_j$). If C is the set of all indexes for atomic profiles ($C = \{1, 2, \ldots, i, \ldots, N\}$), let $\tilde{C}_j$ be the partition on C arising from splitting on X_j, which has cells $\tilde{C}_{j(1)}, \ldots, \tilde{C}_{j(|\underset{\sim}{X}_j|)}$. Denote the number of elements in the lth cell of this partition n_{jl}. Note that

$$\sum_{l=1}^{|\underset{\sim}{X}_j|} n_{jl} = N \qquad \text{(EQ 104)}$$

Denote the *marginal profile for* x_{jl} by $\underline{Y}|x_{jl}$, where

$$\underline{Y}|x_{jl} = \frac{\sum_{i \in \tilde{C}_{j(l)}} \underline{Y}_i}{n_{jl}} \quad \text{(EQ 105)}$$

A simple way to isolate the effect of one of the predictors X_j in $\underline{X}^F$ on profile shape is to compute the *effect profiles* for that variable. Denote the *effect profile for* x_{jl} by $\underline{E}(x_{jl})$, where

$$\underline{E}(x_{jl}) = n_{jl}\,(\,(\underline{Y}|x_{jl}) - \underline{Y}) \quad \text{(EQ 106)}$$

$\underline{Y}$ being the grand centroid of all the atomic load profiles,

$$\underline{Y} = \frac{n_{j1}\underline{Y}|x_{j1} + \ldots + n_{j|\underline{X}_j|}\underline{Y}|x_{j|\underline{X}_j|}}{N} \quad \text{(EQ 107)}$$

Note that loads in effect profiles can take both positive and negative values. The various effect profiles for a predictor X_j sum to $\underline{0}$ (the zero profile),

$$\underline{E}(x_{j1}) + \ldots + \underline{E}(x_{j|\underline{X}_j|}) = \underline{0} \quad \text{(EQ 108)}$$

We can also plot *difference profiles* for any two profiles $\underline{P}_1, \underline{P}_2$; like an effect profile, a difference profile can take positive or negative values, and requires that both profiles are first normalised (by each dividing through by the number of raw profiles thereat); thus a difference profile for $\underline{P}_1, \underline{P}_2$ is given by

$$diff(\underline{P}_1, \underline{P}_2) = \frac{\underline{P}_1}{|\underline{P}_1|} - \frac{\underline{P}_2}{|\underline{P}_2|} \quad \text{(EQ 109)}$$

Where $|\underline{P}_1|$ and $|\underline{P}_2|$ are the respective number of underlying raw profiles making up a particular profile. Thus we can plot the difference profile between a customer's weekend profiles and their weekday profiles, or between a group of customers' December to March profile and their April to November profile, for instance, for comparison purposes.

Plots of effect profiles give at-a-glance indications of the effect of a particular attribute taking a particular value, on load. For example the $\underline{E}(x_{37})$, i.e. $\underline{E}$(Sun) effect profile represents the effect on load of the day-type being a Sunday. We would obviously expect this effect profile to take mostly negative values, since Sunday loads are lower

than average.

12.3 Conditional Marginal and Effect Profiles

As thus far described, the marginal and effect profiles convey the influence of particular predictor values on total (deweathered) load for the database. However, the discrete predictors $\underline{X}^F$ have interactions; for instance the effect of (say) month index on (say) Monday profiles is not necessarily the same as the effect of month on (say) Saturday profiles, nor the same as the effect of month on the all-days profile. The concepts of section 12.2 are extensible in a natural manner to *conditional* marginal profiles/ effect profiles. Suppose X_j is first used to partition the set of atomic profiles (indices C), and subsequently $X_{j'}$ $(j \neq j')$ is used to subpartition a cell $\tilde{C}_{j\,(l)}$ of $\tilde{C}_j$. We need only apply the definitions of section 12.2 with $\tilde{C}_{j\,(l)}$ in place of C to obtain the conditional versions. If $\tilde{C}\,\{x_{j'l'};x_{jl}\}$ is the l'th cell in the subpartition of $\tilde{C}_{j\,(l)}$, define the *marginal profile for* $x_{j'l'}$ *conditional on* x_{jl}, denoted $\underline{Y}|x_{j'l'};x_{jl}$, by

$$\underline{Y}|x_{j'l'};x_{jl} \;=\; \frac{\displaystyle\sum_{i \,\in\, C\,\{x_{j'l'};x_{jl}\}} \underline{Y}_i}{n_{j'l';jl}} \qquad \text{(EQ 110)}$$

where $n_{j'l';jl}$ is the number of constituent atomic profiles, $\left|\tilde{C}\,\{x_{j'l'};x_{jl}\}\right|$. Similarly, the *effect profile for* $x_{j'l'}$ *conditional on* x_{jl} is given by

$$\underline{E}(x_{j'l'};x_{jl}) \;=\; n_{j'l';jl}\,(\,(\underline{Y}|x_{j'l'};x_{jl}) - \underline{Y}|x_{jl}) \qquad \text{(EQ 111)}$$

The conditional effect profiles again contain both positive and negative values. They give indications of the effect of some attribute taking a certain value, conditional on some other attribute taking a certain value. For example we might wish to compare $\underline{E}(\text{Jan};\text{Sun})$, the effect that the day being Sunday has on January profiles, with $\underline{E}(\text{Jan};\text{Mon})$, the Monday-effect on January profiles, or with $\underline{E}(\text{Oct};\text{Sun})$, the Sunday-effect on October profiles.

Note that in marginal conditional profiles, the order in which variables are selected for splitting is irrelevant:

$$\underline{Y}|x_{j'l'};x_{jl} \;=\; \underline{Y}|x_{jl};x_{j'l'} \qquad \text{(EQ 112)}$$

for in-range choices of j, l and j', l'. However, the order of variables does affect con-

ditional effect profiles: in fact

$$\underline{E}(x_{j'l'};x_{jl}) - \underline{E}(x_{jl};x_{j'l'}) = n_{j'l';jl}\,(\underline{Y}|x_{j'l'} - \underline{Y}|x_{jl}) \qquad \text{(EQ 113)}$$

12.4 Multivalue Marginal and Effect Profiles

A final extension to the ideas and notations of 12.4 and 12.5 allows marginal and effect profiles (conditional or otherwise) to be calculated for disjunctions of predictor values. For example, we might want to know the effect profile of weekend day-types on the profiles of customers whose two-digit SIC codes start with a '5'. This might be written $\underline{E}(d \in \{\text{Sat,Sun}\};X_5 \in \{50, 51, \ldots, 59\})$. The formal definitions for such profiles are obvious extensions of the definitions in sections 12.2 and 12.2; for instance, replace partition $\tilde{C}_3 = \{\tilde{C}_{3(1)}, \ldots, \tilde{C}_{3(5)}, \tilde{C}_{3(6)}, \tilde{C}_{3(7)}\}$ with partition $\tilde{C}'_3 = \{\tilde{C}_{3(1)}, \ldots, \tilde{C}_{3(5)}, \tilde{C}_{3(6)} \cup \tilde{C}_{3(7)}\}$ in order to construct a marginal profile for 'weekend' rather than separate marginal profiles for 'Sat' and 'Sun', and so on. An alternative notation for marginal and effect profiles of this type uses disjunctions of predictor values rather than set memberships (where no confusion will arise), so that the above cited example of a multivalue effect profile is written $\underline{E}(\text{Sat} \vee \text{Sun};50 \vee 51 \vee \ldots \vee 59)$.

12.5 Visualisation of Seasonally Varying Daily Load Shape

12.5.1 Time-of-Day/Time-of-Year Visualisation

A two dimensional plot of load against hour of day visualises load shape for a single day. However, by using colour to represent an extra dimension, the daily load shape can be visualised at a glance over a period of many months. Let the horizontal axis of a Cartesian graph measure the day index over the period for which a visualisation is required, whilst the vertical axis measures time of day (from $t = 1$ to $t = 48$). For a specified set $\{\underline{Y}_{d^*};d^*_{\min} \le d^* \le d^*_{\max}\}$ of atomic, marginal or effect profiles indexed by day index d^* over the range of days $[d^*_{\min}, d^*_{\max}]$ [1], let $\underline{Y}_{dt}$, $1 \le t \le 48$ be the half hourly loads. Linearly scale these loads so that the lower bound for load has value 0.0 and the

1. As described in sections 12.2 to 12.4, d^* is not a disaggregating variable for marginal or effect profiles, but the concepts of marginal and effect profiles are equally applicable when d^* is included as a predictor. Month index m would be a viable alternative to d^*.

upper bound for load has value 1.0. Let $colour(\,)$ be a function of a real number which associates a colour with real values in $[0.0, 1.0]$, and also two 'out of bounds' colours for values less than 0.0 or greater than 1.0. The function must be designed so that the hue changes smoothly with the argument between 0.0 and 1.0. In the scheme that was selected, the colour varies from deep purple (for low loads), through blue to cyan (the centre point, 0.5), to green to yellow to orange to bright red. There are two out-of-bounds colours; transparent (i.e. white) for loads below a lower bound, and wheat for loads above an upper bound. By plotting small non-overlapping rectangles of the colour associated with the scaled load at each of the $(d^*_{\max} - d^*_{\min}) \times 48$ coordinates, a larger multi-coloured rectangle is produced.

Such a plot simultaneously conveys load/time-of-day behaviour and load/day-of-month-or-year behaviour. Naturally, conventional 3 dimensional surface plots of load against time of year against time of day could also be used to present this behaviour, though problems arise because many features are frequently hidden from view in a surface plot since load can vary so rapidly from day to day and hour to hour; load does not always vary smoothly with time-of-day or day of year, though surface plots are somewhat better than the coloured type of plot at presenting responses which vary only gradually.

Various such plots are illustrated in the Appendix (Colour Figures). A key is provided with all such plots to indicate what percentage of average half-hourly load is represented by the different colours. Such plots can be calculated for many customers' profiles averaged together as in (COLOUR FIGURE 10.) or for individual customers, as in (COLOUR FIGURE 1.) to (COLOUR FIGURE 4.).

12.5.2 Customer/Time-of-Year Visualisation

A variant of this visualisation technique was also implemented which allows a whole database to be visualised in such a way that differences between customers are apparent. Instead of using time-of-day on the vertical axis, a customer index is used (running from 1 up to the number of customers in the database). Instead of half-hourly loads being plotted as small coloured rectangles, daily load averages (for a given customer and given day index) are plotted as appropriately coloured small rectangles). Customer index is not a (meaningful) continuous variable, and customers plotted side by side in such a diagram may have nothing in common, but such a diagram is still useful as an at-a-glance guide

to the extent of variability in daily load between customers and seasonally; such a diagram is also useful for visualising the extent to which load profiles are missing in a database, and for which days.

All of the 1995/6/7 whole (i.e. not deweathered) load profiles database is displayed in such a plot in (COLOUR FIGURE 9.). Here, transparent (white) represents missing load profiles, and wheat represents a daily load total that is 200% or greater of the customers average daily load total, with cyan representing average daily load total (i.e. 100%).

Chapter 13 — Model for Deweathered Loads

13.1 Discussion

Recall the atomic weather-free model AFM of section 10.6, repeated here:

$$\underline{Y}_i^F = \hat{f}_{AF}(\underline{X}^F(i)) + \underline{\varepsilon}_i^{(mus)} + \underline{\varepsilon}_i^{(mod)} \qquad 1 \le i \le N \qquad \text{(EQ 114 : AFM)}$$

whose atoms derive from the maximal partition of $\underline{X}^F = (d, m, X_5, X_6, X_L)$, the day type, month, SIC code, tariff code and load factor category. Since the load data for this model are the centroidal load profiles[1] for the N atoms $\underline{A}_i$, and the predictors for the model are exactly those predictors used to construct the partition defining the atoms, trivially the lowest error option for $\hat{f}_{AF}$ is that which returns the untreated weather free model data (the training data):

$$\hat{f}_{AF}(\underline{X}^F(i)) = \underline{A}_i \qquad \text{(EQ 115)}$$

Substituting this into (EQ 114 : AFM), we get

$$\underline{Y}_i^F = \underline{A}_i + \underline{\varepsilon}_i^{(mus)} \qquad 1 \le i \le N \qquad \text{(EQ 116)}$$

Note that the modelling-error term $\underline{\varepsilon}_i^{(mod)}$ completely disappears; the only errors in estimating the ‘true’ atomic profiles with the observed atomic profiles arise from measurement error, unknowable factors, and errors due to sampling. Essentially, there are N distinct, exhaustive and exclusive identity fitting functions. The measurement component of the error is assumed very small. We have estimates for the variance of the errors due to sampling and unknown factors in each atom, namely the sample variances of the atomic training data. Many of these are likely to be large, since there are relatively few profiles represented in each atom. Moreover, for atoms which represent only a few constituent profiles, the variance estimates are unreliable.

There are two major (and closely related) drawbacks to the trivial atomic weather-free model (EQ 115), (EQ 116):

1. Though the form of the fitting function is simple, the model scores very low on inter-

1. These profiles comprise aggregate square-loads as well as aggregate loads, enabling the construction of the 48 standard deviations over the constituent loads.

pretability, since there are just too many (N) distinct fitting functions within $\hat{f}_{AF}$. The data needs to be distilled into a more compact (lower resolution) model from which a planner can more easily glean the nature of the load's relationship to the discrete predictors.

2. Whilst the modelling errors $\underline{\varepsilon}_i^{(mod)}$ disappear, the errors due to unknown factors and due to sampling are high. This means that when test data (distinct from the training data) are presented to the model, fit is likely to be poor. If we merge atoms so that their underlying partition cells contain more data, these errors may be reduced (provided that the atomic profiles merged have similar distributions). Again, what is required is a lower resolution model.

Opting for a lower resolution model reintroduces modelling errors $\underline{\varepsilon}_i^{(mod)}$ but is intended to reduce errors due to sampling, and hence $\underline{\varepsilon}_i^{(mus)}$. Ideally, the errors we aim to minimise are the *cross-validation errors*; these are the total errors arising when a model built using training data is fed unseen test data. If the fitting function $\hat{f}_{\text{train}}$ is built from training data, then the cross validation errors $\underline{\varepsilon}_i^{(CV)}$ are given by

$$\underline{\varepsilon}_i^{(CV)} = \underline{Y}_i^{(test)} - \hat{f}_{\text{train}}(\underline{X}^{(test)}(i)) \qquad \text{(EQ 117)}$$

with $\underline{X}^{(test)}(i)$, $\underline{Y}_i^{(test)}$ the test data (predictors, responses).

A slightly different way of looking at the trivial model and its drawbacks is in information theoretic terms. To represent the fitting function $\hat{f}_{AF}$ inside a computer, the N profiles need be stored. This is the maximum entropy representation (because it is at the highest resolution allowable), but is also very bulky. By merging certain atoms so that $P < N$ profiles are stored we reduce the bulk of the data (by a factor N/P), but also lose information (entropy). We wish to minimise bulk whilst maximising the information retained. Recalling that information in some data can be thought of as the size (in bits) of the theoretical smallest equivalent coding (reversible data compression), we can cast this problem as maximising theoretically optimal storage size (i.e. information) for the model whilst reducing the actual storage size (bulk) P. Since high signal-to-noise representations carry more theoretical information than low signal-to-noise representations, errors $\underline{\varepsilon}_i^{(mus)} + \underline{\varepsilon}_i^{(mod)}$ are implicitly kept low by this criterion (cross-validation errors could also be computed, their minimisation being an auxiliary criterion to any bulk-to-information criterion).

Whatever the precise criterion we use to guide the choice of partition underlying $\hat{f}$, this is an atomic cluster analysis task (see section 8.8.2). Extrinsic decision tree clustering (also 8.8.2), a subspecies of atomic clustering, is a viable option. The patterns to be clustered are the atoms $A_i; (1 \leq i \leq N)$: they have predictor values $\underline{X}^F(i)$ and response values $\underline{A}_i$ $(=A_{it}; (1 \leq t \leq 48))$. We can think in terms of partitioning the patterns (notation of chapter 8), or of partitioning predictor space $\underset{\sim}{X}^F$ (notation of chapter 4) interchangeably, but it will be somewhat more natural for us to use the latter notation.

Note that the centroids for the various clusters which can arise in an atomic clustering of profiles are unnormalised versions of the marginal and conditional marginal profiles introduced in Chapter 12.

13.2 Atomic Clustering for Weather-Free Profiles

Recall from 8.8.2 that atomic clustering is simply a clustering of the centroids of the sets of patterns indistinguishable by their predictors; any clustering algorithm can be used to obtain the atomic clustering. Due to the large size of N (number of atoms) in the monthly billed business customer database, it is very computationally demanding to perform a graph theoretic clustering (which requires the $\frac{1}{2} N(N-1)$ distances between atoms to be calculated in advance). This task could be made much smaller by first applying a conservative one-pass clustering (see section 8.4) to the atoms (that is, one which leaves quite a large number of clusters). The clusters discovered thus could then be used as the patterns to be clustered in a graph theoretic method.

The most flexible type of clustering algorithm we can apply to the atoms is probably *K*-means clustering, in any of its varieties. The computational complexity for this set of algorithms is rather dependent on the particular algorithm used: a simple *K*-means clustering, seeded by a simple one-pass clustering, is not so demanding computationally as a graph theoretic approach.

Having applied any clustering algorithm to the atomic data, we are left with $P \ll N$ cluster centroids $\overline{C}_{(1)}, \ldots, \overline{C}_{(p)}, \ldots, \overline{C}_{(P)}$ in place of the original data as the output of $\hat{f}_{AF}$; $\hat{f}_{AF}(\underline{X}^F(i))$ simply returns the cluster centroid which A_i has been incorporated into. Thus the atomic clustering has reduced the initially vast number of load profiles in the database to P representative profiles; P can be directly or indirectly adjusted to be as small as desired. However, the partition of the atoms (whether simple or hierarchical) is not based on the actual predictor values $\underline{X}^F(i)$, which are used only in determining the

atoms.

13.3 Extrinsic Decision Tree Clustering for Weather-Free Profiles

13.3.1 Top-Down Variable-By-Variable Clustering

Now suppose that the clustering is based on a hierarchical partition of the predictor space $\underset{\sim}{X}^F = \underset{\sim}{d} \times \underset{\sim}{m} \times \underset{\sim}{X}_5 \times \underset{\sim}{X}_6 \times \underset{\sim}{X}_L$, in which at every splitting stage, the partition or subpartition is based on the values of one of the predictor variables in $\underline{X}$. If the selected variable takes v distinct values within the current cell to be split, this defines v cells in the subpartition, whose centroids are treated as the v patterns to be clustered. As mentioned in section 8.8.2, the problem is to select at each step (*i*) the 'best' attribute on which to partition; and (*ii*) to select the 'best' partition based on that attribute's values.

One might choose to seek the attribute and the partition simultaneously by checking all partitions (of a certain type) of all variables for the 'best', where this approach is computationally feasible (call this *top-down general extrinsic decision tree clustering*). Otherwise it is necessary to define a criterion for 'goodness of partitioning variable'. Note that the partitioning variables $\underline{X}^F$ are discrete (and categorical except for X_L which is ordered, and m which is ordered and periodic), whilst the proximity metric is between patterns of continuous responses, so a simple information theoretical as is often used in decision tree classification is not an option; information theoretical measures for load profiles may need to be approximated (see 14.4.2).

Having selected a variable X_j for partitioning, a 'good' partition is sought; this depends crucially on the resolution of the partition which is desired. If each variable is allowed to be selected more than once for partitioning, the resolution can be very coarse—a binary partition (into just two cells) is always acceptable, since either cell can be split again at a later level in the hierarchical partitioning process. Supposing that X_j takes v distinct values within the current cell to be split. Then there are $2^{v-1} - 1$ binary partitions possible $(v > 1)$. It is thus only possible to exhaustively search all binary partitions (seeking that which maximises between cluster scatter) if v is small (though note that each time a variable is reselected for splitting v becomes smaller). The binary splitting algorithm described in section 8.6 provides a non-exhaustive (generally sub-optimal) search for a binary partition on v cases. This involves searching just $\frac{1}{2} v (v + 1)$ binary partitions.

Another sub-optimal search algorithm is the *join-two* algorithm, a bottom up cluster-

ing algorithm which hierarchically partitions the v patterns. Initially the v patterns are searched to find that pair of patterns which are closest by our distance metric. They are replaced by their centroidal pattern, and at the next stage the $v - 1$ remaining patterns are searched to find the closest pair, and so on. When p patterns remain, the algorithm terminates, p being preset by the user ($p = 2$ finds binary partitions). This has complexity of order v^3, which limits the v for which it is applicable, but is less greedy than the binary splitting algorithm.

Where each variable is prohibited from being used for partitioning more than once, the user can supply appropriate numbers of clusters per variable in advance (binary partitions are unlikely to be fine enough)—this is a much less flexible approach.

If we do not cluster at all, so that there are v cells in the partition (one for every variable value) and continue to do so until all variables have been used, we arrive at the maximal partition (the highest resolution partition possible, where every atom is in its own leaf). This is one possible starting point for a bottom-up clustering method; the order of variable selection would strongly influence such a bottom-up (or tree-pruning) algorithm.

13.3.2 Top-Down General Clustering

Rather than choosing the ‘best’ variable on whose values to partition, and then the ‘best’ partition, now suppose we seek the ‘best’ partition regardless of which variable whose values it partitions. There are J-fold more partitions to be considered in general (J the number of predictors, J=5 for predictors $\underline{X}^F$). To achieve similar computational complexities compared to the variable-by-variable clustering algorithms, it would be necessary to examine fewer candidate partitions for each predictor (i.e. use a greedier algorithm; for example, the computationally more demanding join-two algorithm might be dropped in favour of the binary-splitting algorithm).

13.3.3 Termination Criteria for Top-Down Clustering

In sections 13.3.1 and 13.3.2 it was not suggested how one should determine when to stop partitioning, and accept the current set of leaf clusters as our model. Whilst locally we aim to minimise within-cluster scatter (maximise between cluster scatter), this does not help us decide when to terminate the procedure, since the within-cluster scatter is minimised globally only when we reach the maximal partition (in which every cluster

is an atom), returning the trivial model of section 13.1.

Local criteria as well as global criteria may be applied to determine when to terminate top-down clustering procedures. The most obvious global criteria are (*i*) to set in advance a maximum number $P < N$ of clusters and terminate just before this is exceeded or (*ii*) to set in advance a certain proportion of scatter S_B/S which must be accounted for by the clustering, and terminate when this is reached (proportion of scatter accounted for: see end of section 8.3).

Local criteria, which dictate when a particular cell should not be subpartitioned, but not when all hierarchical partitioning should terminate, could also be specified. The local scatter (the value of S_W/S for a particular subpartition) could be used to terminate partitioning locally when it goes above a certain preordained threshold. Note that this is sensible only when the number P of cells in the local subpartition is fixed (e.g. in binary partitioning).

Another crucial factor in determining when to cease partitioning is whether or not one intends to overgrow the decision tree, and subsequently prune it back by joining sibling clusters. If pruning is to occur, the splitting termination criterion will generally be less important than the pruning termination criterion, and must allow for overgrowing. One option is to not cease splitting until the maximal partition is reached (with atomic profiles at leaves) and to prune back from there.

13.3.4 Bottom-Up Decision Tree Clustering and Tree-Pruning Algorithms

Rather than splitting partition cells we can start with the maximal partition (whose cells comprise one atom each) and successively join cells together (in other words, combine profiles). The final partition in such a scheme is the universal cluster (which contains all the atoms). An alternative to starting with the maximal partition is to apply joining procedures to the leaf cells (leaf profiles) of a top-down decision tree clustering (this is tree pruning, see chapter 4).

Many joining algorithms for decision tree clustering are functionally equivalent to corresponding splitting algorithms, and we will not describe them. Using joining procedures for pruning decision trees may be of much value (see, e.g. [9]) however.

The simplest pruning procedures simply cut off the decision tree below a certain level. However the real power of tree pruning can sometimes lie in its local nature — pruning back overfitted or uninformative structure where it exists whilst retaining valid

structures elsewhere.

Now if the pruning criterion is identical with the splitting criterion, we are simply reversing the original top-down procedure. Thus we require a different criterion: for instance, a splitting procedure is selected which aims to minimise within cluster scatter but pays scant regard to tree size or complexity; whilst a pruning procedure is chosen which optimises tree size or complexity *and* within cluster scatter simultaneously.

13.4 An Adaptive Decision Tree Clustering Technique for Load Profiles

13.4.1 General Approach

Various clustering algorithms were incorporated within a single adaptive, top-down, binary decision tree clustering framework. Note that when the same variable is selected for splitting more than once in a decision tree procedure, the number of locally observed distinct values *v* decreases each time. Consequently, it is appropriate to apply greedier searches for good partitions nearer the root of the tree, and more exhaustive searches nearer the leaves.

The relaxation on computational complexity brought about by such adaptivity (using less expensive algorithms where the problem is locally harder) in fact makes general top-down (rather than variable-by-variable top-down) procedures viable, at least for the data sets studied.

The most expensive algorithm employed is exhaustive search (in which all possible binary partitions on a particular variable's domain are computed and compared). The cheapest algorithm employed is the Binary Splitting Algorithm (see 8.6). An algorithm of intermediate complexity, the *Join-Two* algorithm, is also employed (details follow in section 13.4.2).

Note that when partitioning on month index and load factor (ordered rather than categorical variables), the assumption than any cluster contains only contiguous categories would enable faster searches (see section 8.8.1). These were not implemented, in the end, because exhaustive search was computationally feasible for load factor X_L (with 7 load factor categories), and because month m has periodicity and special features, which means that the best clusters might not always feature contiguous months. In particular, December has special properties due to the holiday period. More generally, a spring month may be more similar to an autumn month than to a summer month.

13.4.2 Sub-Algorithms Employed in the Decision Tree Clustering Framework

Suppose N cases are to be clustered (i.e. that the variable being considered as a candidate for splitting on has N categories present at the current level in the tree).

1. *Exhaustive Binary Search*. For every possible binary clustering $\tilde{C} = \{C_{(1)}, C_{(2)}\}$, the distance $d(\overline{C_{(1)}}, \overline{C_{(2)}})$ is computed. That clustering with the greatest computed distance wins. Complexity is $O(2^{N-1} - 1)$.

2. *Binary Splitting Algorithm*. This algorithm only finds binary clusterings (exactly two clusters, $C_{(1)}$ and $C_{(2)}$). Initially $C_{(1)}$ contains all the profiles, and $C_{(2)}$ is empty. At each step, one of the profiles in $C_{(1)}$ is transferred to $C_{(2)}$. The profile selected for transfer is the profile in $C_{(1)}$ which most increases (or least decreases) the distance between the clusters, $d(\overline{C_{(1)}}, \overline{C_{(2)}})$. That distance is recorded, and the transfer process is repeated until $C_{(1)}$ is empty. That pairing of $C_{(1)}$ and $C_{(2)}$ which produced the greatest recorded between-cluster distance is the eventual winning binary partition. By recursive application, an arbitrary number of clusters can be generated. Complexity is no greater than $O(N^2)$ (see [13]).

3. *Iterative Join-Two Algorithm*. Initially there are N clusters, one singleton cluster for every profile. At each stage, the distance between each pair of clusters $C_{(j)}, C_{(k)}$ is calculated. That pair of clusters whose distance $d(\overline{C_{(j)}}, \overline{C_{(k)}})$ is the least are merged into a new cluster $C_{(jk)} = C_{(j)} \cup C_{(k)}$, then $C_{(j)}$ and $C_{(k)}$ are discarded. Distances are recalculated (where necessary) between the $N + 1 - 2 = N - 1$ resulting clusters, and again the closest pair are merged. This continues until only n clusters remain. Complexity is no greater than $O(N^2 (N - n))$, thus less than $O(N^3)$ when a binary partition is sought.

In fact, the Iterative Join Two algorithm is employed as a hybrid with Exhaustive Search; once the number of clusters falls to a threshold value n_e, an exhaustive search of all possible combinations of the remaining n_e clusters into two superclusters is performed (complexity $O(2^{n_e - 1} - 1)$).

For any given decision tree clustering, the same distance measure is used in all three sub-algorithms. This is either unweighted Euclidean distance between cluster centroids, or a modified Euclidean distance measure

$$d_b(C_{(j)}, C_{(k)}) = bias_b(|C_{(j)}|, |C_{(k)}|) \times d(C_{(j)}, C_{(k)}) \quad \text{(EQ 118)}$$

where $d(C_{(j)}, C_{(k)})$ is regular Euclidean distance $\sqrt{\sum_{t=0}^{47}\left(\left(\overline{C_{(j)}}\right)_t - \left(\overline{C_{(k)}}\right)_t\right)^2}$. $\overline{C_{(j)}}$ and $\overline{C_{(k)}}$ are the cluster centroids (i.e. each is the mean of all profiles in the cluster) as usual, $|C_{(i)}|$ is the number of original profiles that are combined into $C_{(i)}$ (where $i = j$ or k), and

$$bias_b(n, m) = (4nm / (n + m)^2)^b \quad \text{(EQ 119)}$$

Here, $bias_b(|C_{(j)}|, |C_{(k)}|)$ is a term which is introduced to bias against the selection of unevenly sized clusters (size being judged by the number of profiles combined in a cluster rather than the number of categorical predictor values). Call b the *bias coefficient.* Note that when either $|C_{(j)}| = |C_{(k)}|$ (i.e. the clusters each combine equal numbers of the original profiles) or $b = 0$, the bias term equals unity, and (EQ 118) reduces to regular Euclidean distance. However when $b > 0$, the more $|C_{(j)}| / |C_{(k)}|$ differs from 1 (i.e. the more uneven the numbers of profiles in the two clusters), the less the bias term (EQ 119) becomes. Hence the distance between $C_{(j)}$ and $C_{(k)}$ is adjudged to be smaller, and so in turn $\{C_j, C_k\}$ is adjudged to be a worse binary clustering. Note the bias term is symmetric in j and k, and unaffected if $|C_{(j)}|$ and $|C_{(k)}|$ are each multiplied by the same scalar.

It is important that the same bias coefficient b is used globally during the decision tree building, regardless of which sub-algorithm is being used locally. This is because different sub-algorithms may be used for clustering on different variables at a particular node in the tree, so the goodness of clustering measure d_b should be the same in order

to judge equitably which variable should be selected for splitting.

13.4.3 Building the Adaptive Decision Tree Clustering

Prior to any analysis, it is assumed that profiles have first been normalised so that each customer has the same total energy consumption over the study period, as per section 10.2. The clustering tree may either be built using these normalised profiles or, in the full methodology, be built using deweathered (or 'weather-free') profiles constructed by subtracting the correctly rescaled profiles from a MARS regression weather model from the original data (as per section 11.2.5).

The decision tree clustering procedure is a divide and conquer algorithm which can be summarised as follows:

1. Call the clustering we are building $\tilde{C}$. Initially $\tilde{C}$ has one element C_U, the universal cluster (which contains all the profiles).

2. The cluster c in $\tilde{C}$ that has the greatest within-cluster scatter (EQ 73) is selected for splitting (initially this must be the universal cluster).

3. For each predictor $x \in \underset{\sim}{X}^F$ that allows further splitting in c, a locally determined clustering algorithm is selected (see note below) and used to generate a binary partition of c into subclusters c' and c'' using x as splitting variable.

4. Of the binary partitions calculated in 3, that with the greatest modified Euclidean distance $d_b(c', c'')$ (EQ 118) between the two partition centroids is the winner. Cluster c is removed from $\tilde{C}$ and replaced by the two clusters c' and c'' corresponding to the winning partition.

5. Repeat from 2, until a predetermined total number of clusters is reached.

Experiments were performed to determine a suitable policy for choosing which local clustering algorithm to employ in stage 3. For the data sets under study, it was not deemed necessary to use the Binary Splitting Algorithm at all for the extrinsic decision tree clusterings presented in the results section 13.5, since the slower but more thorough Join Two Algorithm was fast enough to be practical. The predictor with the greatest number of categories (X_5, SIC code) has 41 categories (which includes a 'missing' category which is assigned to any customer for which SIC was not recorded in the data, and

an ‘invalid’ category for customers whose recorded codes were not valid), yet even near the root of the decision tree the Join Two Algorithm can be applied with SIC code as the splitting variable in under a minute. However the Binary Splitting Algorithm is retained as part of the methodology as an alternative to the Join Two Algorithm, so that fast exploratory clusterings are possible, and so that in the event that larger data sets are studied by the project Sponsors (with more profiles and/or more SIC codes) a faster algorithm is available. The criteria decided upon for using exhaustive search rather than the faster algorithms is that the predictor variable being used for splitting took seven or fewer values amongst the customers whose profiles were being clustered (entailing an exhaustive search of $2^{7-1} - 1 = 63$ binary clusterings). The threshold value n_e at which the Join-Two algorithm switches to exhaustive search (see section 13.4.2) is also seven.

Note that only month m and SIC code X_5 (and potentially, load factor category X_L) have more than 7 possible values (SIC has 41, month has 12, tariff and day of the week each have 7, and load factor category X_L was also chosen to have seven categories for the studies presented, though is allowed to have more), and hence cause the Join-Two algorithm to be invoked rather than exhaustive search. As the global clustering procedure is carried out and the decision tree built, the number of profiles at the start of a local clustering sub-algorithm tends to get less, since attributes have already been used for splitting (for example, if the very first split chosen at stage 4 is on day of the week, and results in dividing the database/decision tree into Monday-to-Friday profiles and Saturday/Sunday profiles, then all further candidate sub-clusterings using day of the week will have at most 5 initial profiles to cluster). Thus the local clustering sub-algorithms become much quicker as the tree grows, and Join-Two clustering becomes less used (as the number of SIC codes/months represented at leaf nodes in the tree falls to 7 or below).

13.5 Results of Extrinsic Decision Tree Clustering

13.5.1 Data Used in the Extrinsic Decision Tree Clustering

Three versions of the 1995/6/7 load profiles database were used to build the models of this section. Firstly, the original whole loads database (cleaned and with special days removed, as per 11.3.1) was used, *dataset 1*; secondly, a loads database deweathered using the final weather model α (see sections 11.2.5, 11.3.12) was used, *dataset 2*; and thirdly a loads database deweathered using the final model β was used, *dataset 3*.

A problem arises with the deweathered data of datasets 2 and 3. Because of the way

they are generated (by subtracting profiles drawn from a weather model from the original whole profiles), there are occasional negative data in the deweathered datasets. Whilst it would be possible to leave the negative data in the datasets and still build viable clustering models, negative entries cause two problems: (i) interpretation; how should negative entries be plotted, or indeed interpreted, when a negative load is meaningless?; (ii) computation; as a precaution against error, the clustering code checks (in many places) that all profile entries are non-negative, and removal of such fail-safe mechanisms in very large complicated programs is dangerous.

It was decided to remove negative entries, by the simple mechanism of replacing each negative datum with 0. As shown in (TABLE 14.), where negative entries do occur (which is in less than 1% of load values in either dataset), they are on average very small (-4.74 and -7.51 in datasets 2 and 3 respectively - recall were normalised so that the average half-hourly load reading of any customer is 100.00). Whilst there were exceptional deweathered readings that were much more negative than this, these were truly rare; and since exceptionally low load readings are in any case of much less interest to utilities than exceptionally high load readings (since higher than expected demands can be very expensive to supply), removing such readings is not thought to be significant.

Following the removal of negative entries, the deweathered databases were normalised again on a per-customer basis (just as in 10.2) so that each customers mean half-hourly load reading is 100.00 (to which they were all already very close), in order to ac-

TABLE 14.

dataset	weather model	number of negative entries (before removal)	smallest entry (before removal)	mean negative entry (before removal)
2	α	91027 (0.76%)	-26.90	-4.74
3	β	108990 (0.91%)	-50.36	-7.51

count for the removal of negative entries, and the fact that the mean value of MARS model was not always exactly 0.00 for all customers.

Upon examination of the datasets using examples of the visualisation described in 12.5.1, four of which are illustrated in (COLOUR FIGURE 10.) to (COLOUR FIGURE 13.)[1], it was apparent that there were certain whole months of dubious data in the 1995/6 loads database, namely April, July and August of 1995, as well as one month of dubious data in the 1996/7 loads database, namely January 1997. The data for April 1995 is

clearly misrecorded. Recall that the 1994/5 raw data and the 1995/6 raw data overlapped and, for many customers, disagreed; profiles for a given customer in the latter database were scaled by the addition of, or the multiplication by, a fixed scalar (different fixed scalars for different customers). Evidently, for the month of April 95, a normalisation scheme was applied to the raw data that is inconsistent (for some customers) with whatever normalisation scheme was used for the rest of the 1995/6 and 1996/7 raw data. Whether or not the loads are deweathered before plotting, April 1995 loads are on average far higher than those for any other month, so much so that the data could not possibly be correct.

The Sponsor was unable to give any definitive answers on how reliable various periods of the raw data files might be, or how the figures may have been normalised. However, by looking at which profiles are missing from the 1995/6 data - see (COLOUR FIGURE 9.), it becomes apparent that the raw data files have the same customers missing throughout a given calendar month; it must be assumed that different months' data raw were in some cases drawn from different sources, and could be normalised in different ways. Therefore all the April 1995 loads were removed from the data presented to all of the clustering models discussed.

Whilst the April 95 data is without question mismeasured, the unusually high (whole) loads for July and August 1995 (COLOUR FIGURE 10.) do not look impossibly high; it is feasible that they could be due to cooling loads (and the summer of 1995 was an unusually warm summer in the UK). However, these loads still appear to be unusually high even in the deweathered data plotted in (COLOUR FIGURE 12.). There is no obvious reason why the weather models (α, β) described in 11.3.12 would not be able to account for increased loads resulting from high summer temperatures. Particularly, by comparing the visualisations of datasets 1 and 2, it is apparent that loads for the cold winter months have been very effectively normalised to levels comparable to other months by the deweathering process; yet only July and August 1995 (aside from the misrecorded April 1995 loads) appear immune to the deweathering process. Furthermore, plots of certain ANOVA terms of various MARS weather models reveal a definite positive relationship between load and very high summer temperatures, for example (COLOUR FIGURE 5.).

1. Any differences in the visualisations of datasets 2 and 3 (deweathered according to models α, β respectively) are barely discernable, so the visualisation for dataset 3 is not provided.

If the weather models α, β genuinely failed to account for higher cooling loads in July and August of 1995, it is perhaps due to the amount of missing data in August 1995, and particularly, July 1995; recall that each datum presented to MARS was weighted according to the number of profiles which were summed therein, and so the July 1995 data would have had very little weight in the weather models. However it seems more likely that the data for those months is in fact misrecorded. In either case, it was decided to also omit the data for July and August 1995 from all the clustering models that are presented here.

Finally, there is some very strange profile behaviour apparent in the January 1997 loads, which can be seen more clearly in (COLOUR FIGURE 14.). The Thursday and Friday mean profiles are markedly different from the Monday to Wednesday Profiles, and rather similar to the Saturday and Sunday profiles. The Saturday and Sunday profiles also look suspiciously high. It does not seem at all likely that the January 1997 data is correct (at least, not for all customers), though how it might have come to be misrecorded in such a way is not known. Accordingly, this whole month's data was also excluded from all clustering models presented here.

It must be noted that the failure to remove April 1995, January 1997, and possibly July and August 1995, from the data prior to weather modelling must have introduced some slight bias into the models; in particular, the January 1997 data is almost certainly responsible for the curious grouping together of Thursday and Friday in some basis functions involving day-of-the-week. The dubious nature of the April, July and August 1995 and January 1997 data did not become apparent until all of the weather experiments had been performed, and time did not allow for their repetition. It is in any case not unusual that the data analysed by data mining techniques is not wholly reliable, that being the nature of real world databases, and the weather models presented cover too short a time period to be definitive, even if all the data were reliable.

Finally, note that as well as making the profiles much more uniform in level over the course of the study period, the deweathering process has also lessened the starkness of the impact on load shape of daylight saving clock changes. In (COLOUR FIGURE 10.) and (COLOUR FIGURE 11.) there are obvious changes in daily load shape following clock changes near the end of March and end of October (though in fact a clock change at the end of March 1996 is obscured because it falls between the two diagrams; it can be seen in (COLOUR FIGURE 15.)). These clock changes are still visible in diagrams of deweathered load (COLOUR FIGURE 12.) and (COLOUR FIGURE 13.), but con-

siderably less so. Note that all loads, and all weather variables, used in the MARS models were recorded in hours Greenwich Mean Time, and no explicit variable carrying GMT/BST information was used. Thus the weather models have partially accounted for daylight saving through the use of the time-of-year variables available to them.

13.5.2 Use of Percentile Load Factors

Load factor (100% $\times$ mean load/maximum load over some specified period) is often used in the electricity supply industry as a measure of the extent to which a customers' loads maintain their peak level throughout the year (or some other period). A customer with a high load factor has a peak load close to its average load; a customer with a low load factor over a certain period has at least one instance of a much higher than average load measurement. Customers with high load factors generally have profiles that are quite flat; those with low load factors tend to have profiles that are more bumpy or 'peaky'. There are three important drawbacks to the use of load factor as a predictor of load shape:

- First, a customers' load factor, measured over a certain time period $[T_0, T_1]$, may be rather different from its load factor as measured over another time period $[T_2, T_3]$. Therefore a customers' load factor as stored in a customer database may not match its load factor as measured in any particular set of its profiles.
- Second, where the maximum load in a set of profiles used to calculate load factor is mismeasured, the load factor will be strongly affected. As noted in 11.3.1, there are instances in the data of solecisms - abnormally high or low readings that are suspected to be inaccurately recorded, which where undetected might lead to inaccurate calculation of load factor.
- Third, load factor is by definition heavily influenced by an outlying data point, namely the maximum load. Where this maximum load is a particularly atypical load for a given customer, a single outlying datum can significantly affect any model that uses load factor as a predictor. Thus models relying on load factor may not be particularly robust.

Load factors as recorded in the customer database were compared to load factors as calculated on the original whole loads database (over all of 1995/6/7), and there were many severe discrepancies between recorded and calculated figures, indicating that the first and/or the second points above come into play for some customers. Therefore it was

decided to ignore the load factors supplied with the customer database and instead rely on recalculated 1995/6/7 load factors in all studies. However, to minimise the problems associated with the second and third points above, it was decided that recalculated *percentile* load factors would be used for the weather and the clustering models presented in this thesis. A p-percentile load factor is defined to be equal to

$$\frac{\bar{L}}{L^{p\%}} \times 100\% \qquad \text{(EQ 120)}$$

measured over some given period, where $\bar{L}$ is the mean load for the period and $L^{p\%}$ is the upper p-percentile load for the period (the load reading which as close as possible to p% of loads are greater than or equal to). Initially, 1-percentile load factors were used, and later other values for p were tried. Clearly this measure is more robust to outlying data (including mismeasured outlying data) than a regular load factor calculation.

One reason that load factor is widely used in load profiling problems is that it is more easily measured and recorded by a conventional analogue meter than other measures of profile flatness/uniformity - just one value need be recorded, i.e. the peak power drawn over the period between meter readings. However, with half-hourly digital metering becoming much more prevalent, alternative measures of profile flatness, which may be of greater value in load profiling, are becoming viable.

13.5.3 Effects of Bias Coefficient on Decision Tree Clusterings

To examine the effects of the bias coefficient b, a clustering was performed using the variables d (day-of-the-week), X_5 (SIC code), X_6 (tariff code) and X_L (1-percentile load factor category with seven categories), using dataset 2 (model α-deweathered loads), repeated several times with varying coefficient b. In each case, the predetermined number of splits to be performed was set at 11 (resulting in 12 leaf profiles).

The results with $b = 0$ (i.e. no bias against uneven binary clusters) are shown in (). Some notes on the interpretation of the decision trees presented here are necessary. Each node is marked with a numeral to its side; these numerals describe the order in which the nodes were added to the tree. Thus nodes 5 and 6 are always the nodes added by the third split, for instance. Directly below each node (except node 0, the root) is printed the number of profiles represented at that node; directly beneath that, in parentheses, is printed the within-cluster scatter found at that node, expressed as a percentage of the total

scatter in the data (i.e. as a percentage of the within-cluster scatter at the root node). Above the root node, a description of all the variable values of the model predictors that are present in the data is given (though when SIC code is used, this description may be too large to fit on the page). These variable values are expressed as three letter abbreviations (for month or for day-of-the-week values); or as a two figure number (for SIC code category); or as an upper-case L followed by a number from 1 to 7 (for load factor category); or as an upper case A, G or L followed by a two-figure number (for tariff code). Finally, each branch is marked with the predictor values which were clustered together into the corresponding child node at the bottom of the branch.

SIC code categories are numbered sequentially from 1 to 41 in such plots, and do not correspond to the actual two-figure SIC codes which they represent.

After each split in the decision tree building process, the total within cluster scatter (the sum of the scatters for the current leaf profiles) is calculated. The amount of total scatter accounted for by the clustering model after the n th split, S_n, is given by subtracting the n cluster scatters of the leaf clusters in the current model from the pooled (total) scatter S

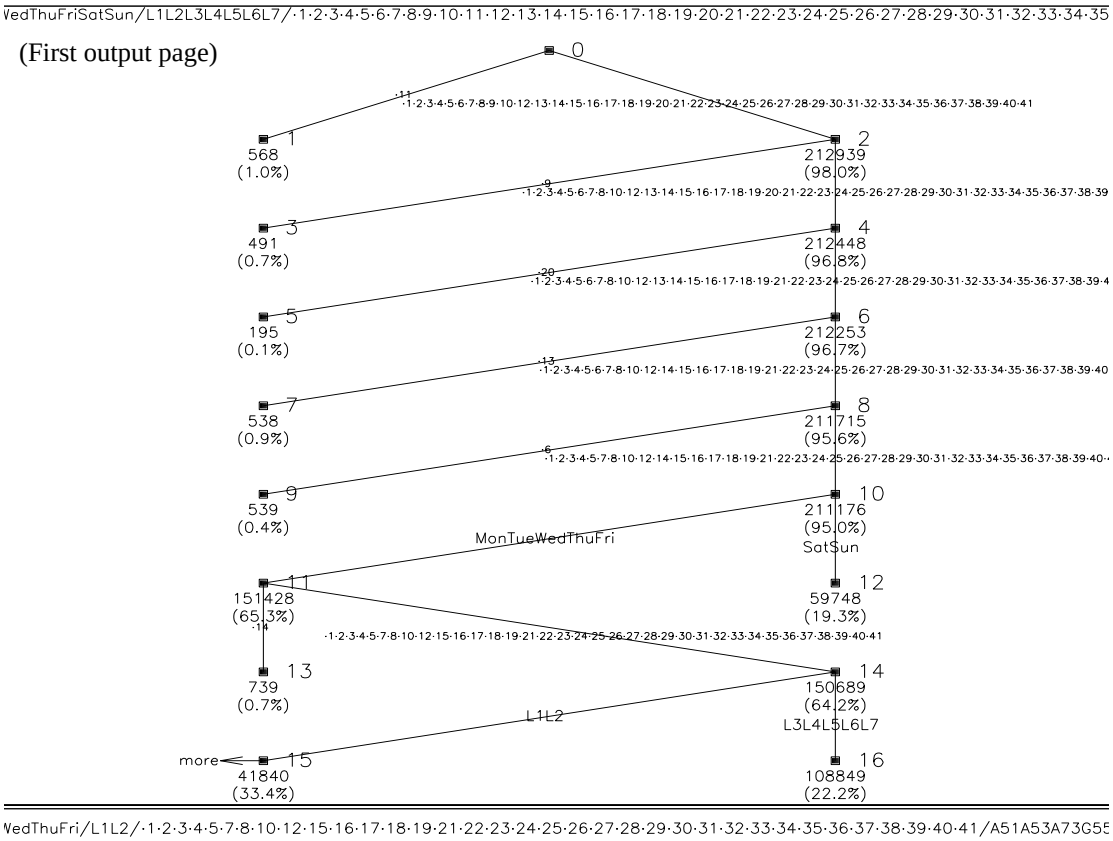
$$S_n = S - \sum_{p=1}^{n} S^{(p)}. \qquad \text{(EQ 121)}$$

(where $S^{(p)}$ is given in (EQ 72)) and the *root-percentage of scatter* accounted for after the n th split, RPS_n, is given by

$$RPS_n = 100\% \times \sqrt{S_n / S}, \qquad \text{(EQ 122)}$$

with the square root accounting for the fact that scatter is a squared quantity.

FIGURE 10.



Bias coefficient $b = 0.0$ (no bias). The decision tree is very deep, and so takes a second output page to display the subtree below node 15. Without bias, many very small clusters are chosen.

Note that scatter accounted for is a measure of how much variation in the *original*

data is accounted for by the clusters, but it is not the only viable measure of model goodness, and in fact, is not what the model attempts to maximise at each split. The model tries to maximise (biased) distance between cluster centres over the range of allowed binary splits; even when the distance measure is unbiased Euclidean distance, this is equivalent to attempting to minimise the scatter of the *atomic profiles* within each cluster, which is not the same as trying to minimise the scatter of the *original profiles* within each cluster, the type of scatter accounted for that S_n measures. Either type of scatter is valid as a measure of model goodness, although time did not allow for the inclusion of an option to use original-profile scatter accounted for as a distance metric.

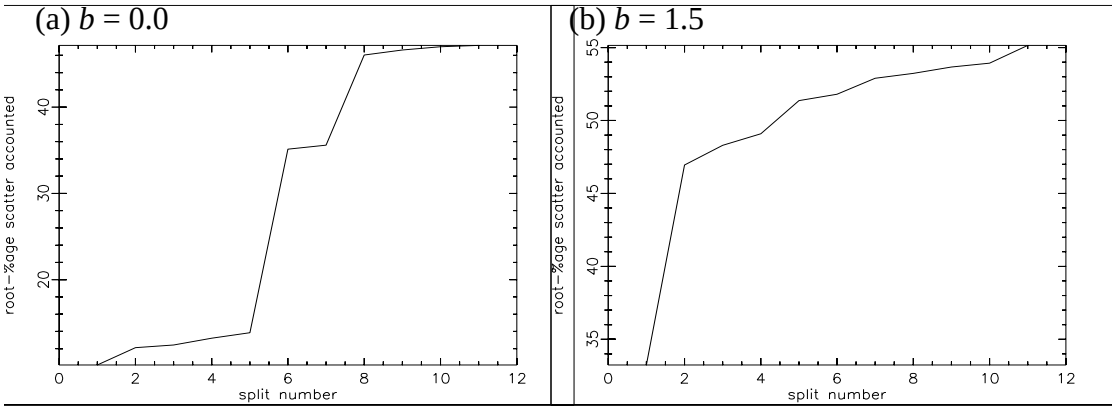
In addition to calculating the total scatter accounted for after each split, after every split in the decision tree building process the amount of total scatter accounted for by that particular binary split, $S_n - S_{n-1}$, is added to a *scatter-by-variable sum* S_v, where v is the variable whose values were used to determine the split. Before any splitting has taken place, the scatter-by-variable sums S_d, S_m, S_{X_5}, S_{X_6}, S_{X_L} are all zero. If at the first split, day-of-the-week is the splitting variable selected, and the first split accounts for scatter $s_1 = S_1 - S_0$, then s_1 is added to S_d, and so on. Following the final split, the scatter-by-variable sums serve as a guide to the relative variable importance of the various model predictor variables. Relative variable importance for variable v, imp_v, is given by

$$imp_v = 100\% \times \sqrt{S_v / S} \qquad \text{(EQ 123)}$$

Again, the square root takes account of the fact that scatters are measured in squared units. This is only one viable measure of variable importance, and (like GCV in MARS modelling) cannot be taken as the definitive guide to relative variable importance for any given model.

The root-percentage of scatter accounted for after the nth split (for the zero-bias model of (FIGURE 10.)) is graphed against n in (FIGURE 11.(a)).

FIGURE 11.



RPS_n against n for models with bias coefficient 0.0, 1.5. Note that the scales for the y axes differ.

The same clustering was repeated with biases coefficients of $b = 0.5, 1.0, 1.5$ and

TABLE 15.

bias coefficient b	relative variable importances:				final root-% scatter accounted for
	day-of-the-week	load factor	SIC	tariff	
0.0	32.2753	29.2146	18.1208		47.1546
0.5	33.2348	33.4844	22.1930		52.1375
1.0	33.2348	37.0760	16.7887	12.4196	53.9934
1.5	33.2348	39.777	12.5758	14.0232	55.1503
2.0	33.2348	39.5162	15.7292	8.82469	54.6934

2.0. The final root-percentage scatter accounted for (after 11 splits in each case) as well as the relative variable importances are given in (TABLE 15.) - an empty cell indicates that a variable was never selected for splitting in a given experiment. Looking at the final root-percentage scatters accounted for by these 5 models, it is immediately clear that at least some bias against small clusters is beneficial in terms of model goodness of fit; (FIGURE 11.) in particular shows how much more quickly scatter is accounted for with a reasonable degree of bias against small clusters - (FIGURE 11.(b)) shows scatters accounted for with $b = 1.5$.

The decision tree for the $b = 1.5$ model is shown in (FIGURE 12.). Considerably more scatter is accounted for in this model than when there is no bias (FIGURE 10.). Without bias the model tends, for most of the eleven splits determined, to pick off just one (or occasionally two) SIC codes into one very small cluster, which does not get split again (or at least, not by the eleven splits calculated). In all of these splits, only one or

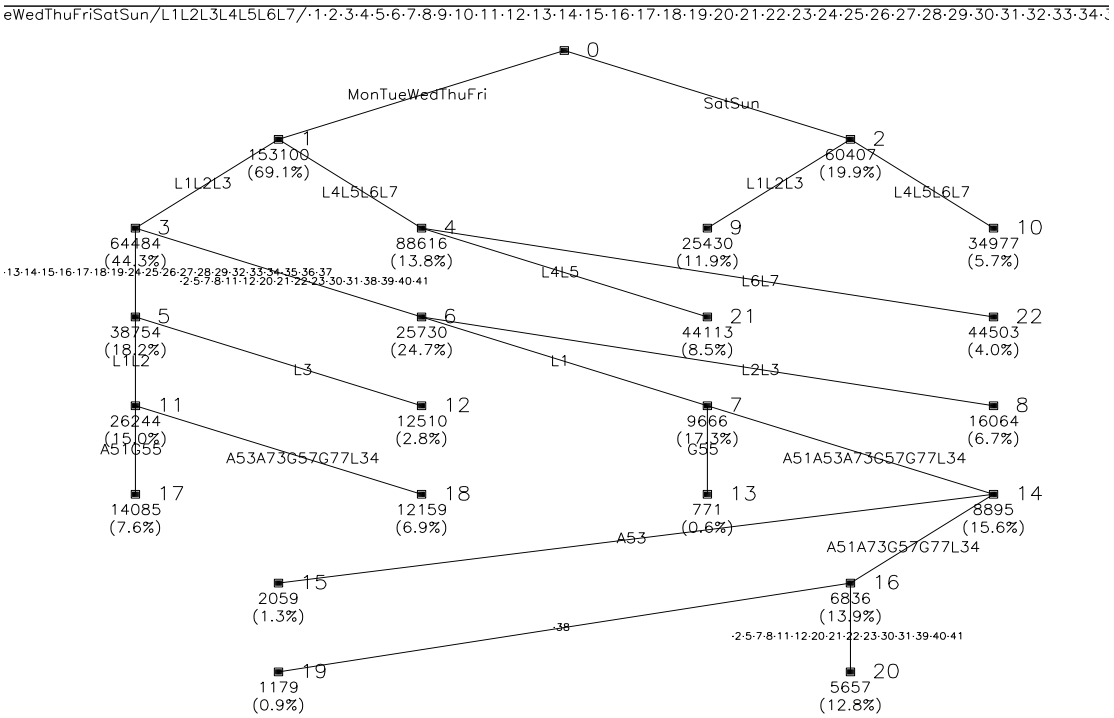
two customers belong to the SIC code categories that make up the small cluster; whilst that customer's centroidal profile over the relevant dates has a very great Euclidean distance from the centroidal profile of the remainder of the customers begin considered, because just one or two customers end up in the small cluster the amount of scatter accounted for is generally small.

When bias is introduced against the small clusters, not only is the total amount of scatter accounted for increased, but the informativeness of the model (informally speaking) is much better. (FIGURE 12.) tells us more about the nature of load shape variation in relation to load factor category and tariff code than does (FIGURE 10.); but arguably it tells us more useful information about the effect of SIC code on load shape than does (FIGURE 10.), even though SIC code is used much more as a splitting variable when bias is absent. Split number 3 (into nodes 5 and 6), tells us that the customers belonging to the 16 SIC codes represented at node 6, with load factors in categories L1, L2 or L3, have relatively similar load shapes (at least on weekdays); all the third (say) split in the bias-free model tells us is that one particular SIC code category has load shape relatively unlike that of other categories - and since only one customer in the data has that SIC code, this might not generally be the case in the population at large anyway.

Looking at (TABLE 15.) we see that day-of-the-week is rated as the most important variable when $b = 0.0$, ahead of load factor category and SIC code. Only in this model was day-of-the-week not selected at the first split; the $b > 0$ models all split day-of-the-week into weekday and weekend first of all. Load factor is rated slightly more highly than day-of-the-week for $b = 0.5$, and more so for the higher biases. Tariff code is not selected at all when $b = 0.0$ or 0.5; however with higher biases, SIC code-based clusterings with very uneven sized clusters are more heavily penalised, and tariff code gains in relative importance (though tariff code falls slightly in importance when bias increases to 2.0).

A bias coefficient figure of $b = 1.5$ was selected for all the clustering models that follow, not only because this coefficient accounts for a relatively high amount of within cluster scatter after 11 splits in the models of this section, but because a bias coefficient of 1.5 has been found to produce trees which carry (informally) interesting information about load shape and about the available predictor variables in several experimental variations on these models (including when month is also included in the set of predictor variables).

FIGURE 12.



Bias coefficient b = 1.5. The tree is shallower (fitting on one output page), and SIC code is selected for splitting much less frequently.

13.5.4 Comparison of Clustering Models on Datasets 1, 2 and 3

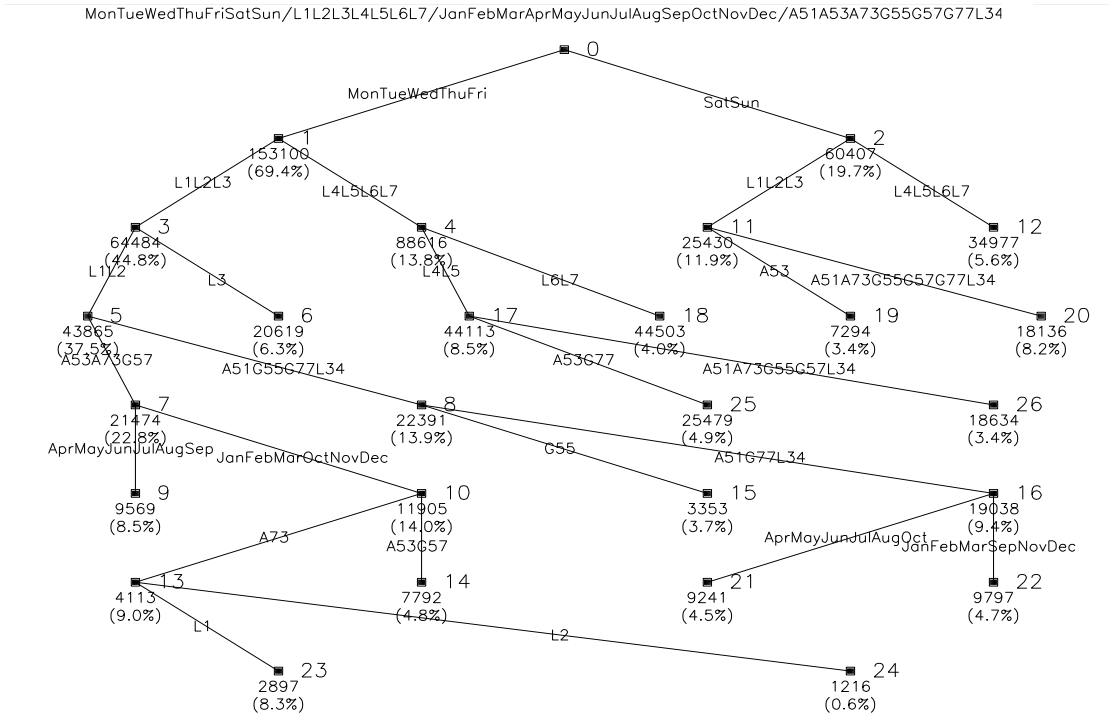
With the bias coefficient b fixed at 1.5, models were built on all 3 datasets described in 13.5.1. The variables used were day-of-the-week, load factor category, month and tariff code, and the number of splits was slightly increased from the models in 13.5.3, to 13 (generating 14 leaf profiles). Results from the three models appear in (TABLE 16.),

TABLE 16.

dataset	relative variable importances: day-of-the-week	load factor	month	tariff	final root-% scatter accounted for
1 (whole loads)	33.0211	39.4169	7.25369	14.6994	53.9701
2 (deweathered using model α)	33.2347	40.0555	3.75736	14.3839	54.1295
3 (deweathered using model β)	33.222	39.9864	2.8251	14.5316	54.0533

whilst their respective decision trees appear in (FIGURE 13.) to (FIGURE 15.).

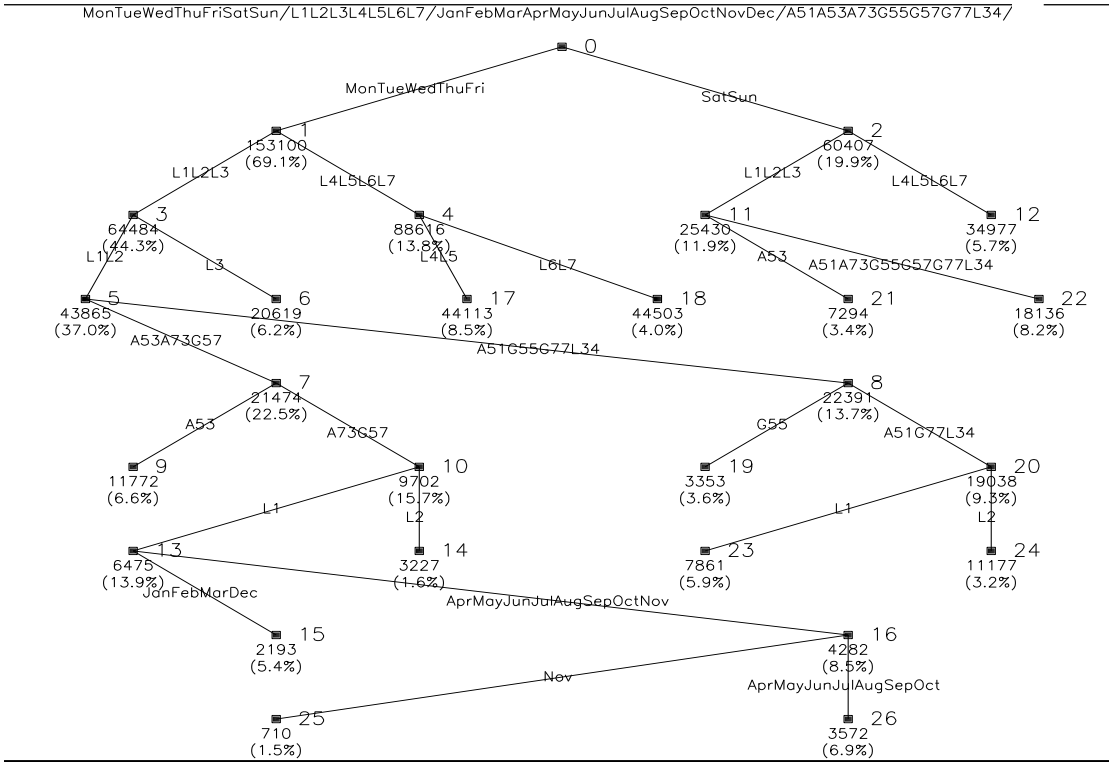
FIGURE 13.



Dataset 1 (whole loads), b = 1.5, clustering with day-of-the-week, month, load factor, SIC code.

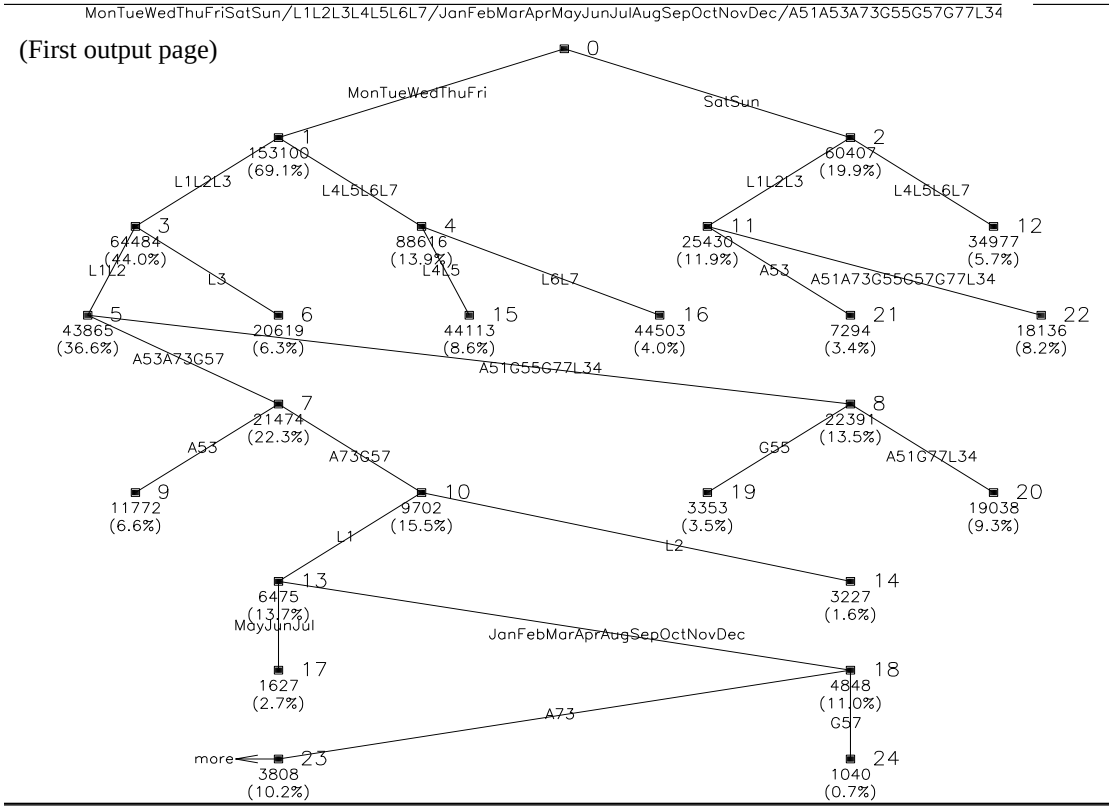
From (TABLE 16.) it is immediately clear that the principal effect of deweathering the loads database is that month becomes far less important as a splitting variable in the clustering methodology. This was expected, since weather's effect on load shape varies much more from month to month than it does from tariff code to tariff code, from load factor category to load factor category, and so on, because the weather itself is different from month to month. In the model built on dataset 1 (whole loads), month is first selected as the splitting variable for the fifth split, and for a second and final time for the eleventh split. In the model built on dataset 2 (deweathered without using categorical variables in the weather model), month is not selected until the eight split, and then once more for the final (13th) split; in the model built on dataset 3 (deweathered using day-of-week and load factor in the weather model), month is not selected until the ninth split, and is selected also on the final (13th) split.

FIGURE 14.

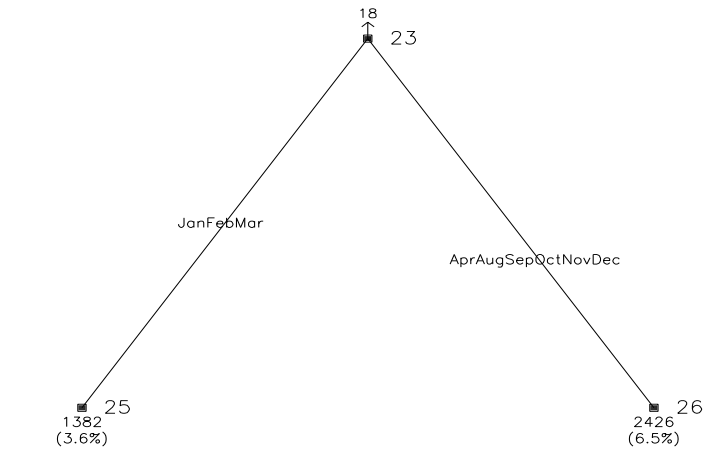


Dataset 2 (loads deweathered with weather model α), b = 1.5; variables as in (FIGURE 13.).

FIGURE 15.



(Second output page)



Dataset 3 (loads deweathered with weather model β), b = 1.5; variables as in (FIGURE 13.). A second output page is required to display the subtree below node 23.

It appears that the weather modelling removes most, though not all, of the time-of-year dependence in the deweathered loads databases; and that it can do so even more effectively when certain categorical variables (day-of-the-week and load factor category) are used as predictors in the weather model. In fact, for the models presented here, month

is only selected as a splitting variable in the deweathered datasets in a part of the model which applies only to the lowest load factor category and to two tariff codes; for the whole loads dataset, month is selected in parts of the dataset that apply to the lowest two load factor categories (but various tariff codes). All three models account for a very similar root percentage of scatter RPS_n after 13 splits, though the model built for dataset 3 accounts for scatter slightly the fastest - after 6 splits only, the respective RPS_6 scores for datasets 1, 2 and 3 are 51.2045%, 51.4895% and 51.6171%. Whilst there is a large change between whole and deweathered data in *where* in the clustering tree month is selected, month is used in similar ways in all the models, i.e. to divide colder/darker months from warmer/lighter months. In the whole loads clustering, note that the daylight saving clock changes occur at the end of March and towards the end of October, and so rather close to the month splits that occur in the whole loads model (FIGURE 13.). For dataset 2 the warmer/lighter months (as determined by the clustering) begin with April and end with November, though November is subsequently separated from April to October. For dataset 3 the warmer/lighter months (as determined by the clustering) do not appear to be closely related to daylight saving clock changes. Note that whatever dataset is used, the lower load factor categories tend to be much more intensively modelled (i.e. much more splitting occurs in the parts of the model with lower load factors), because disproportionately more scatter exists in those parts of the model (since customers with high load factors tend to have much flatter profiles, and accordingly less scatter amongst their profiles).

Two further experiments were performed to try and determine the effect of deweathering loads on clustering, when month is not present as a predictor variable in the clustering model. The same parameters (b = 1.5, number of splits = 11) are used as were used in the clustering of (FIGURE 12.), so direct comparison is possible, but the datasets used were 1 and 3 (not dataset 2, which was used in generating (FIGURE 12.)). Results for the three clusterings are displayed in (TABLE 15.), and graphs for the decision trees

TABLE 17.

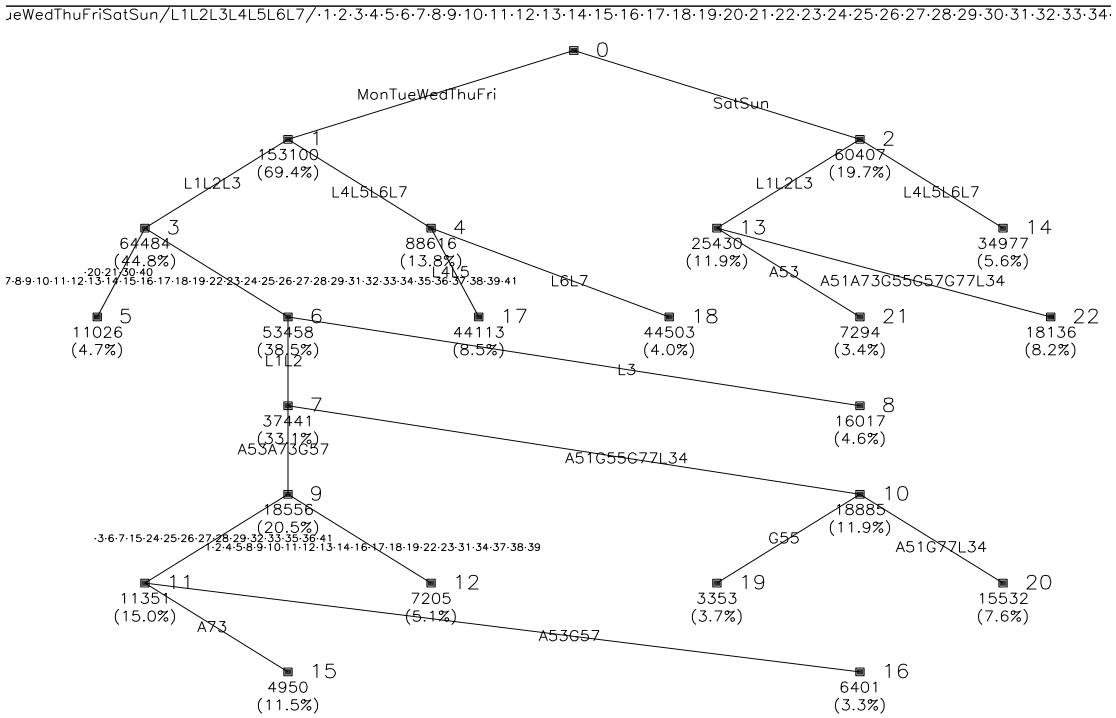
	relative variable importances:				**final root-% scatter accounted for**
dataset	**day-of-the-week**	**load factor**	**SIC**	**tariff**	
1 (whole loads)	33.0212	38.9320	14.0695	13.3177	54.6023

TABLE 17.

dataset	relative variable importances: day-of-the-week	load factor	SIC	tariff	final root-% scatter accounted for
2 (deweathered using model α)	33.2348	39.777	12.5758	14.0232	55.1503
3 (deweathered using model β)	33.222	39.9645	13.1577	12.6813	55.089

in (FIGURE 16.) for dataset 1 and (FIGURE 17.) for dataset 3, as well as (FIGURE 12.) for dataset 2.

FIGURE 16.

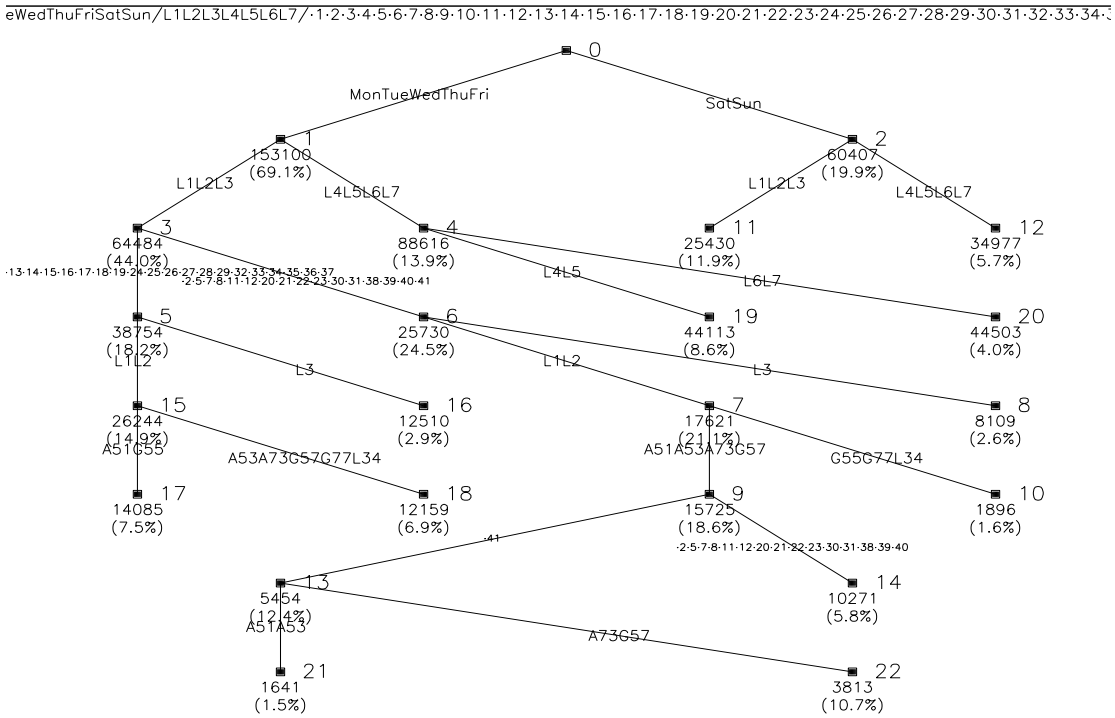


Dataset 1 (whole loads); $b = 1.5$.

The results of (TABLE 15.) suggest that deweathering a dataset before performing a decision tree clustering affects the resulting clustering somewhat even when time-of-year information (i.e. month) is absent from the model. The percentage scatter accounted for is somewhat better for the deweathered datasets; and whilst the relative variable importances remain similar for all three models, there are substantial differences between the clustering decision trees for whole and deweathered loads; this is further evidence that weather has rather different effects on the load shape of customers who differ in their

customer attributes (SIC, tariff, load factor), which was already clear from experiments in section 11.3.11. The trees for datasets 2 and 3 (deweathered with weather models α and β respectively) also differ, though rather more subtly.

FIGURE 17.



Dataset 1 (loads deweathered with weather model β); $b = 1.5$.

13.5.5 Comparison of Clusterings Using Different Percentile Load-Factor Variables

As discussed in 13.5.2, rather than calculating load factors as a ratio of average load to maximum load, they may be calculated as a ratio of average load to the top p-percentile load. The clustering models so far (and, in fact, the weather models where they have used load factor) have used 1-percentile load factors. This was motivated more by the danger of misrecorded peak loads biasing the calculated load factor[1] than by the more general problems of using (conventional) load factor as a predictor. The more general problems are that a few (correctly recorded) peak loads can heavily affect a final model when load factor is a predictor, and that a customer's load factor can change considera-

1. Indeed, the 1-percentile load factors vary little from the true (or 0%, i.e. conventional) load factors, in general.

bly depending on the time period over which it is recorded.

The atomic profiles for the clustering model were recalculated from dataset 2 using various p-percentile load factors; p values of 0.0% (conventional load factor), 1.0%, 10.0%, 20% and 33% were tried. Using these differing sets of atomic profiles 4 new models were built using day-of-the-week, p-percentile load factor category, tariff code and SIC code as predictors, setting the number of splits to 11 and the bias coefficient b to 1.5 (note that the 1-percentile version has already been built with this dataset and these parameters - see (FIGURE 12.)). Results appear in (TABLE 15.).

TABLE 18.

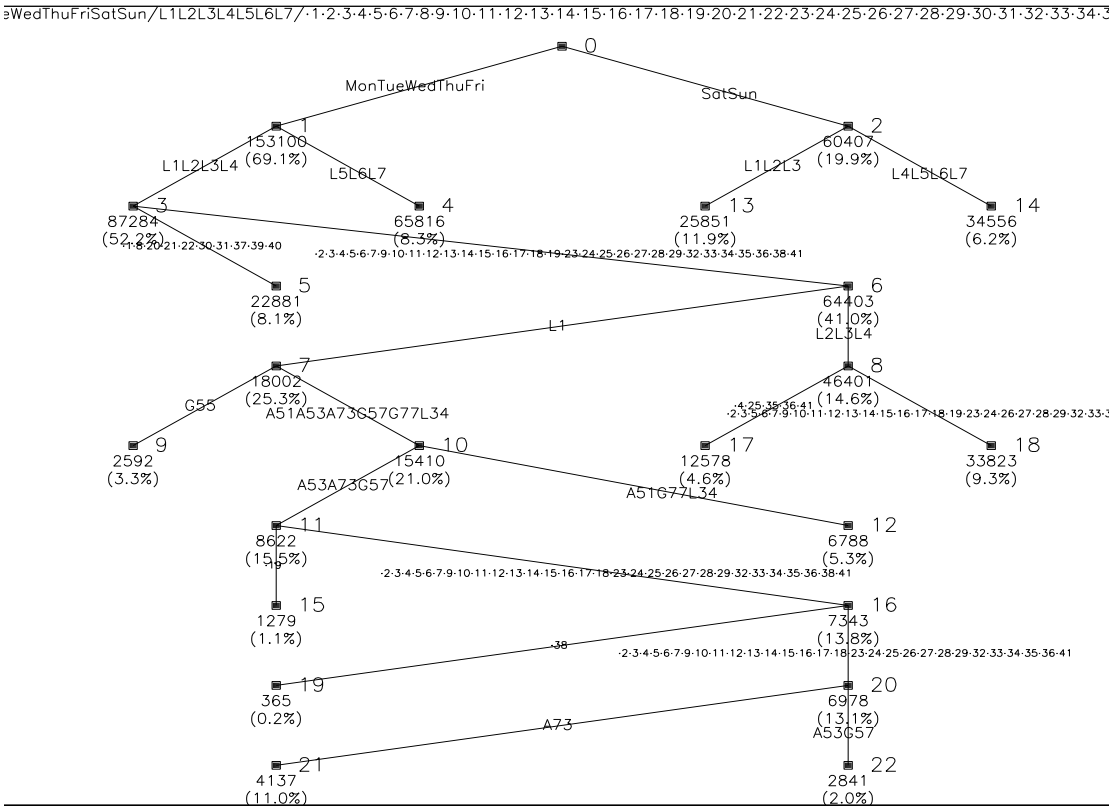
percentile point p	**relative variable importances:**				**final root-% scatter accounted for**
	day-of-the-week	**load factor**	**SIC**	**tariff**	
0.0%	33.2348	33.7846	22.0294	11.8534	53.5886
1.0%	33.2348	39.777	12.5758	14.0232	55.1503
10.0%	33.2347	44.6534	12.9444	10.2584	58.0626
20.0%	41.7316	39.8381	15.4424	6.8826	60.1202
33.0%	33.2348	23.2382	15.0157	26.4896	50.7123

Using conventional load factor ($p = 0.0\%$), less scatter is accounted for (after 11 splits) than in the previously built model (with $p = 1.0\%$); load factor loses importance, at the expense of SIC code. SIC code is selected for splitting four times, load factor just 3 times (FIGURE 18.); whereas when using 1-percentile load factors (FIGURE 12.) load factor was selected 5 times (SIC just twice). Thus ignoring as little as the top 1% of a customers' loads when calculating its maximum load is enough to make load factor a more useful splitting variable.

The gains in scatter accounted for when increasing p to 10% and 20% are even more impressive; load factor attains its greatest importance, as measured by imp_{X_L} (EQ 123), when $p = 10\%$ - the decision tree for that model is given later in (FIGURE 21.)). The greatest amount of total scatter accounted for (after 11 splits) occurs when $p = 20\%$, where the presence of 20-percentile load factor as a predictor allows day-of-the-week to take on more than its usual importance. We can see in (FIGURE 19.) that with $p = 20.0\%$, load factor is actually selected for the first split, ahead of day-of-the-week. The usual weekend/weekday split does occur lower in the tree: immediately afterwards for the lower 3 load factor categories, and on the tenth split for some customers with load factors in the 4th and 5th load factor categories (though not at all, for some customers). This arrangement actually allows day-of-the-week to take on a greater importance (as

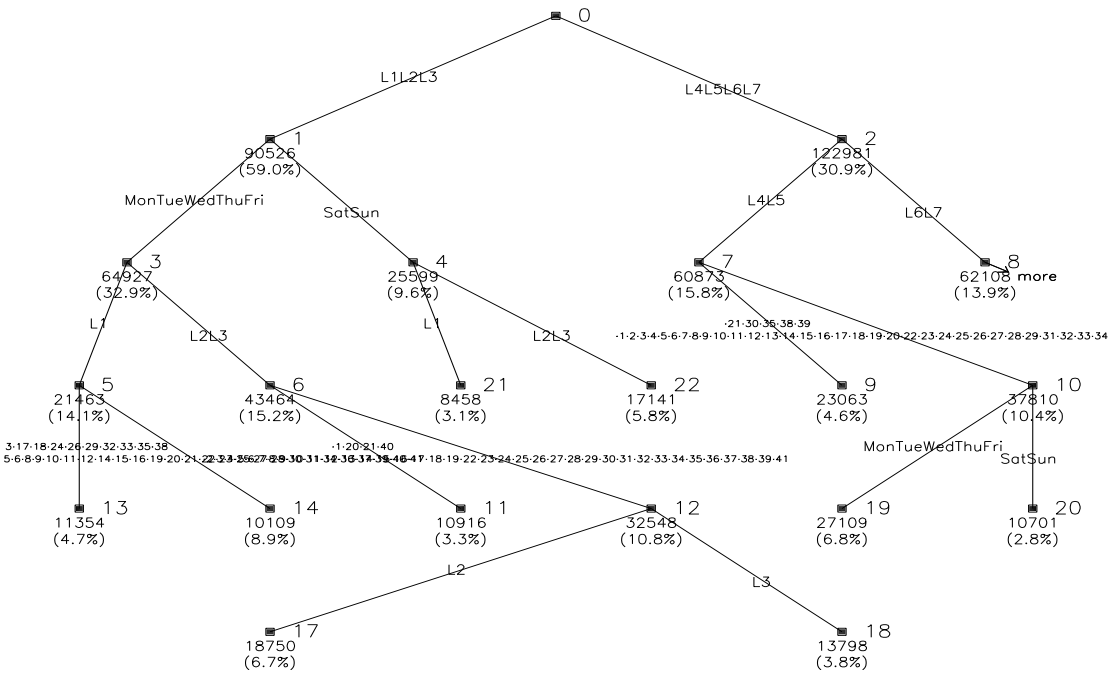
measured by imp_d) than in the other models where it is picked first.

FIGURE 18.



Use of conventional load factors (p =0.0%). SIC is used for splitting 4 times, load factor just 3 times; with p =1.0% (FIGURE 12.), load factor was selected 5 times (SIC just twice).

FIGURE 19.

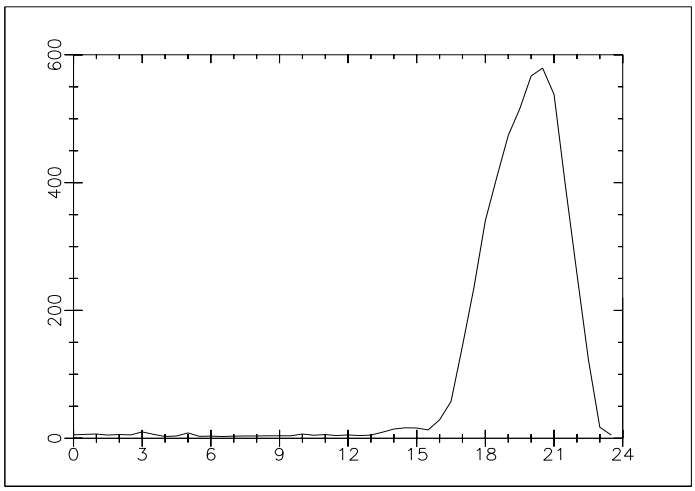


With p =20.0%, load factor is actually picked ahead of day-of-the-week.

When increasing the percentile point p to 33.0%, the gains made in terms of scatter accounted for disappear, and the model accounts for less scatter than when conventional load factor is used.

We can examine some problems associated with setting p too high by looking at the (whole) profiles of a particular customer (call them customer A), who has a very low load factor as calculated conventionally. Customer A's centroidal profile, over the study period, is shown in (FIGURE 20.). The y-axis is scaled between 0% and 600% of average half hourly load - the customers daily peak average load is nearly six times its average load; the x-axis shows time of day. Customer A's SIC code is missing from the customer database, but the customer is listed as 'Tennis Courts' in the Sponsor's full customer database.

FIGURE 20.



Centroidal profile over study period of customer A.

In fact, customer A has the lowest conventional (0-percentile) load factor of all the customers in the 1995/6 database, with a peak half hourly load 12.77 times its mean half hourly load. However, as shown in (TABLE 19.), customer A's 20-percentile and 33-

TABLE 19.

percentage point p :	**0.0%**	**10%**	**20%**	**33%**
p -percentile load factor for customer A	7.83%	22.71%	256.31%	1468.61%

percentile load factors are extremely high; in fact customer A has the highest 20-percentile load factor in the database, and the highest 30-percentile load factor in the database.

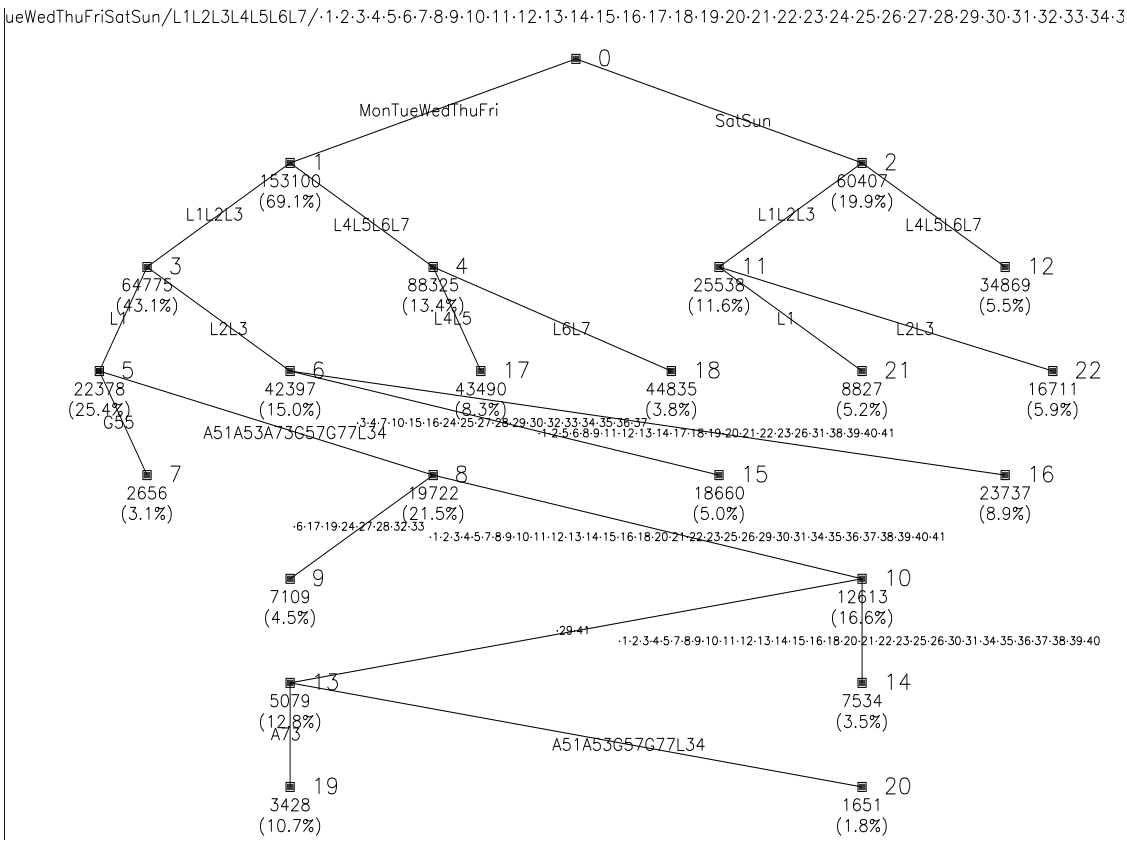
Whilst percentile load factors are intended to be a more ‘forgiving’ measure of profile flatness than conventional load factor, in that the highest p% of loads for a given customer have no influence on percentile load factor, it would seem that using too high a percentage point p can be much too forgiving; in the case of customer A, well over half of its loads are very small in comparison to its mean load. Most of A’s power is used when load levels are greater than the denominator of percentile load factor (EQ 120) when p = 20% or 30%, and so A is rated (by percentile load factor) as having a very flat, uniform profile, whereas the opposite is true.

Thus some caution should be exercised if p-percentile load factor is to replace conventional load-factor as a measure of profile flatness, that p is not set too high.

13.5.6 Marginal, Difference & Effect Profiles in a Decision Tree Clustering

The model of the previous section using 10-percentile load factor, bias coefficient $b = 1.5$, predictor variables d, X_5, X_6 and X_L and 11 splits, is illustrated in (FIGURE 21.).

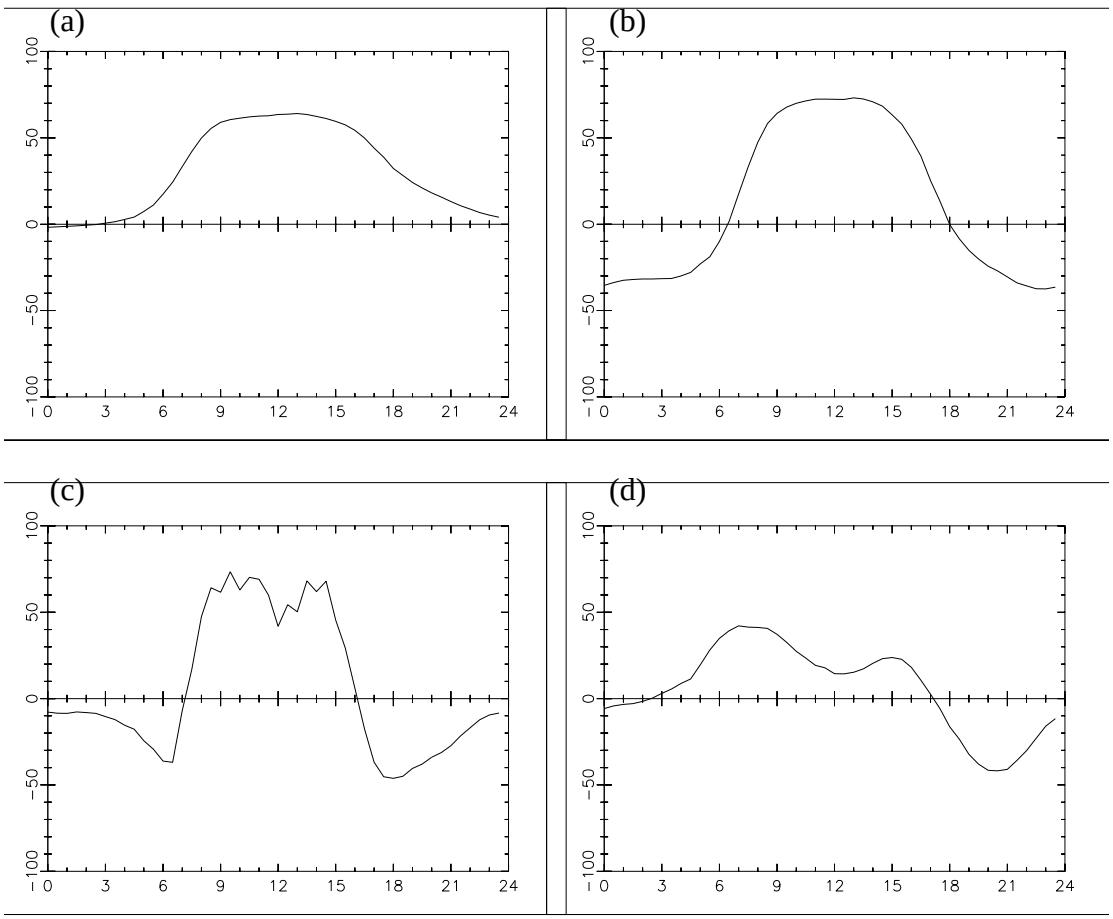
FIGURE 21.



Decision tree with p =10%, b =1.5%; described fully in section 13.5.5.

The first split is into weekday and weekend clusters. The amount of scatter in the weekend cluster (node 2) is much smaller, and consequently much less recursive splitting goes on beneath node 2 than node 1 (the weekdays node). The difference profile (see section 12.2) for nodes 1 and 2 (weekday/weekend) is given in (FIGURE 22.(a)). This shows that weekday profiles are somewhat higher between 05:00 and 22:00 GMT, and much higher between 08:00 and 16:00 GMT, but almost the same from 22:00 to 5:00 GMT. A seasonal overview of the profiles in node 1 (weekday profiles averaged for all customers) is given in (COLOUR FIGURE 16.). Note that due to deweathering there is little seasonal variation among the profiles; white (i.e. paper coloured) areas indicate missing/omitted days and Saturdays and Sundays.

FIGURE 22.



Difference/Effect Profiles for the clustering of (FIGURE 21.)

The next two splits are subdivisions according to (10-percentile) load factor catego-

ry, and it is the three lowest load factor categories (at node 3) which carry the bulk of remaining scatter, and which are recursively split the most times subsequently, particularly load factor category 1 (node 5) which is split another 4 times, according to tariff category (twice) and SIC code (twice). A difference profile for nodes 3 and 4 (weekday low load factor (L1, L2, L3) profiles and weekday high load factor profiles is given in (FIGURE 22.(b)). Between about 06:30 and 18:00 GMT, the lower load factor profiles are typically much higher than the higher load factor customers, and this trend is reversed for the remainder of the day. The difference is most marked between 09:00 and 15:00 GMT. The seasonal plots for nodes 3 and 4 are given in (COLOUR FIGURE 17.) and (COLOUR FIGURE 18.) respectively. Notice that while there is little seasonal variation in node 4, there remains rather more seasonal variation unaccounted for by the weather model in node 3.

As we move further down the tree, the difference profiles between sibling nodes, and effect profiles (differences between daughter and parent profiles) tend to become less smooth, and also more interesting. For example, the effect profile of node 7 (representing one particular tariff code amongst customers in load factor category 1, on weekdays) on node 5 (load factor category 1, all tariffs, on weekdays) is given in (FIGURE 22.(c)). It demonstrates that customers with this tariff code tend to have higher loads towards the middle of the day (07:00 to 16:00) than other customers in the same load factor category, much lower loads during early morning and early evening, but similar loads at night. The seasonal diagram for node 7 is given in (COLOUR FIGURE 19.). The difference profile between node 15 and 16 (differing groups of SIC codes for customers in load factor categories two and three, weekdays) given in (FIGURE 22.(d)) shows how subtle the differences between the clusters can become lower down in the decision tree.

13.6 Subatomic Clustering at Leaves of the Extrinsic Decision Tree Clustering

A clustering algorithm which seeks clusters of profiles of any form, rather than a decision tree approach which always partitions using values of a particular variable, allows for more flexible clusterings. Whilst we might expect such a clustering to be very much slower, without extrinsic variables to guide the search for clusters, we might also expect the final clusters to better satisfy goodness-of-clustering criteria (when the number of clusters is the same in either model) as a result of the freer form of its clusters.

However, this increased flexibility is arguably very much a disadvantage, since the end model is vastly less interpretable than a decision tree. Each leaf in a decision tree has a single path to the root, marked with simple conditions on attributes. Thus the exact meaning of any cluster (whether a leaf cluster or a coarser higher level cluster) is instantly interpretable. Furthermore, effect curves allow for comparison of the effects of predictors between the various clusters at various levels, and the decision tree itself is a highly interpretable at-a-glance visualisation of both global and local data structure.

However, the leaves of a decision tree clustering like those presented in section 13.5 often contain a significant amount of scatter unaccounted for; it seems more than likely that there are patterns of variation at the leaves that are hidden by the atomic structure of the data used in extrinsic decision tree clustering. Customers represented within the same leaf may have very different load shapes, but be indistinguishable because they have the same values for each of the predictor variables under consideration. There may be customers in the same leaf cluster with very different load shapes, but that would require several more splits using extrinsic variables to end up in different leaves - whereas a single split that was ‘free’ rather than dictated by extrinsic variable values might immediately separate them.

Since the number of profiles in any leaf of a decision tree tends to be much smaller than the number of initial profiles, a *free-form* (or *subatomic*, i.e. intrinsic, not guided by extrinsic temporal and customer variables) clustering on the profiles at a given leaf may be viable, provided the clustering algorithm is a very rapid one; however the subatomic clustering of profiles at a leaf can be made very much faster still by imposing that all the daily profiles of any given customer end up in the same cluster; then if m distinct customers are found at a particular leaf, there are just m patterns (the customers’ centroidal profiles for the dates represented at the leaf) to be clustered.

A faster algorithm is required than the join-two algorithm, since m may still be rather large, so the *binary splitting algorithm* of section 8.6 is employed to generate binary clusterings at the leaves of a decision tree. The framework within which this happens is the same as for the extrinsic decision tree clustering we have already seen: the leaf cluster with the greatest within-cluster scatter is selected for sub-atomic binary clustering with the binary splitting algorithm; the two clusters so generated replace the old leaf cluster in the decision tree; and these new leaves are made available as candidates for further subatomic splitting, should either of their within-cluster scatters become the greatest remaining within-leaf scatter. In fact, the same biased distance measure is used

by the binary splitting algorithm when performing sub-atomic clustering. Thus the subatomic clusters can be viewed on the same decision tree as the preceding extrinsic atomic clustering - though the branches are merely marked with the number of customers represented at the node below the branch.

13.7 Subatomic Clustering Results

Subatomic clustering as described in 13.6 was applied at the leaves of an atomic decision tree clustering. The atomic decision tree clustering used 11 splits using the variables d, X_5, X_6 and X_L, bias coefficient b = 1.5, 10-percentile load factors and the deweathered data of dataset 2 (this is the clustering illustrated in 13.5.6). An additional 10 subatomic splits were generated on the leaves of the original atomic decision tree, still using a bias coefficient of 1.5.

A graph of root-percentage scatter accounted for after n splits RPS_n is given in (FIGURE 24.). The dashed line marks the boundary between the 11th (final) atomic split and the first subatomic split. Note that the rate of increase of RPS_n accounted for is falling sharply before the beginning of the subatomic splitting algorithm. However as soon as the subatomic splits begin to be generated, the rate in increase of RPS_n rises sharply, until after the first five subatomic splits the rate of increase of RPS_n slows down again.

TABLE 20.

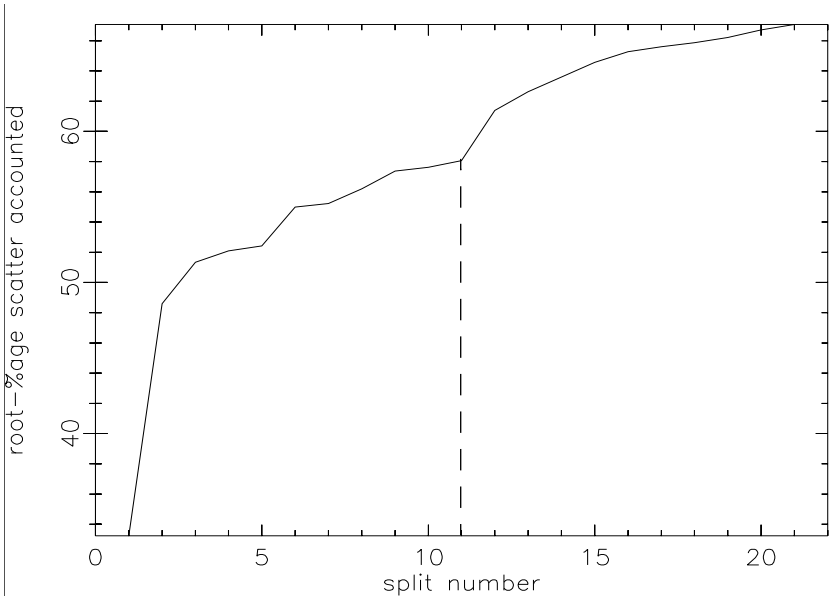
Model	# Atomic splits	# Subatomic splits	relative variable importances:					final root-% scatter accounted
			day-of-the-week	load factor	SIC	tariff	sub-atomic	
atomic	21	0	33.500	45.186	17.889	11.899	n/a	60.2133
mixed	11	10	33.235	44.653	12.944	10.258	33.586	67.0765

(TABLE 15.) shows a comparison of the performance of the subatomic clustering model in comparison to a model with the same parameters, and also using 21 splits, but using only extrinsically guided atomic splits. All scatter accounted for by subatomic splits have been added and converted to a root percentage to give a relative ‘variable’ importance for subatomic splits, though of course no extrinsic variable guides these splits. As would be expected, more scatter is accounted for by the mixed atomic/subatomic model. The subatomic splits are awarded a combined importance similar to day of the week but less than load factor (though of course it is not a very fair comparison, as on the one hand these splits take place after the other splits, when much of the scatter is

already accounted for; and on the other hand, these splits are much freer in the profiles that they are allowed to put into different clusters).

The decision tree for the mixed atomic/subatomic model is given in (FIGURE 24.). Since the tree is very large, the weekday model (descending from node 1) and the weekend model (descending from node 2) are given separately. Note that on some occasions

FIGURE 23.



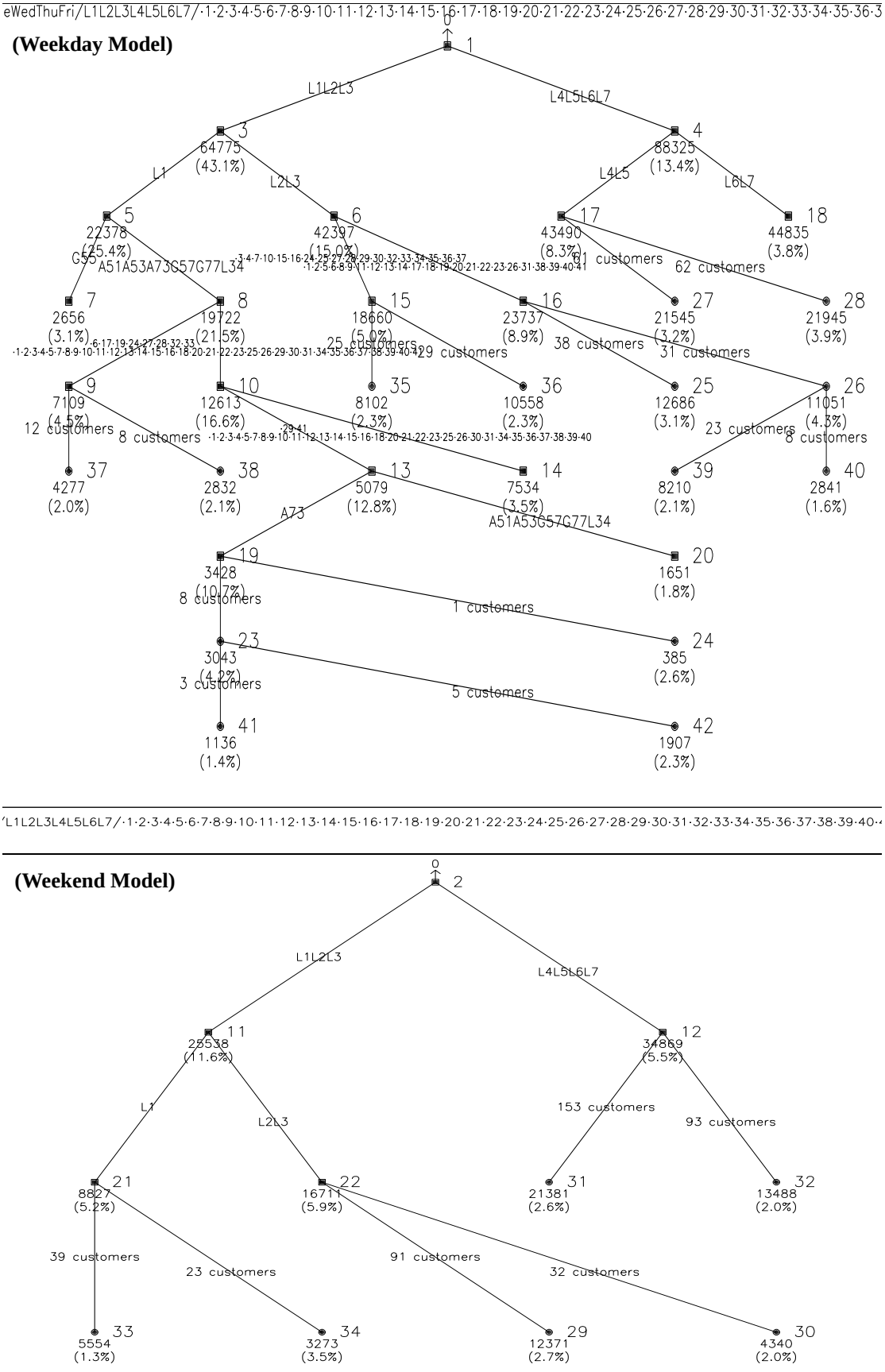
RPS_n against number of splits n for a mixed atomic/subatomic clustering.

nodes that were generated by subatomic splitting are selected again for subatomic splitting.

It is hoped that the subatomic clusters that can be generated using this method may be a useful tool in identifying niche markets for particular tariffs. By identifying small clusters of customers who have similar profiles to each other, but dissimilar to those of other customers with similar attributes (load factor, tariff group, SIC code), it may be possible for a utility to identify a load shape for which it can price electricity competitively, and to attempt to court similar customers from competing utilities.

However, the subatomic part of the model is of little use in predicting a new or potential customer’s load shape given their attributes alone, because there is no extrinsic variable to suggest which half of a binary subatomic clustering the customer should belong to.

FIGURE 24.



Weekday/weekend halves of a mixed atomic/subatomic decision tree clustering.

Chapter 14 — Possible Directions For Further Research

There are a number of suggested possible refinements of and extensions to the methods presented in this thesis for data collection and cleansing; for weather modelling of half-hourly load data using weather, temporal and customer variables; for deweathering of whole load using such a weather model; and for clustering whole or deweathered profiles using a variety of customer and temporal variables (and also without using extrinsic variables). A number of minor possible enhancements to the methodologies have already been suggested in Chapters 11 and 13, and these are, in general, not repeated here. However most of the extensions and alternative approaches suggested in this chapter would be quite substantial research undertakings in their own right.

14.1 Improvements in Data Quality

One obvious way to improve the quality of results would be to procure more and better data; data for more customers over more dates, data which contains fewer missing dates and months, customer data without missing SIC codes, customer survey data concerning end uses (such as presence of storage heating, air conditioning, etc.), and perhaps foremost, data which is known to be consistently collected and normalised across all dates and all customers, and free of erroneous measurements.

Unfortunately it is not always possible, in the real world, to get clean reliable data such as this. Where improvements such as those above are impossible, there may be more sophisticated ways of trying to detect erroneous or inconsistently recorded data than have been described in this thesis - for example, automated methods to find out which customers have dubious records in a certain month, rather than rejecting all the data for a month which appears to have some dubious entries.

A more general way of removing (or severely down-weighting) outlying data points than the somewhat crude solecism detection of section 11.3.1 would also be desirable. One way to remove all variety of extreme outliers would be to build a preliminary model for whole load (composed from the weather-dependent model and the weather-free clustering model, or by just applying the decision tree clustering technique to whole loads), and then identify outlying data in the original dataset as those that the constructed model predicts very poorly. Single half-hourly data points, or whole profiles, or whole atoms, or whole customers/dates, could be removed or down-weighted automatically if their

Euclidean distance (say) from their predicted values in the preliminary model was too great. Examining which data were removed by such a process may be revealing in itself, and would also allow a secondary model to be constructed from the cleaned/weighted data which was less distorted by outliers.

14.2 Enhancements to Weather Model

One problem with the weather modelling methodology presented in Chapter 11 is that it relies on an estimate of available natural illumination that is by no means accurate, together with cloud coverage figures and time of day/year information, to assist the modelling of lighting loads. If actual figures for illumination could be collected, the model might improve, and we might also be able to do without time-of-year variables, relying more on meteorological variables to model seasonal variations in load.

However the greatest problem with the presented model is that it can take extremely long times and vast amounts of memory to calculate; this is especially the case when one or more categorical customer variables are used as predictors, since then the number of data points increases n-fold when there are n distinct combinations of customer variable values present. This made it impractical to use SIC code in large models, or to use two customer variables at once.

Since it would be desirable to build weather models over longer periods, and for more customers, than were present in the databases provided, ways to reduce the memory and CPU-time requirements of the presented weather methodology might need to be found. A prior clustering of the customers' whole or weather dependent profiles, using customer variables as extrinsic variables in a mixed atomic/subatomic clustering, could be used to generate a new customer variable, *weather dependence category*, whose value was determined by which leaf cluster a customer belonged to in this model. Provided that the number of clusters (hence the number of values of the weather dependence category) was reasonable, then load factor category, tariff code and SIC code could be replaced by a one categorical customer variable, perhaps allowing for improved weather models without too much additional computational complexity.

Another area of research would be to establish how much goodness of model fit is sacrificed when various variables are excluded from the weather model. It may be possible to achieve a similar goodness of fit using a smaller variable set, thus reducing the computational burden of the method.

If the amount of data to be modelled was so great that there was no way to maintain computational feasibility within MARS, a less complex method (such as interaction splines - section 6.5.1 - featuring just the variable interactions most frequently chosen by the MARS models in this thesis) might need to be adopted. Categorical variables could be employed in such a scheme by building separate models for each ‘weather dependency category’ (see above) of customers.

14.3 Enhancements to Deweathering

A problem with the presented methodology of modelling weather dependent loads and then deweathering whole load by subtracting the weather model is that every customer with the same or with sufficiently similar customer categorical variables will be assigned the same weather model; in fact, if customer categorical variables are not used as predictors to MARS, then *all* customers are assumed to have the same weather model. Thus the deweathered loads for a given customer, which consist of subtracting the weather model from the customer’s initial whole loads, may in fact overcompensate for the effects of weather. In particular, some customers may have very little weather dependency in their loads relative to the majority of customers, and hence have their winter loads and/or their summer loads artificially lowered in the deweathered data for no good reason.

Whilst this fact is largely disguised in the presented clusterings of deweathered loads (because each customers profiles are composed into atoms with other customers, so that the extent of an individual customer’s weather dependency becomes blurred), it could be an important source of bias where an individual customer’s loads are important, such as in the subatomic clustering phase of a mixed atomic/subatomic clustering model.

The use of a ‘weather dependence category’ variable determined by clustering weather dependent customer profiles (as discussed in section 14.2) might help to reduce this problem. However it might also be possible to do something about it at the deweathering stage; a customers’ deweathered loads could be generated from its whole loads by subtracting a *scaled* version of the weather model, using a different scalar λ_j for each customer c_j; customers with less weather dependence would employ smaller scalars. If a customer’s N deweathered load readings $Y^{WF}{}_i$ are calculated from their original whole loads Y_i using modelled weather dependent loads f_{AW_i} using

$$Y^{WF}{}_i = Y_i - \lambda_j f_{AWi} \quad \text{(EQ 124)}$$

for $1 \le i \le N$, then we can determine an appropriate λ_j for each customer c_j so that the deweathered loads $Y^{WF}{}_i$ appear as uniform throughout the year as possible; an obvious criterion for maximising the degree of uniformity of c_i's deweathered profiles throughout the year (with respect to λ_j) is to minimise

$$\sum_{i=1}^{N} \left(Y^{WF}{}_i - \overline{Y^{WF}}_i \right) \quad \text{(EQ 125)}$$

where $\overline{Y^{WF}}_i$ is their average deweathered load, which since the weather model is very nearly zero sum, can be replaced with their average whole load. It would be fairly straightforward to minimise this criterion with respect to the single coefficient λ_j.

14.4 Improvements to Decision Tree Clustering Model

14.4.1 More Alternatives to Load Factor

We have already seen how replacing load factor with a percentile load factor can improve overall scatter accounted for by the model, and that percentile load factor is generally a more useful predictor in the presented extrinsic decision tree clustering technique than conventional load factor.

There might be some mileage in considering other measures of a customer's profile flatness/uniformity other than load factor or percentile load factor. One problem with these measures is that they do not differentiate between, on the one hand, customers whose daily load total varies greatly from day to day, and on the other hand, customers whose daily load total does not vary much but whose typical peak load each day is much greater than their mean load each day. Thus we might desire two measures of profile uniformity, one describing typical daily profile uniformity, the other describing typical annual uniformity of daily load.

One statistical measure that is of possible interest is the *skew* of a customers loads (either the skew of their mean profile or the skew of their individual half hourly loads over the period of study). Whereas a mean describes a typical value and a standard deviation describes how much values typically stray from the mean (the *amount* of variation), skew describes the amount of asymmetry in that variation. High load factor customers generally have a more negative skew than lower load factor customers.

14.4.2 Alternative Metrics in the Decision Tree Clustering Algorithm

There are three principal criteria which dictate the final form of the extrinsic decision tree clustering models presented in this thesis. Firstly, there is the ‘next node’ criterion deciding which node should be split next; secondly there is the distance criterion between the centroidal profiles in the clustering algorithms used to split that node; and finally there is the tree size determination criterion.

In the methodology presented, total Euclidean scatter amongst all the constituent profiles at a node was used to determine which node to split; a biased Euclidean distance, which discriminated against clusters with uneven numbers of constituent profiles was used as a distance metric in the clustering algorithms; and the tree was grown only until it reached a predetermined size.

There is a great deal of research which could be done on comparing these criteria with a several alternative criteria. The Euclidean scatter amongst the underlying original profiles at a node might be replaced by Euclidean scatter amongst the underlying atomic profiles at a node, in the node selection criterion. Euclidean scatter is not robust to outliers, and a distance metric less punitive to outlying data could also be considered.

More ambitious would be a scheme which found the best binary clustering it could, not at just one node, but at many. Then whichever of the binary clusterings at each of those nodes was judged best would be the node that was split. This would require rather more calculation, however, than the current scheme.

The distance criterion used (modified Euclidean) is also very sensitive to outliers, and less punitive measures could be tried.

In the current scheme, when one binary clustering has been determined for each candidate variable, the ‘best’ variable is chosen to be that whose distance between the binary clusters is greatest; however, depending on the goodness of model fit criterion applied, this might not always be the split which most reduces lack of fit globally; looking at various criteria for overall model goodness-of-fit (rather than always choosing the binary clustering which satisfies a local goodness of fit criterion) is another possible area of research.

Rather than stopping at a fixed sized tree, an overgrowing and pruning approach may yield better results. A more complex system involving repeatedly overgrowing, then over-pruning, then overgrowing again and pruning again, repeatedly until no model improvements occur, may also be worth investigating.

An advantage of growing a tree according to one criteria and pruning according to another is that the local greediness of the growing criteria may be corrected by a globally determined goodness of fit criterion applied in the pruning phase.

A major extension to the work in this thesis would be to perform thorough cross-validation experiments to determine the best size tree, and to determine the best values for various parameters, including the bias coefficient b. n-fold cross validation would involve randomly dividing the customers into n sub-populations, as described for the MARS cross-validation scheme, and testing n models, each built using $\frac{n-1}{n}$ of the data, against the remaining $\frac{1}{n}$th of the data. The lack of fit arising when comparing what each test profile should look like according to the model to what it actually looks like, would be the criterion by which the model size and various model parameters would be determined. Note, however, that other criteria than minimising cross-validation errors are also important; an engineer, for example, may require a fixed number of profiles for a certain task, in which case the final model size is not flexible; and various ratios between the number of splits and the number of subatomic splits may be desirable depending on to what extent the final clusters need to be dictated by known customer attributes.

Another major area of research which could be investigated with a view to extending or adapting the clustering methodology would be information theoretical measures for load profiles. Due to the stochastic nature of load profiles (the load at time t in a profile is certainly not statistically independent of the loads at other times t', particularly when $|t-t'|$ is small) choosing meaningful estimators for quantities such as (i) the self-information of a profile (ii) the transinformation between profiles, and (iii) information gained by splitting a profile according to the values of that variable, are very difficult to determine.

In section 3.6 the concept of band limitation was used as a simplifying assumption about stochastic ensembles in order to derive meaningful information theoretic measures for them. Other simplifying assumptions included time limitation, and independent identically distributed Gaussian additive noises. How appropriate is the assumption of band limitation when applied to 48 half-hour load profiles? And since the highest frequency we can investigate is limited very much by the sampling frequency for the load profiles (i.e. half-hourly), would the concept of band limitation be useless anyway?

Since entropy is defined as the smallest theoretical storage space (in bits) for a signal under a reversible coding (which is another way of saying a reversible compression technique), information theoretical measures for load profiles might be possible which are

based on the number of points which are necessary to reconstruct the profiles (to within a certain tolerance), just as the sampling theorem for band limited continuous signals describes the number of points necessary to reconstruct a band-limited signal in the formulation of Shannon's theorem for the transinformation of band-limited signals (EQ 20). The number of knots required by a given cubic spline fitting technique to model a load profile to within a certain accuracy might be used in entropy-like or transinformation-like measures for load profiles.

How to best use these pseudo-information theoretic measures in a decision tree clustering procedure would require investigation, though there are many well known information theoretical decision-tree classification/clustering techniques (for *categorical* responses) on which to model such a procedure.

14.5 Application of Methods to Other Databases

Of the three profiles databases discussed with the Sponsor (see section 9.1), the one studied has the least complexity (the fewest number of predictors). The techniques presented would be applicable to more complex databases which include questionnaire data (and/or other variables) without major modification: the non-weather predictors could be employed in the weather-free (cluster analysis) model in exactly the same ways. More discrete predictors would entail more atoms, which could present complexity problems, though these problems might be overcome by using cheaper algorithms (say, the Binary Splitting Algorithm in place of the Join-Two algorithm) towards the top of the decision tree.

The weather modelling part of the methodology might be put under particular strain if applied to databases for which there were many more categorical predictors (such as domestic customer databases accompanied by questionnaire data on end uses and family make-up), and it seems certain that the number of categorical variables would need to be reduced (probably by extrinsic clustering of weather dependent loads, as discussed in 14.2) before the categorical information could usefully be incorporated.

14.6 Discovered Predictors

One of the most important predictors of winter load shape after day type is the presence or absence of a storage heating load for a given customer. No variable recording the presence or absence of storage heating loads for each customer is recorded in our month-

ly billed business customer database, but it would probably not be too difficult to construct such a predictor by examining loads at and shortly after the onset of night time cheap-rate supply. If a customer has cheap-rate loads which are significantly higher during spells of cold weather, this is almost certainly due to a storage heating load. A discovered discrete variable recording whether or not a customer has storage heating would be particularly useful in the weather dependent model, and of possible use in the weather-free model; it might even be feasible to discover a continuous storage heating variable which estimates the percentage of annual load due to storage heating devices for each customer, for use as a continuous regressor in the weather dependent model.

Similarly it might not be difficult to discover the presence or absence of air conditioning and/or storage air conditioning loads for each customer; where a customer's daytime loads have a significant positive correlation with temperature and/or humidity, space conditioning is almost certainly used by that customer. Where night-time cheap rate loads are significantly correlated with daytime temperature/humidity, storage space conditioning is almost certainly installed. Such discovered variables could be incorporated into customer databases, and might have uses other than in load profiling tasks.

Chapter 15 — Summary and Conclusions

The load profiling task described in this thesis covers a large number of customer, temporal and meteorological variables, both supplied and derived. Because there are many variables of potential importance to the task, particularly in the case of weather variables where there are many derived candidate variables, the task is very large. High dimensional modelling tasks present computational difficulties, and are also much harder to interpret than low dimensional problems. Partly to keep the dimension of the problem in check, partly to allow for improved interpretability, and partly because different types of model are better suited to modelling different relationships, a scheme was devised which separates a weather-dependent component of load from a weather-independent component.

The chief difficulties of the load profiling task, aside from the high dimension of the problem, arise from the extreme heterogeneity of response in the data. Different customers may have dramatically different load shapes on a given day, and a customer's load shape may vary dramatically from day to day, from season to season, and with differing weather conditions. This problem is exacerbated by the fact that even customers with the same tariff code category, and/or the same SIC code, and of similar load factors, cannot be expected to always have the same load shape characteristics or weather dependencies. Another major problem with the particular load profiling task studied here arises from the poor state of the loads database. Although some measures were employed to automatically remove probable erroneous data, and visual inspection employed to detect contiguous dates of questionable data, better results would be expected from cleaner databases.

A non-parametric and highly adaptive data mining regression algorithm (MARS) was employed to model the effects of weather on load, separately from the principal effects of the other variables on weather independent load; the residuals from this model are assumed to be due to non-weather variables, so are recombined with the weather independent loads prior to the second phase model, the model for the weather-insensitive portion of load. A variety of different combinations of supplied and derived weather and temporal variables were made available to the model, and various parameters varied, in order to obtain good model fit, whilst guarding against overfitting the data.

The biggest drawbacks of the use of MARS for the load/weather analysis are its high computation times and high memory demands when categorical customer variables are

used. This is a peculiarity of the task rather than a general problem with using categorical variables in MARS; because every customer is considered as experiencing the same weather conditions at any given time, the number of data points in the model can be hugely reduced by aggregating the loads for all customers; but when a variable which disaggregates the customers is supplied, the number of data points grows, and does so nearly exponentially as more such variables are supplied.

However, the weather modelling methodology presented proved itself capable of accounting for a great deal of the variation in weather dependent load, with or without categorical customer variables. In particular, the order 2 and order 3 interaction terms generated by MARS frequently corresponded to known phenomena in the load/weather relationship (such as the combined effects of humidity with temperature, of cloud with time of day and year, of windspeed and temperature, and the order three interaction of windspeed, humidity and temperature); indeed, MARS appeared to be as good at modelling such effects as summer discomfort and wind chill by itself (synthesising high order terms as necessary) as when variables representing these concepts were explicitly provided.

Exponentially smoothed versions of the weather variables, particularly medium and long term smooths of temperature, proved to be important in the generated models. In fact, medium and long term temperature smooths were generally rated as more important than the current or very recent temperature. Lagged versions of the weather variables generally proved much less useful than smoothed versions (though the maximum, minimum and mean temperatures from the previous day often proved to be of much value), and delta temperatures were only of much use when temporal variables were excluded; there was no evidence that delta variables were necessary to model temporal asymmetry in the model.

It is only the highly adaptive nature of a model like MARS that allows so many interactions of so many variables to be considered at the same time; since new variables and new interaction terms are only introduced on a local basis where they are shown to reduce lack of fit, it is possible to consider many more multiplicative combinations of variables than could reasonably be considered in a non-adaptive regression scheme.

The introduction of categorical variables into the weather model, though limited in its scope due to the computational difficulties mentioned above, appeared to be very successful. All of the categorical customer variables introduced were found to be useful predictors of load/weather behaviour; load factor (which was only tested in a 1-percentile

version) looked to be a little more effective as a predictor than tariff code, but SIC code (which could not be tested in a model of comparable size) was picked more frequently than any of the other categorical variables in a smaller experimental trial, and might be the categorical variable with the most predictive power in the weather model.

More investigation is necessary in order to determine a way to present more categorical customer information to the load/weather model without generating computationally impractical models; a prior clustering of weather dependent loads to obtain categories of customers with similar load weather relationships has been suggested as a major extension to the weather modelling methodology.

An adaptive decision tree clustering technique which recursively subdivides the domain using locally appropriate binary clustering algorithms, and which models the data at higher resolutions where the data is locally most heterogeneous, was devised especially for the task of modelling the (nominally) weather-free loads generated using the weather model. A biased distance measure was found to be required in order to discourage uneven clusters (which generally account for less scatter and are less informative) occurring early on in the tree building process, and this resulted in great improvements in the resulting models, in terms of interpretive power as well as scatter accounted for. Alternatives to conventionally computed load factor were tested as predictor variables, and significant improvements in the amount of scatter accounted for, and the speed with which scatter was accounted for, were observed.

A scheme which attempts to seek interesting patterns existing at the leaf clusters of the extrinsic decision tree clustering was implemented and tested. The principal motivation behind this is the observation that customers that, because of their load factors, tariff codes and SIC codes, will often end up in the same leaf of an extrinsic atomic decision tree clustering, will sometimes have very different load shapes. By freeing the clustering sub-algorithms employed in the later stages of a decision tree clustering from the need to keep profiles from the same atom together, clusters are generated that account for significantly more scatter that when the same sized tree is built using only atomic clustering.

The much improved fit resulting from employing subatomic clustering in the latter part of modelling indicates that there are ‘hidden’ patterns in the Sponsor’s business customers’ load profiles that cannot be isolated using the recorded customer attributes alone. It is anticipated that close investigation of the customers found in the subatomic leaf clusters would expose certain types of customers with unusual load shapes that it

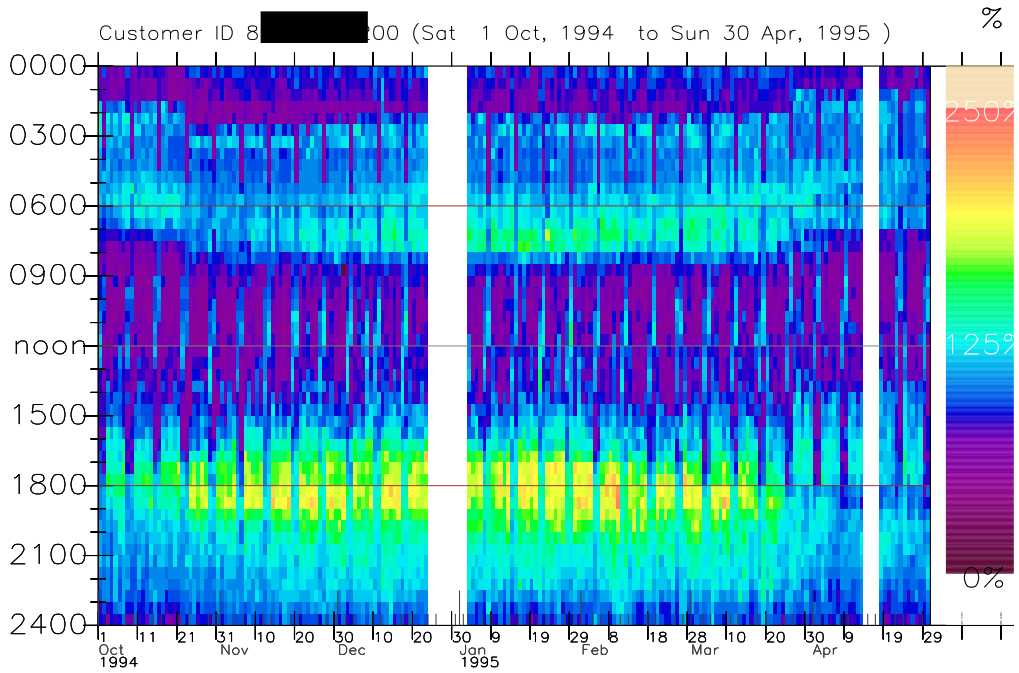
might be of special benefit for the Sponsor to try and court.

The modelling procedure described satisfies the principal stated aims of the load profiling task: to build models which estimate, for certain subsets of customers, their load shapes (and confidence estimates for those load shapes), for different weather conditions, times of year, and days of the week. The leaf profiles in an atomic or a mixed atomic/subatomic decision tree clustering serve as a set of standard profiles, which can be used as a tool in determining tariff policies and identifying patterns in load shape. Additionally, the structure in the variation in load shape can be visualised using the decision tree, and the relative variable importance determined.

The load weather model can be applied on top of the weather-free clustering model (by simply adding the relevant profiles from either part of the model) to determine a predictive model for load shape given a particular customer type and a particular set of weather conditions. This could be of use in predicting the probable demand surplus/deficit arising from unusually cold or mild weather conditions, and of predicting the overall demand profile at any given time of year, given hypothetical changes in the proportions of differing types of business customers supplied.

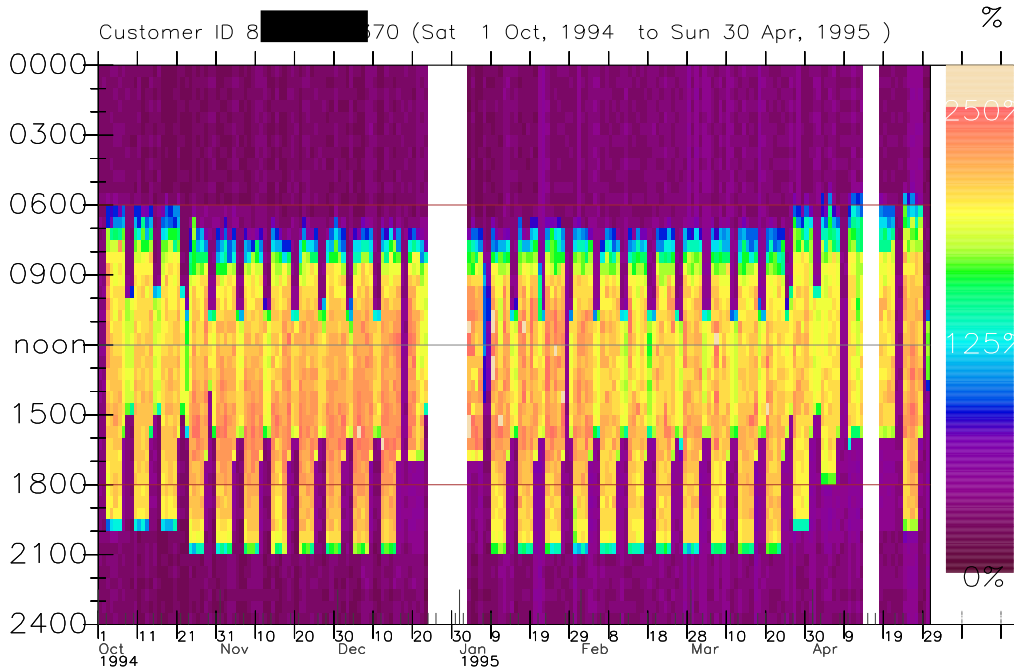
Appendix — Colour Figures

COLOUR FIGURE 1.



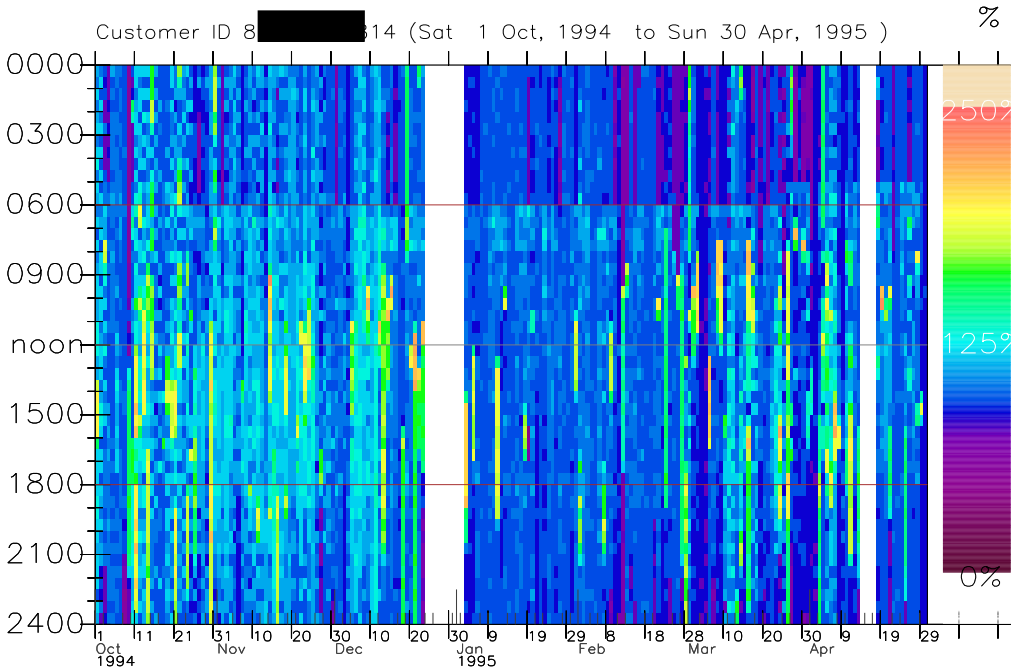
October 1994-April 1995 whole load profiles. The customer's two figure SIC code is 55, listed as "Hotel & Restaurant". The z-axis (i.e.colour) runs from 0% to 250% of mean half-hourly load.

COLOUR FIGURE 2.



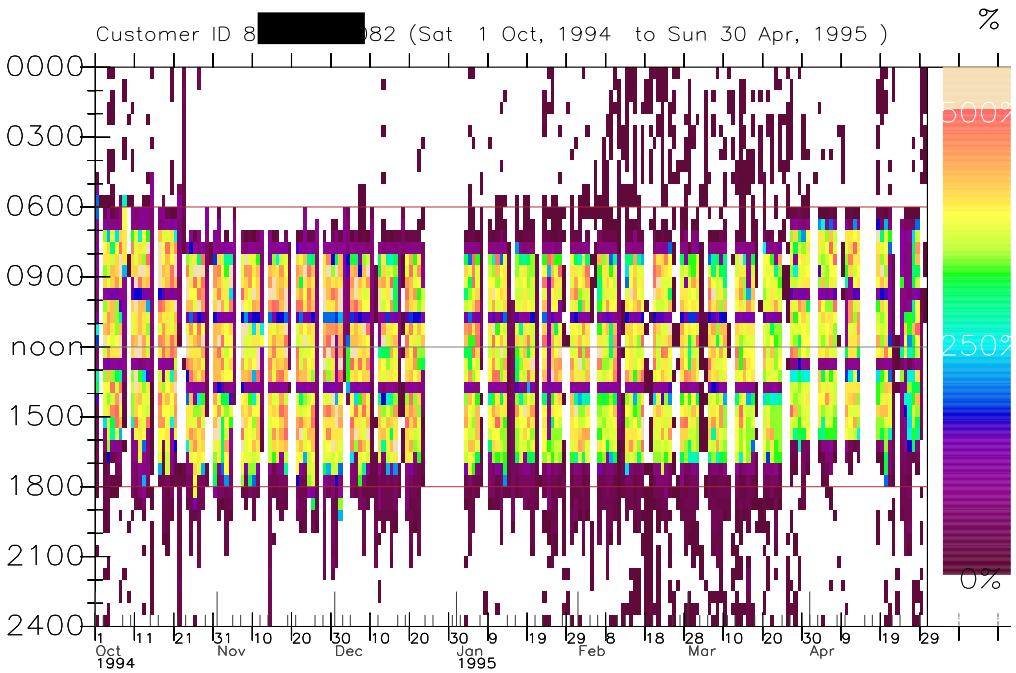
Profiles for the same period for a customer with SIC code 80, which is listed as "Education".

COLOUR FIGURE 3.



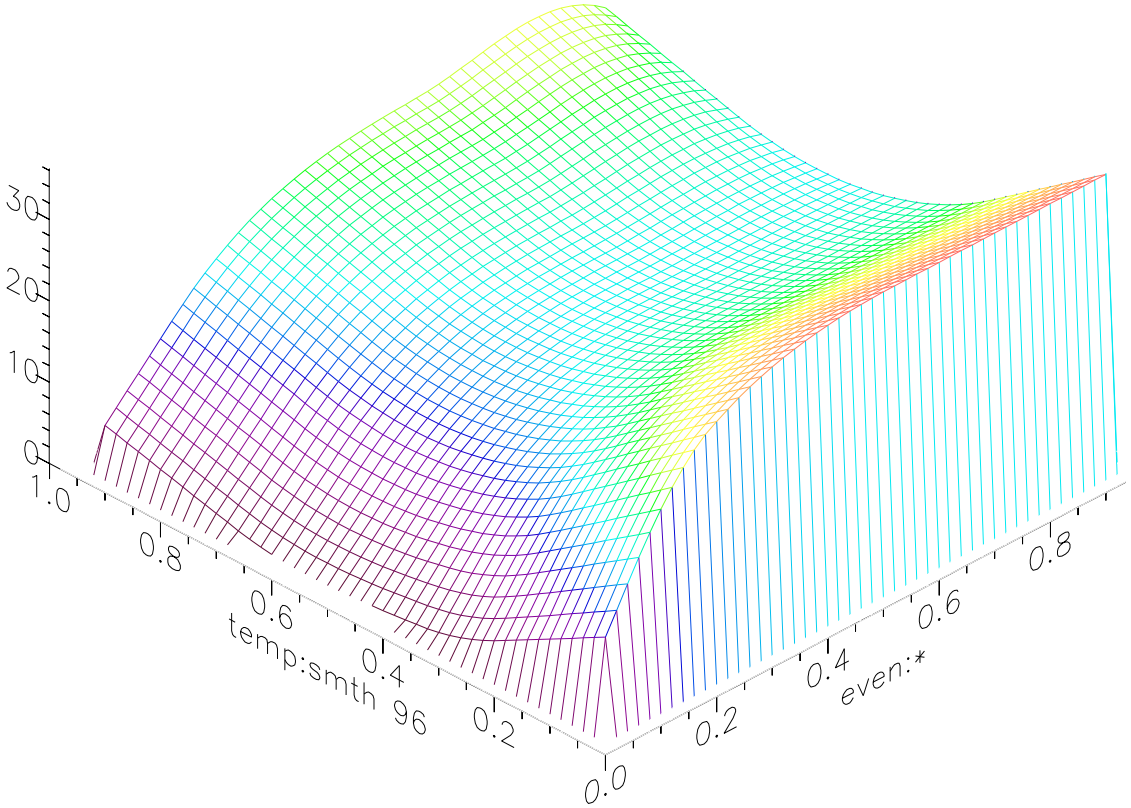
This customer's SIC code is 74, which is listed as "Legal & Marketing". There is very little discernable pattern to the load shape.

COLOUR FIGURE 4.



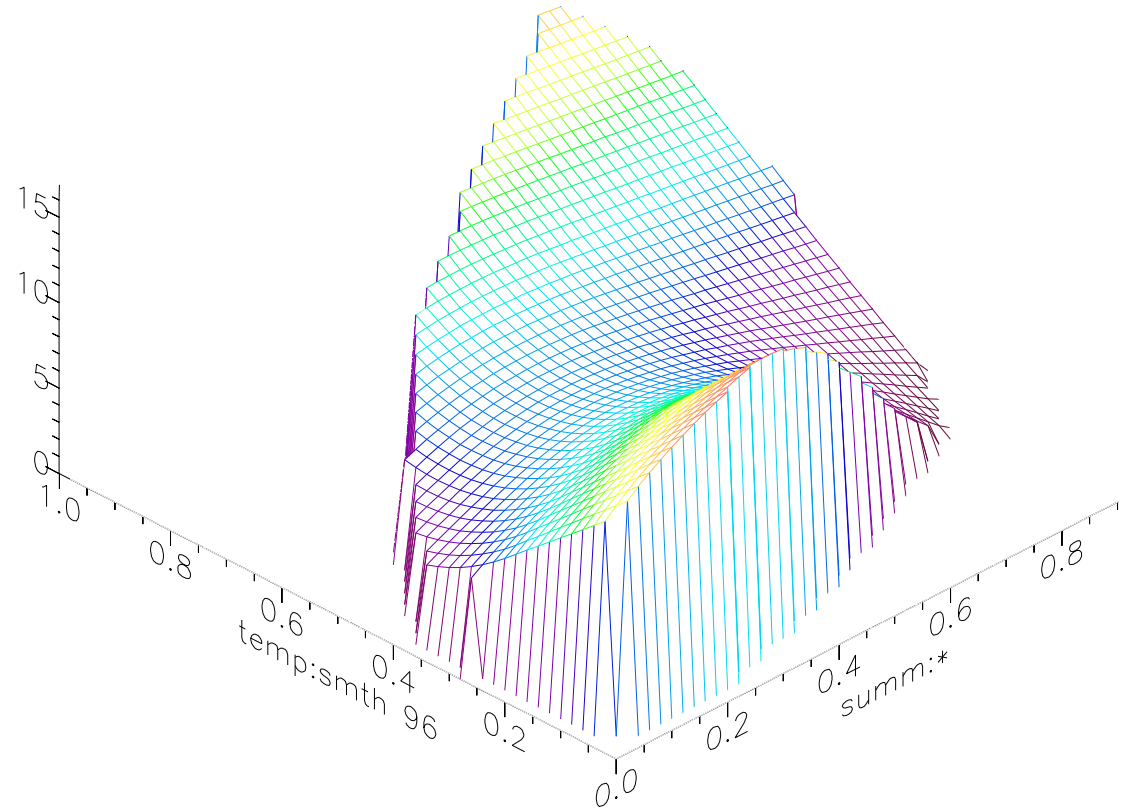
This customer's SIC code is 52, listed as "Retail & Repair". Note that the z-axis (represented by colour) is on a different scale (0% to 500% of mean half-hourly load) as the customer's load factor is very low. Much of the time the customers load is recorded as 0.

COLOUR FIGURE 5.



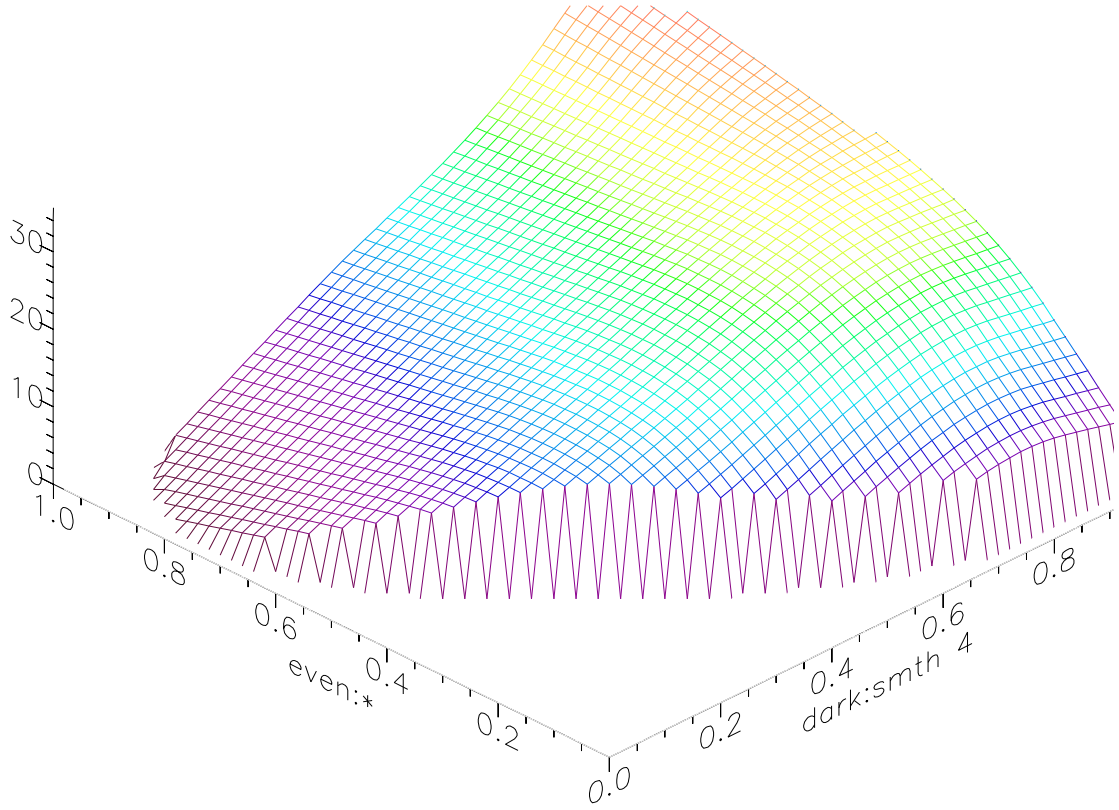
MARS ANOVA plot for 48 hour half-life smoothed temperature and closeness to evening.

COLOUR FIGURE 6.



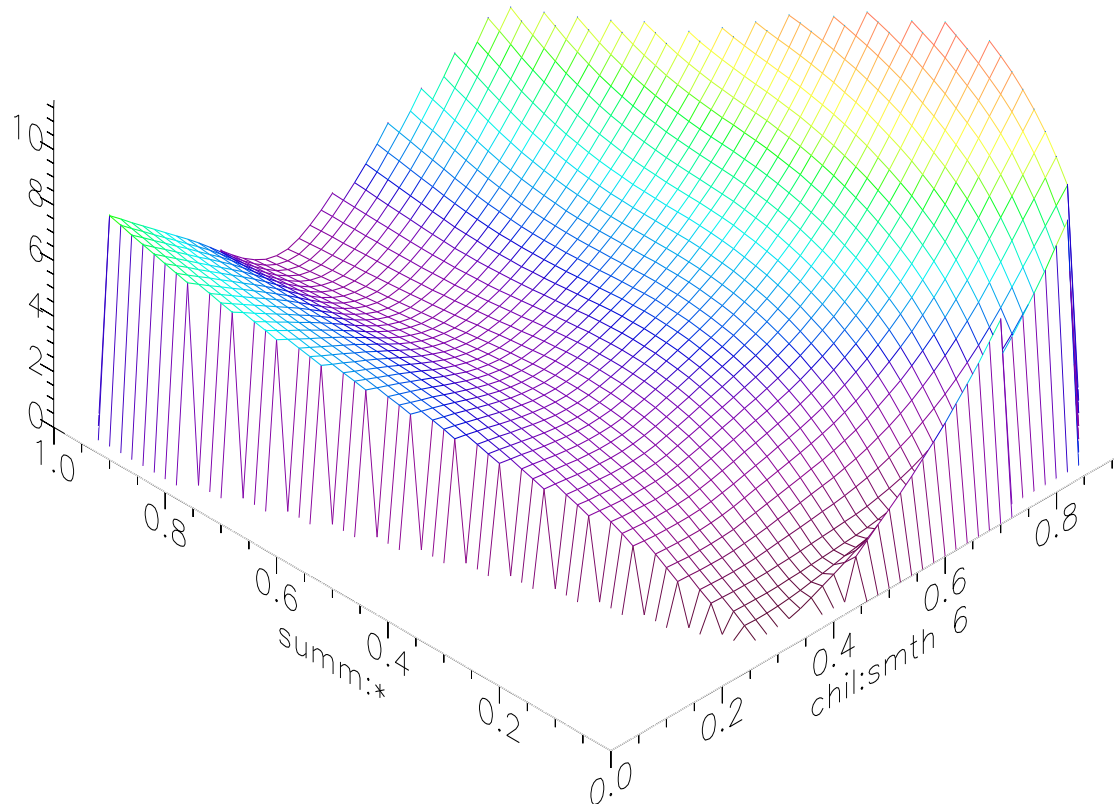
MARS ANOVA plot for 48 hour half-life smoothed temperature and closeness to summer.

COLOUR FIGURE 7.



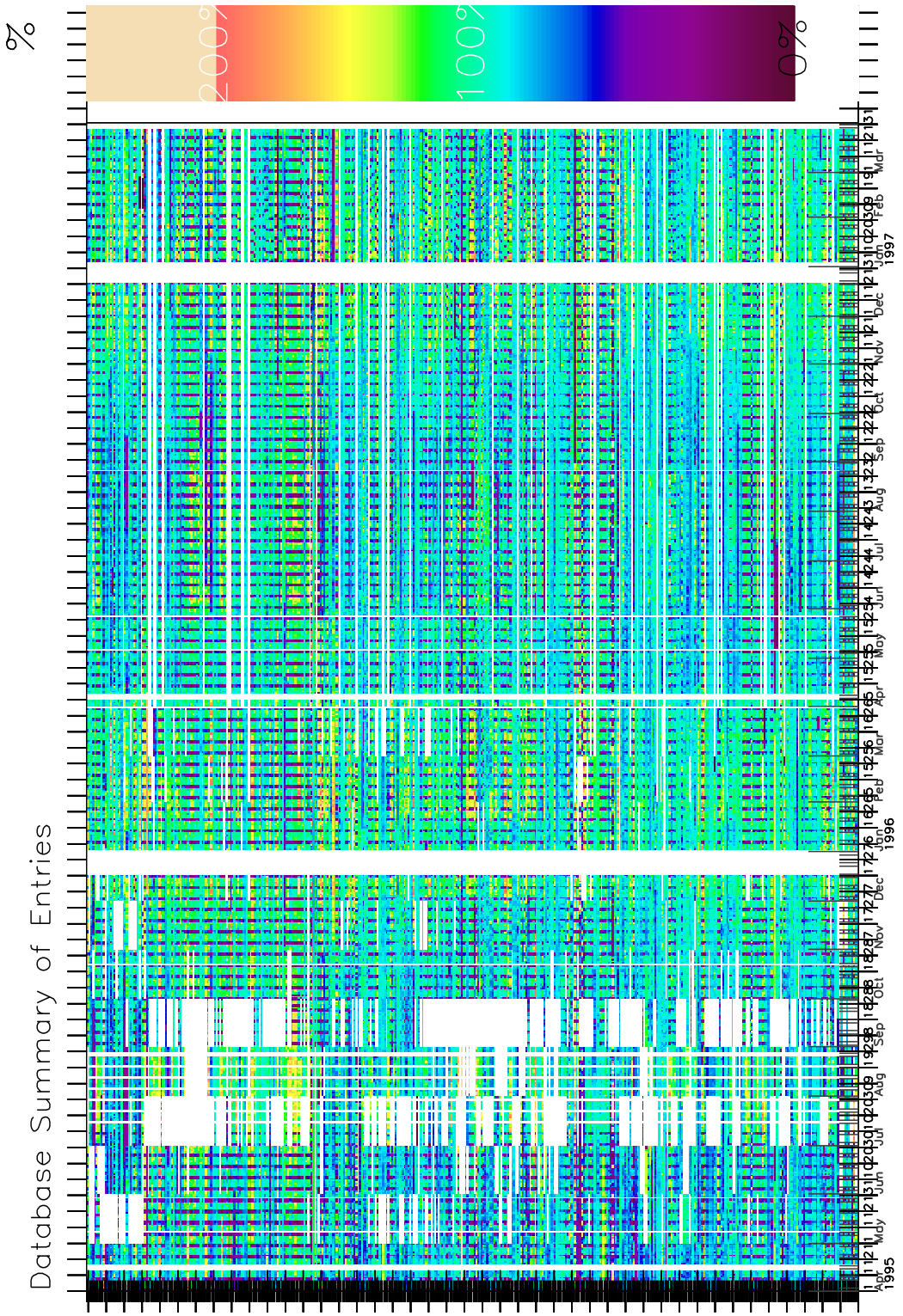
MARS ANOVA plot for closeness to evening and 2 hour half-life smoothed estimated darkness.

COLOUR FIGURE 8.



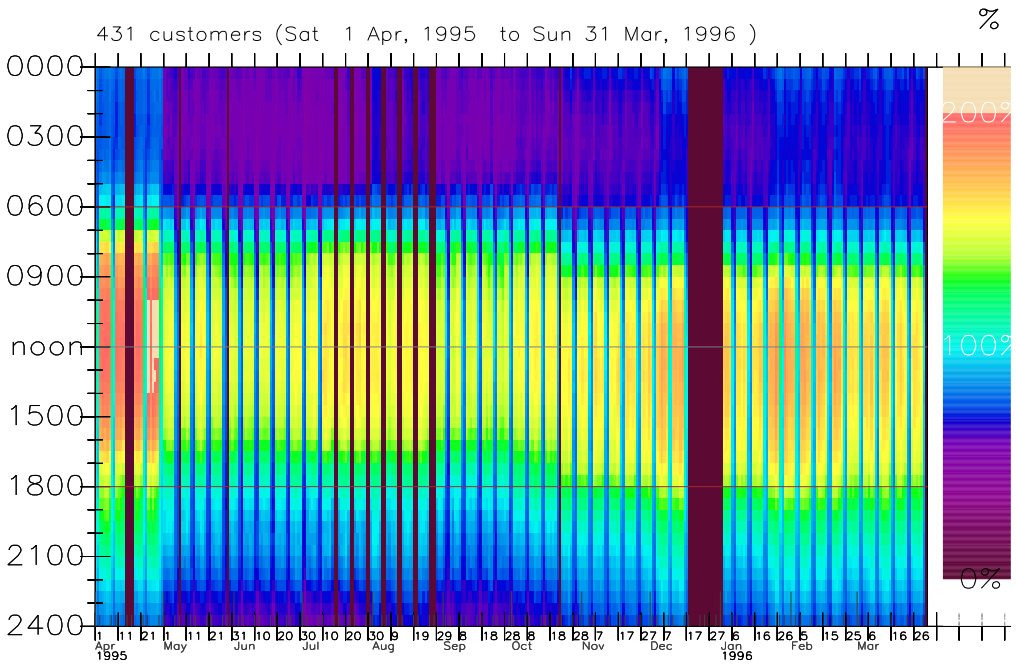
MARS ANOVA plot for 3 hour half-life smoothed wind chill and closeness to summer.

COLOUR FIGURE 9.



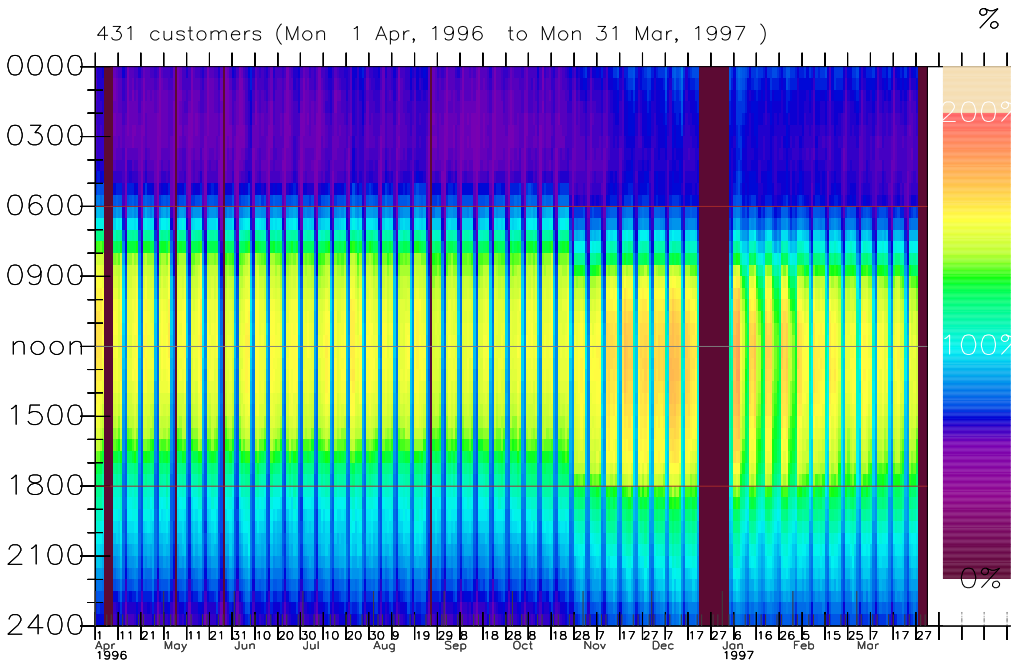
Overview of the entire database (whole loads), customer by customer. A customer's daily total load (represented by colour) is calculated as a percentage of that customer's *average* daily total load. A key between colour and percentage is provided. White represents missing profiles.

COLOUR FIGURE 10.



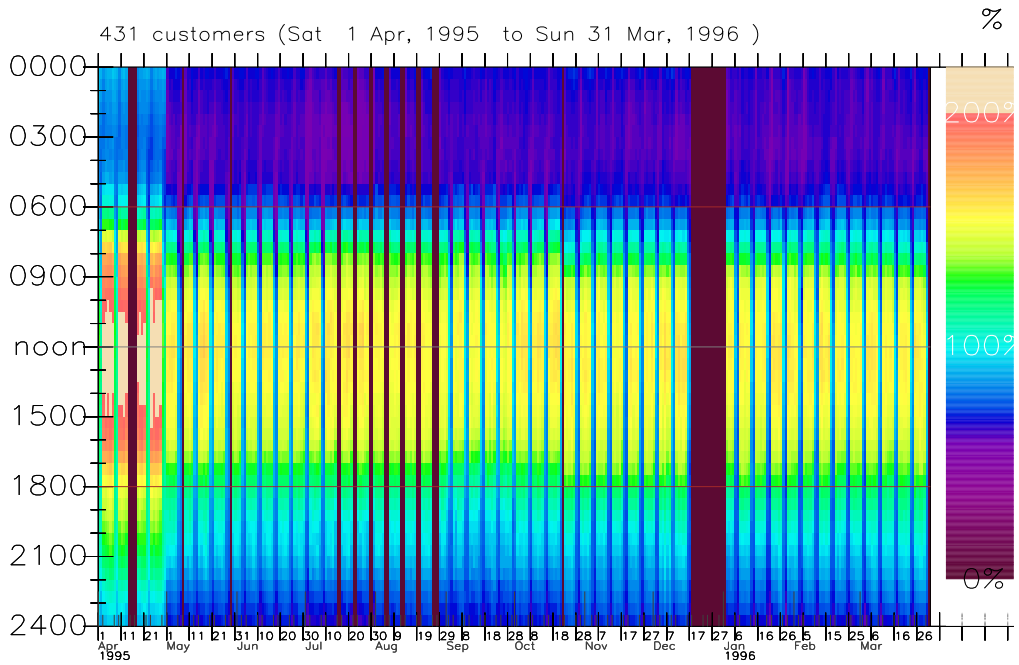
Overview of dataset 1 (April 1995-March 1996), i.e. whole (not deweathered) load. The data for April, July and August, have apparently been measured on different scales from the rest of the data. see 12.5.1 for notes on interpretation.

COLOUR FIGURE 11.



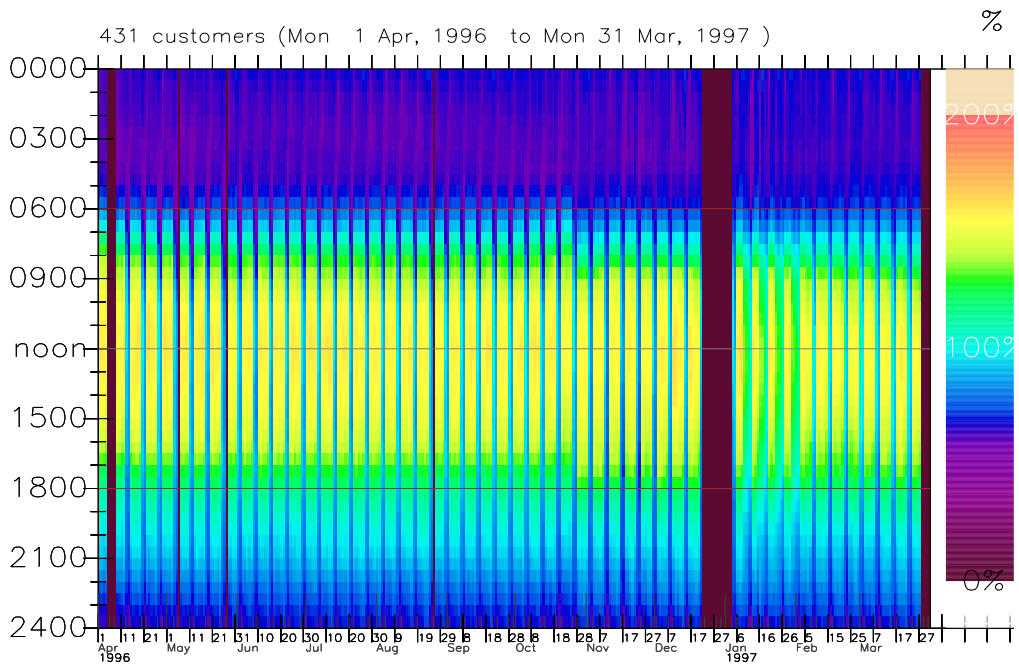
Overview of dataset 1 (April 1996-March 1997).

COLOUR FIGURE 12.



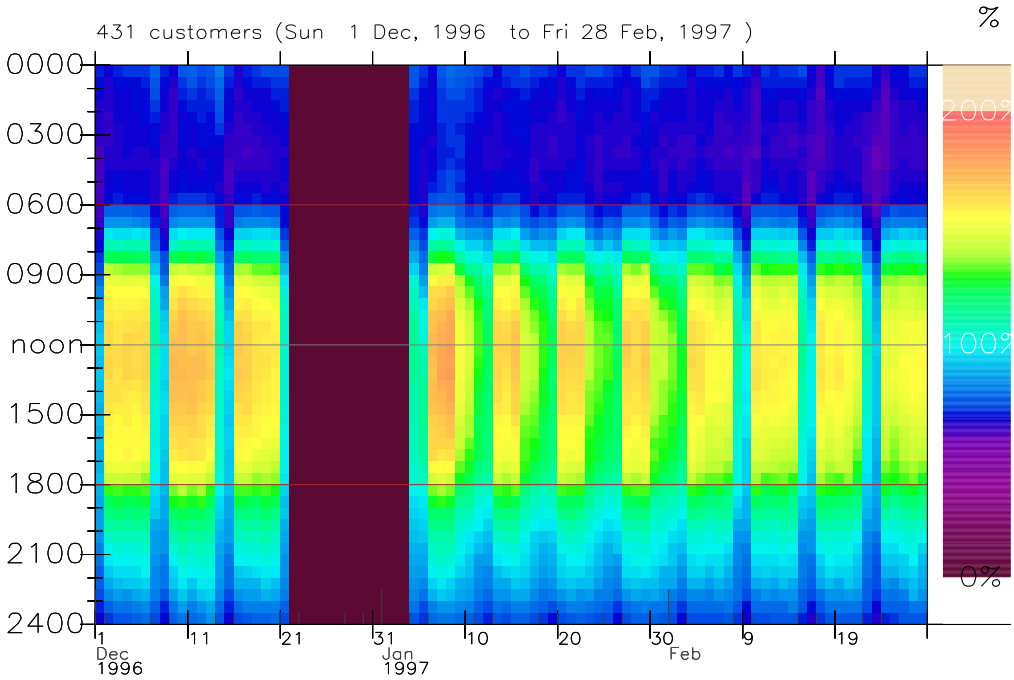
Overview of dataset 2 (April 1995-March 1996), which was deweathered using weather model α. Note the questionable data for all of April, July and August, which are even more apparent in the deweathered data than in dataset 1 (COLOUR FIGURE 10.).

COLOUR FIGURE 13.



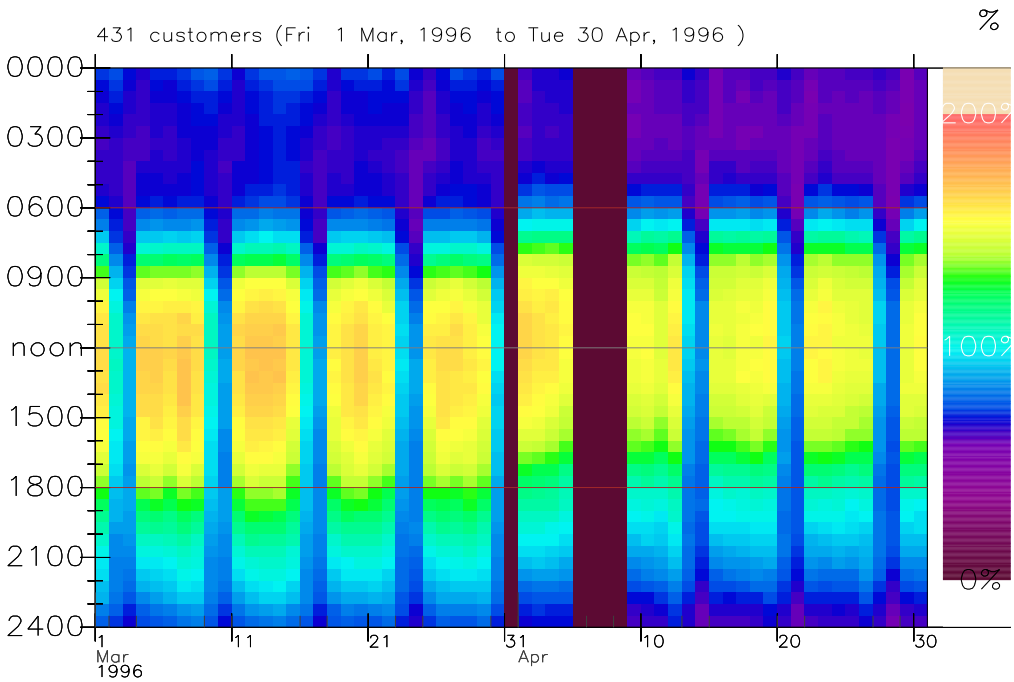
Overview of dataset 2 (April 1996-March 1997).

COLOUR FIGURE 14.



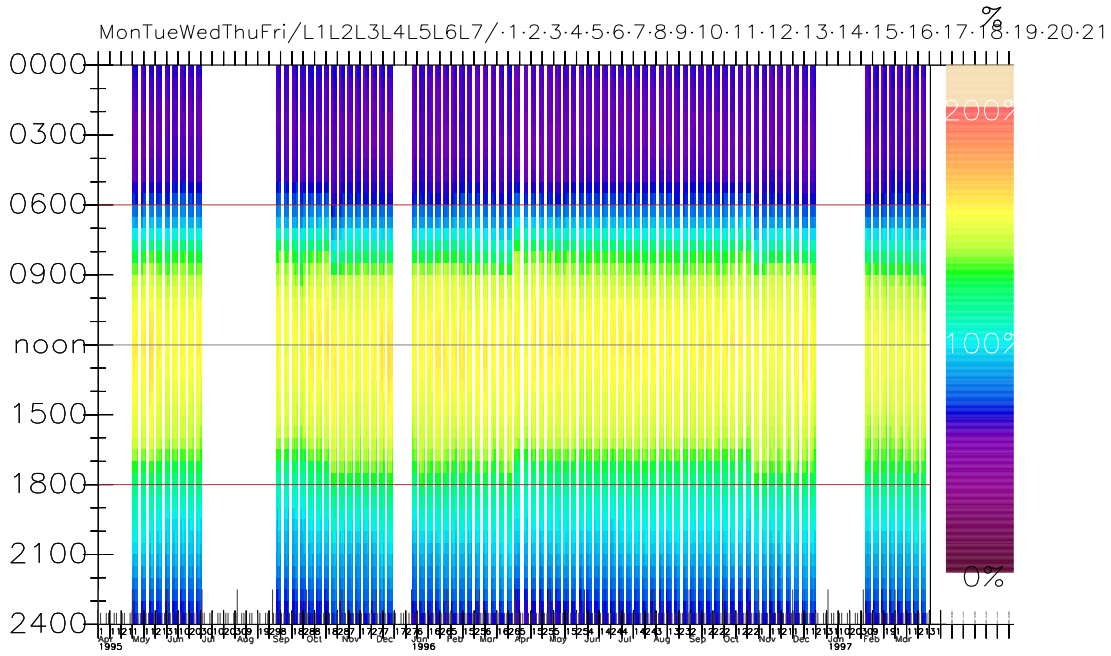
Whole loads (dataset 1), December 1996 to February 1997 - greater detail than (COLOUR FIGURE 11.). Thursday to Saturday profiles look highly suspicious during January 1997.

COLOUR FIGURE 15.



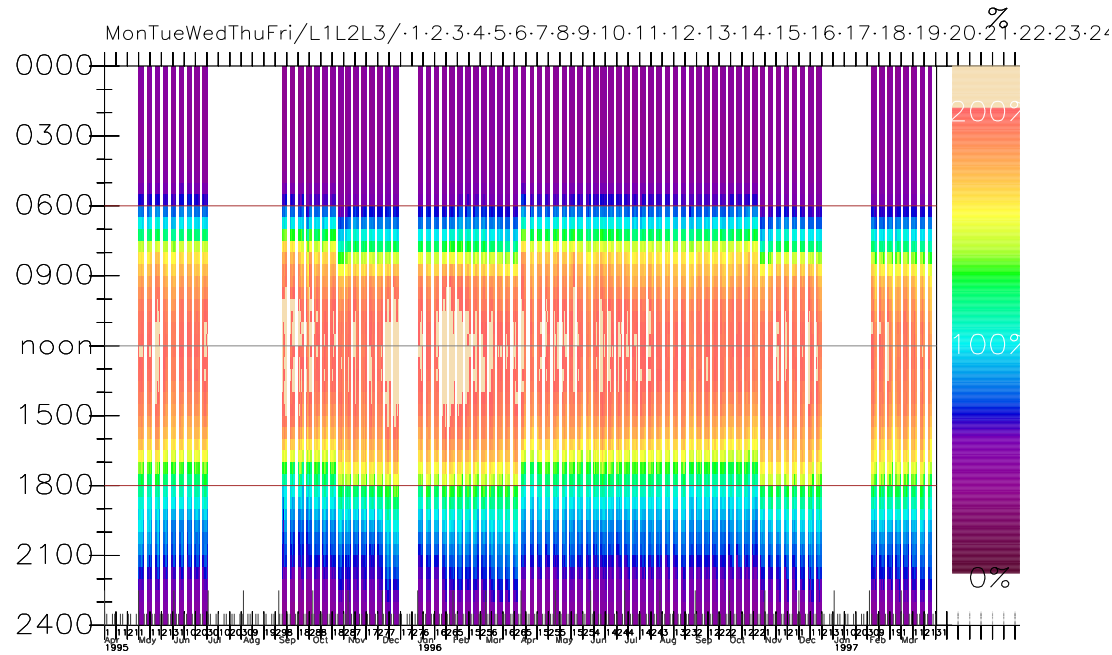
Whole loads (dataset 1), March to April 1996. The effect on loads of a daylight saving clock change in the early hours of March 31 is apparent.

COLOUR FIGURE 16.



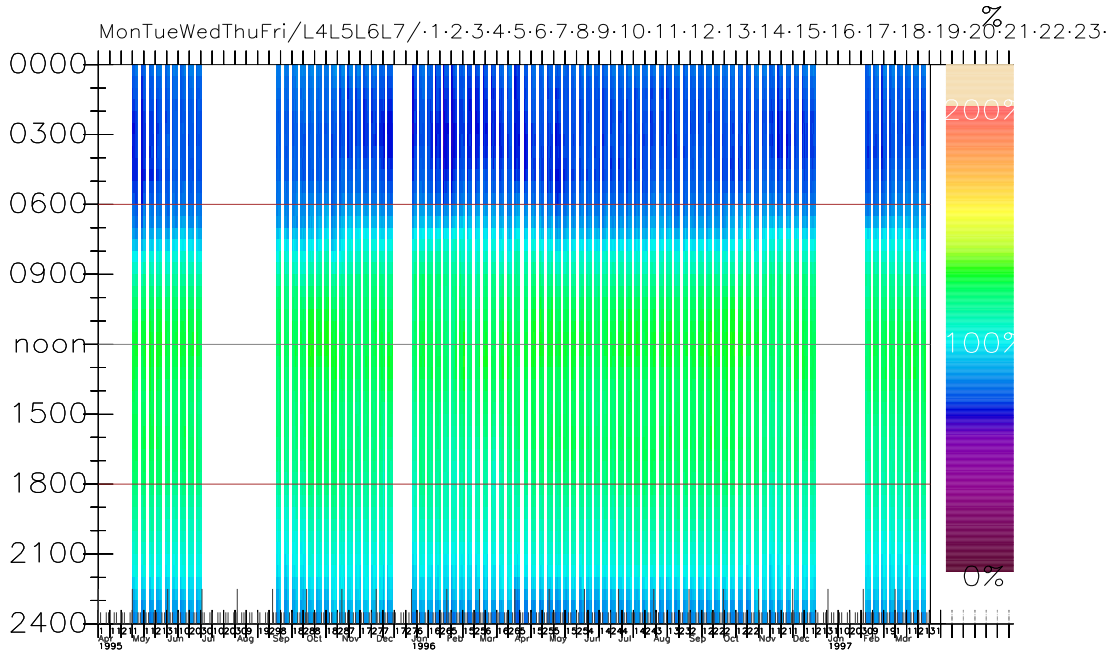
Seasonal profile overview for node 1 of (FIGURE 21.).

COLOUR FIGURE 17.



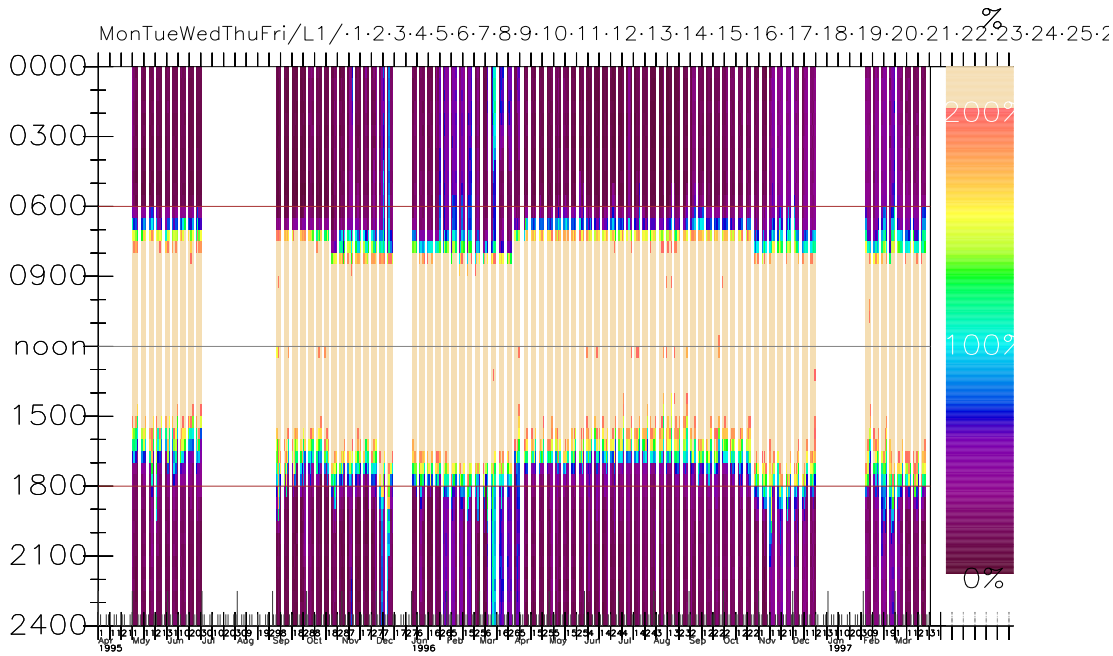
Seasonal profile overview for node 3 of (FIGURE 21.).

COLOUR FIGURE 18.



Seasonal profile overview for node 4 of (FIGURE 21.).

COLOUR FIGURE 19.



Seasonal profile overview for node 7 of (FIGURE 21.).

Bibliography

[1] W. J. Frawley, G. Piatetsky-Shapiro, C. J. Matheus, “Knowledge Discovery in Databases: An Overview”, *AI Magazine*, Fall 1992, Volume 13, Number 3, pp. 57-70.

[2] M. Holsheimer and A. Siebes, *Data Mining - The Search For Knowledge in Databases*, Report CS-R9406, CWI, Amsterdam, ISSN 0169-118-X.

[3] G. Piatetsky-Shapiro, W. J. Frawley (Eds.) *Knowledge Discovery in Databases*, AAAI Press, 1991

[4] F. M. Reza, *An introduction to information theory*, New York: McGraw-Hill, 1961 (McGraw-Hill electrical and electronic engineering series).

[5] G. Raisbeck, I*nformation Theory - An Introduction For Scientists and Engineers*, Cambridge MASS: MIT Press, 1963

[6] D. A. Ratkowsky, *Handbook of nonlinear regression models*, New York: Marcel Dekker, 1990

[7] R. L. Eubank, *Spline Smoothing and Non-Parametric Regression*, New York: Marcel Dekker, Inc., 1988

[8] P. Lancaster, K Salkauskas, *Curve and Surface Fitting: An Introduction*, London: Academic, 1986

[9] J. H. Friedman, “Multivariate Adaptive Regression Splines” (with discussion), *Annals of Statistics*, 1991, Volume 19, pp. 1-141.

[10]J. H. Friedman, *Estimating functions of mixed ordinal and categorical variables using adaptive splines*, Department of Statistics, Stanford University, Tech. Report LCS108, 1991

[11] J. H. Friedman, *Fast MARS*, Department of Statistics, Stanford University, Tech. Report LCS110, 1993

[12]J. R. Quinlan, “Induction of decision trees”, *Readings in Machine Learning* (Eds. J. W. Shavlik, T. G. Dietterich), Morgan Kaufmann, 1990

[13]J. A. Hartigan, *Clustering Algorithms*, New York: John Wiley and Sons, Inc., 1975.

[14]A. K. Jain, R. C. Dubes, *Algorithms For Clustering Data*, New Jersey: Prentice Hall, 1988

[15]G. Cross, F. D. Galiana, “Short-Term Load Forecasting”, *Proceedings of the IEEE*, Volume 75, Number 12, 1987, pp. 1558-1573

[16]A. B. Baker, *Methodology and Process of Forecasting Nominal Demand,* Electricity Pool of England and Wales/National Grid Company, Report 621.315 POO P

[17]I. Moghram, S. Rahman, “Analysis and Evaluation of Five Short-Term Load Forecasting Techniques”, *IEEE Transactions on Power Systems*, Volume 4, Number 4, 1989, pp. 1484-1491.

[18]K. Jabbour, J. F. V. Riveros, D. Landsbergen, W. Meyer, “ALFA: Automated Load Forecasting Assistant”, *IEEE Transactions on Power Systems*, Volume 3, Number 3, 1988, pp. 908-914.

[19]S. Rahman, R. Bhatnagar, “An Expert System Based Algorithm for Short Term Load Forecast”, *IEEE Transactions on Power Systems,* Volume 3, Number 2, 1988, pp. 392-399.

[20]A.S. Dehdashti, J. R. Tudor, M. C. Smith, “Forecasting of Hourly Load By Pattern Recognition - A Deterministic Approach”, *IEEE Transactions on Power Systems,* Volume PAS-101, Number 9, 1982, pp. 3290-3294.

[21]D. P. Lijesen, J. Rosing, “Adaptive Forecasting of Hourly Loads Based On Load Measurements and Weather Information”, *Proceedings IEEE Winter Power Meeting*, 1971, Paper 71 TP 96-PWR.

[22]W. R. Christiaanse, “Short-Term Load Forecasting Using General Exponential Smoothing”, *IEEE Transactions on Power Apparatus and Systems*, Volume PAS-90, Number 2, 1971, pp. 900-910

[23]N. D. Hatziargyriou, T. S. Karakatsanis, M. Papadopoulos, “Probabilistic Calculations os Aggregate Storage Heating Loads”, *IEEE Transactions on Power Delivery*, Volume 5, Number 3, 1990, pp. 1520-1526.

[24]T. M. Calloway, C. W. Brice, III, “Physically Based Model of Demand with Applications to Load Management Assessment and Load Forecasting”, *IEEE Transactions on Power Apparatus and Systems*, Volume PAS-101, Number 12, 1982, pp. 4625-4631.

[25]C. Chong, R. P. Malhami, “Statistical Synthesis of Physically Based Load Models with Applications to Cold Load Pickup”, *IEEE Transactions on Power Apparatus and Systems*, Volume PAS-103, Number 7, 1984, pp. 1621-1627.

[26]C. W. Gellings, R. W. Taylor, “Electric Load Curve Synthesis - A Computer Simulation of an Electric Utility Load Shape”, *IEEE Transactions on Power Apparatus and Systems,* Volume PAS-100, Number 1, 1981, pp. 60-65.

[27]J. H. Broehl, “An End-Use Approach To Demand Forecasting”, *IEEE Transactions on Power Apparatus and Systems*, Volume PAS-100, Number 6, 1981, pp. 2714-2718.

[28]I. C. Schick, P. B. Usoro, M. F. Ruane, F. C. Schweppe, “Modeling and Weather-Normalisation of Whole-House Metered Data For Residential End-Use Load Shape Estimation”, *IEEE Transactions on Power Systems*, Volume 3, Number 1, 1988, pp. 213-219.

[29]I. C. Schick, P. B. Usoro, M. F. Ruane, J. A. Hausman, “Residential End-Use Load Shape Estimation from Whole-House Metered Data”, *IEEE Transactions on Power Systems*, Volume 3, Number 3, 1988, pp. 986-991.

[30]H. L. Willis, C. L. Brooks, “An Interactive End-Use Electrical Load Model for Microcomputer Implementation”, *IEEE Transactions on Power Apparatus and Systems*, Volume PAS-102, Number 11, 1983, pp. 3693-3700.

[31]H. Müller, “Classification of Daily Load Curves by Cluster Analysis”, *Proceedings of the 8th Power System Computation Conference*, 1990, pp. 381-388.

[32]J. R. Dehdashti, J. R. Tudor, and M. C. Smith, “Forecasting of Hourly Load By Pattern Recognition; A Deterministic Approach”, *IEEE Transactions on Power Apparatus and Systems*, PAS-101, no. 9, September 1982, pp. 3290-3294.

[33]C. S. Özveren, L. Fayall and A. P. Birch, “A Fuzzy Clustering and Classification Technique For Customer Profiling”, *Proceedings of the 32nd University Power Engineering Conference*, 1997, pp. 906-909.

[34]SOLPOS [web page]; http://rredc.nrel.gov/solar/codes_algs/solpos/. [Accessed January 6th, 2000]

[35]R.G. Steadman, “A Universal Scale of Apparent Temperature”, *Journal of Climate and Applied Meteorology*, 23, 1984, pp. 1674-1687.

[36]W. J. Pepi, “The Summer Simmer Index”, *Weatherwise*, Volume 3, 1987, pp. 143-145.

[37]University of Waterloo Weather Station Information [web page]; http://weather.uwaterloo.ca/info.htm#windchill. [Accessed January 10th, 2000]