

Food Bytes: Intelligent Systems in the Food Industry

David Corney¹

*Dept. Computer Science, University College London,
Gower Street, London, WC1E 6BT*

Abstract

Computers have transformed the design of everything from cars to coffee cups. Now the food industry faces the same revolution, with intelligent computer models being used in the design, production and marketing of food products.

The combined market capitalisation of the world's biggest food, cosmetics, tobacco, clothing and consumer electronics companies is \$2 trillion, forming 16% of the world's 500 richest companies (Financial Times Survey 1999). Many of these "fast-moving consumer goods" companies now apply intelligent computer models to the design, production and marketing of their products. Manufacturers aim to develop and produce high-volumes of these commodities with minimum costs, maximum consumer appeal, and of course, maximum profits. Products have limited lifetimes following the fashions of the consumer-driven marketplace. With food and drink, little is known about many of the underlying characteristics and processes: why do some apples taste better than others? How "crunchy" is the perfect apple? Product development and marketing must therefore be rapid, flexible and use raw data alongside existing expert knowledge.

Intelligent systems such as neural networks, fuzzy logic and genetic algorithms, mimic human skills such as the ability to learn from incomplete information, to adapt to changing circumstances, to explain their decisions and to cope with novel situations. These systems are being used to tackle a growing range of problems, from credit card fraud detection and stock market prediction to medical diagnosis and weather forecasting.

This paper introduces intelligent systems and highlights their use in all aspects of the food and drink industry, from ingredient selection, through product design and manufacture, to packaging design and marketing.

Keywords: intelligent systems, neural networks, food design, computers

1 Introduction

The fast-moving consumer goods (FMCG) sector is vast and growing. Intelligent systems, including neural networks, genetic algorithms and fuzzy logic, have been applied to a wide range of fields, from business and finance to medicine and pharmaceuticals (Goonatilake & Khebbal 1995). This paper aims to bring together these two developing areas in a discussion the application of intelligent systems to typical FMCG problems.

The usual definition of FMCG comprises food, drink, tobacco, cosmetics, clothing and consumer electronics. These are items purchased on an everyday basis by a wide range of customers, with the minimum of effort. Because FMCG is such

¹D.Corney@cs.ucl.ac.uk

<i>Inspection</i>	<i>Control</i>
Visual detection of damaged food on a continuous production line. Cameras are cheap, aseptic and reliable.	Regulation of heat-sterilisation treatment. This is made difficult by variations in the food being sterilised.
<i>Design</i>	<i>Sales and Marketing</i>
Discovery of attributes that affect consumer preferences, and how these attributes can be controlled.	Consumer choice modelling. Discovering which factors influence consumers as they make purchase decisions. Also e-commerce.

Table 1: Example food industry applications

a vast industry, this paper concentrates on the food and drink sector alone. The mass-production of food and drink has been increasing for thousands of years. However, the recent globalisation of the marketplace and the concurrent introduction of e-commerce has led to greater competition between producers, forcing them to rethink the entire field. Table 1 shows four broad areas of work in the food industry, which are expanded in the hierarchy shown in Figure 1.

Computers are now used everywhere, from factories to supermarkets, generating more and more data for statistical analysts. This makes the manual investigation of data a daunting task, and suggests the need for some automation of analysis and model building. A common definition of data mining is the transformation of data into useful information (Fayyad, Piatetsky-Shapiro & Smyth 1996). This information may be in the form of a forecasting model, or a model describing relationships between variables, or else a way of summarising the data. A further aim may be data exploration, where a visualisation tool is used interactively to aid understanding of the data.

When choosing a modelling technique, several factors must be considered, including: accuracy, readability, the ease and reliability of parameter discovery, and any pre-processing requirements. There is a tendency in intelligent systems literature to emphasise the accuracy of one technique compared with another, and not without good reason: an inaccurate model is of little use. However, the readability (or transparency) of a model is also important, especially in design work. If a model is comprehensible to an expert in the product field, it should be possible to verify that the model is not unrealistic or likely to make implausible predictions, whether or not it fits the data well. As much as 80% of the work done in many IT organisations is the maintenance and support of existing systems. The more transparent the system is, the easier this task should be. Comprehension by experts will also guide further research work.

Table 2 lists several important characteristics of intelligent systems. Combining these features with the specific requirements of the FMCG field, we can see that an ideal analysis technique would:

- produce human-readable models;

Flexibility
Speed
Ability to learn
Adaptability
Ability to deal with complexity
Explanatory power

Table 2: Features of Intelligent Systems

- cope well with large or small data sets;
- provide a confidence measure for its predictions;
- allow existing, expert knowledge to be included; and
- produce models that generalise accurately to novel data.

No single technique satisfies all these criteria. For example, there tends to be a trade-off between simplicity and accuracy, and between accuracy and the flexibility to solve different problems. Every technique matches different subsets of these factors to differing degrees. One solution to this problem is to develop hybrid intelligent systems, where two or more complementary techniques are combined to produce more accurate and reliable results.

Goonatilake & Khebbal (1995) give several reasons for using hybrid intelligent systems, including technique enhancement and multiplicity of tasks. Technique enhancement recognises that every intelligent system has its own strengths and weaknesses, and by combining two or more, the weaknesses may be overcome and the strengths enhanced. For example, a neuro-fuzzy system combines the learning ability of neural networks (Section 3.1) with the transparency of fuzzy systems (Section 3.2). Also, subtasks within the main problem may be best solved with different techniques. For example, neural networks may be used for pattern recognition within a rule-based expert system.

Comparing the essential features of hybrid intelligent systems with the requirements of food and drink systems, a match can be seen: intelligent systems can provide the flexibility, the learning ability and the explanatory power required by the FMCG industry.

The rest of this paper is organised as follows. The next section describes a range of problems within the FMCG industry, and how they have been solved using intelligent systems. A hierarchy of such problems is included. Section 3 describes these and other intelligent systems, and includes a hierarchy of solutions. The paper concludes in Section 4 with a discussion of how intelligent systems can be used in food design.

2 Case Studies

This section describes several applications of intelligent systems to the design, production and marketing of food and drink commodities. More details of the intelligent systems used are given in Section 3. These case studies are by necessity only an

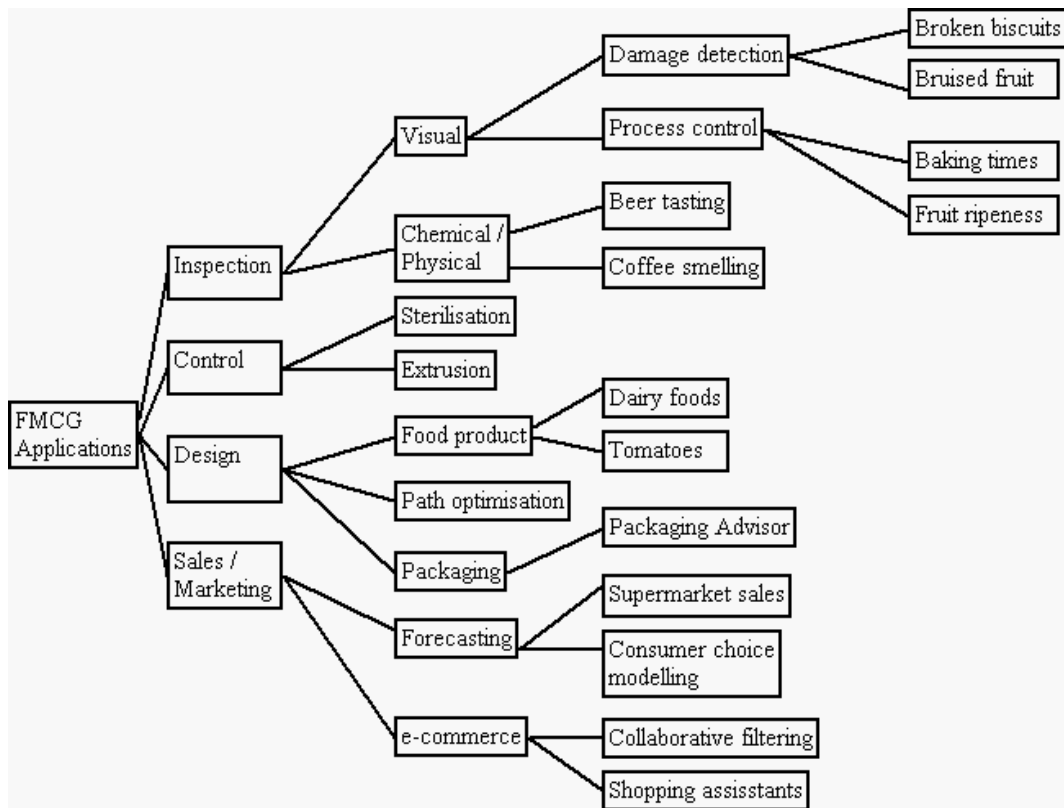


Figure 1: A hierarchy of FMCG applications

overview of the field, yet even these few examples display certain common features and groupings. In order to systematically solve such problems, it is necessary to understand how they relate to each other. Figure 1 shows an attempt to define a hierarchy of FMCG work. If this hierarchy is combined with the intelligent systems hierarchy (Figure 4), then links between problems and solutions become clearer.

2.1 Cheese and Tomato Preference Modelling

Customers' decisions to buy products are based largely on their personal preferences. It seems logical then, to analyse these preferences and manufacture foods that more closely match them. Van Gennert, Wolters & Maarse (1990) describe a generic three-stage plan to develop more appealing, and hence more profitable, food:

1. Discover which food attributes affect consumer preferences and to what degree;
2. Find the chemical and physical properties of the product related to these attributes; and
3. Adjust these properties to develop more appealing products.

However, Van Gennert et al. (1990) report only on the first of these three stages, leaving the others for future work. In common with many food design experiments, two complimentary panels were used. The first was a trained sensory panel, who were

given various tomatoes, and asked to define and use a series of sensory attributes (such as colour, flavour, texture, etc.) to describe each tomato. The second was an untrained preference panel, who simply express a preference for each tomato. Preference panellists tend to suffer from “taste-fatigue” if presented with too many products, so such experiments are very limited.

The sensory and preference data sets were then analysed using principal components analysis (PCA) (Gnanadesikan 1997) to rank the sensory attributes in order of importance when predicting preference. For the whole tomatoes, the first few attributes were size, curvature and soundness, while for the sliced tomatoes, a spicy/sweet flavour and fibrous flesh were most important. This stage of the work is relatively straightforward. Although training a descriptive panel can be expensive, it is a widely used process, as is analysing consumer panel results (Stone & Sidel 1993).

The second and third stages of food design present more of a problem, which is largely unsolved to date. A recent attempt at tackling stage two is outlined by Schonkopf, Naes, Baardseth & Risberg-Ellekjaer (1996), who analysed the design of cheese and other dairy products. They used analysis of variance (ANOVA) to design a series of experiments in which several factors in the production of cheese were varied. Then both ANOVA and PCA were used to analyse these results, with the aim of discovering which factors had the maximum influence on various key attributes of the food, such as taste, texture, juiciness and graininess. This work is shown in the hierarchy (Figure 1) under “design / food product”.

2.2 On-line Beer Tasting

During the fermentation of beer, numerous chemical compounds are formed and broken down, their levels varying over time. In particular, certain diacetyl compounds are found only in mature beer and not in its precursors, so that accurate detection allows the fermenting process to be controlled precisely. Stopping the process too late will waste time and money, as the beer will have to be discarded. Premature stopping may allow the process to be restarted, but will still lead to inefficient production.

Traditional approaches to this type of monitoring and control problem are discussed by Bimbenet & Trystram (1992). They include time-based process control (where each stage is of fixed duration) and off-line monitoring (where samples from the ongoing process are repeatedly removed and analysed, which provides feedback to determine when a process is complete). A third option is on-line monitoring, where sensors are placed within the production line, and must therefore be sterile.

Gardner, Pearce, Friel, Bartlett & Blair (1994) describe a system which monitors the concentration of certain chemicals found in beer during fermentation. An array of chemoresistive sensors was used, each designed to produce distinct responses to a range of chemicals. The outputs from the sensor array were then fed into a multi-layer perceptron, which is a widely used type of neural network (Section 3.1), running on a standard PC. The array was placed in a beer-fermenting tank, and a variety of gases were passed over it. The neural network was then trained to detect trace quantities of diacetyl in the presence of other chemicals, such as ethanol. After training on a number of samples, the system was tested with some novel samples.

Overall, the new system was found to outperform the traditional chemometric fingerprinting. The sensor array/neural network system correctly classified the solutions 93% of the time, whereas the previous system only scored 87%. This allows fine control over the fermentation process, minimising wastage.

2.3 Detecting Damaged Food Products

Although a broken biscuit may taste the same as a whole one, customers are reluctant to buy such damaged goods. It is therefore in the interest of the manufacturer to ensure that the goods leave the plant at as high a standard as possible, even at the cosmetic level. Using people to visually inspect large numbers of items on a production line is very expensive as well as unreliable, due to finite attention spans and limited visual acuity. Non-visual inspection, such as feeling the edge of the product, may damage delicate foodstuffs, as well as introducing bacteria. A non-intrusive camera is, of course, completely aseptic.

Gunasekaran & Ding (1994) describe an automated product inspection system, based on a camera connected to a neural network. The neural network was initially trained to distinguish between whole and broken crackers, and performed very well. However, this task is relatively easy, because all the crackers on the production line were (supposed to be) the same size and shape, so that only a simple template-matching operation was required.

A more challenging task was also described, which involved detecting damaged almonds, again on a continuous production line. Because even undamaged almonds vary considerably in size and shape, the new neural network had to perform a more sophisticated task. Nevertheless, the system still performed well, demonstrating the flexibility of these models. Such systems can detect features invisible to the human eye, and although these features may not be apparent to customers, they may still affect the taste of the product, or its shelf life. Similar systems are used widely to detect flaws in glass, wood veneer, and silicon wafer production (Rosandich 1997).

2.4 Coffee blend classification

Electronic “noses” have been developed to detect and characterise odours. Singh, Hines & Gardner (1996) describe a combination of fuzzy and neural techniques used to classify the odours measured by such a device. They developed a system capable of discriminating accurately between three blends of coffee, and between six types of “smelly” (i.e. tainted) water.

An electronic nose consists of a set of odour sensors, often made from semiconducting polymers. These sensors measure and record as digital patterns fragrances, odours and mixtures of volatiles, based on changes in resistance caused by the chemicals. The sensors are selected to give overlapping responses across the sensor array. One odour constituent may interact with certain individual sensors, but not with others. This selective interaction produces a pattern of resistance changes, known as the “fingerprint.” These fingerprints can then be recognised by a trained neural network. In the work described by Singh et al. (1996), 12 sensor elements were used to successfully distinguish among three coffee blends, and four sensors were used to distinguish between fresh and tainted water samples.

2.5 Consumer choice modelling

Even the best designed and produced commodity will not sell if no customers choose to buy it. It is therefore vital to consider the marketplace throughout the design and production process. Greene & Smith (1987) describe a system which models how and why consumers make purchasing decisions, by using genetic algorithms (Section 3.3) to derive decision rules.

This works by evolving a set of rules describing the decision process of an individual consumer, or a group of similar consumers. Greene & Smith (1987) give the example of deciding whether or not to rent a particular residential property, with rules such as:

IF (Rent < \$400) AND (Commute < 2 miles) AND (Heating included)
THEN (Purchase)

These rules will initially be random, and therefore very poor, but the genetic algorithm will gradually improve their quality, through natural selection, taking into account the individual consumer's historic purchase decisions as well as the current situation. The same modelling process could be used to model food purchasing decisions, which could produce rules such as:

If (Price < £1.50) AND (Icing Colour is White) THEN (Purchase Cake).

Figure 1 shows this work under "Sales/Forecasting".

2.6 Other Studies

Many foods are heat-treated to kill bacteria, but the varying attributes of the food make controlling the temperature of sterilisation units very difficult. Singh & Ou-Yang (1994) describe a process plant where a simple PID (proportional-integrative-derivative) controller had been used to control the sterilisation unit. They developed a fuzzy logic controller that out-performed the PID controller. The fuzzy rules (Section 3.2) used were derived from the (human) operator's intuition, with membership functions derived by trial and error. A typical fuzzy rule used by the system was: "If the holding tube inlet temperature is slightly above the set point temperature and the holding tube inlet temperature is rising slowly then slightly close the steam valve." Fuzzy logic (unlike the inherently linear PIDs) can deal successfully with complex, non-linear systems. The fuzzy controller consisted of a relatively small number of simple rules, and this simplicity led to faster response times. Zhang & Litchfield (1991) provide further discussion of fuzzy control in FMCG production.

Food packaging should be cheap, lightweight and robust, to aid mass production and transportation. Packaging must also provide a barrier to oxygen, to prevent the food from spoiling. Rigid plastic packaging consists of multiple layers of resin, some providing structure and others providing the oxygen barrier. Each resin has different characteristics, constraining the combinations allowable. For example, if the melting point of one resin is above the vaporisation point of another, then they cannot be used together. Topolski & Reece (1989) developed an expert system to aid the design of rigid plastic food containers for DuPont. Their "Packaging Advisor" system consists of a user-friendly front-end, a rule set, and a database of

resin characteristics. The user specifies various requirements of the packaging being designed, such as desired shelf life, optical properties and usage temperature limits. The system then uses its internal rule set to reformulate these requirements in terms of necessary resin properties. A search of the resin database is then performed, and a list of various suitable resin designs is presented to the designer.

A key aim of forecasting food sales by supermarkets is to predict trends in customers' purchases, which then guides stock control, marketing decisions, staffing levels, etc. Thiesing & Vornberger (1997) describe a study to analyse and predict weekly sales of a range of products sold in a German supermarket. They used a standard multi-layer perceptron neural network, with inputs such as the previous few weeks' sales levels, national holidays during the week, product promotions and price changes. The predictions made by the network were better than several alternative models, such as moving averages or a static model. One advantage of neural networks over conventional statistics is their ability to cope with chaotic time series predictions. Conventional time-series models, such as ARMA and ARIMA (autoregressive (integrated) moving average), are effective with non-chaotic series, but fail on the more complex, non-stationary series that characterise sales levels.

3 Intelligent Systems

This section describes in more detail the intelligent systems techniques used in the work described in Section 2, and other systems not yet widely used in the FMCG industry. In each case, a brief description of the technique is followed by some advantages and disadvantages. This is not meant to be an exhaustive list of techniques, nor is the discussion of each one in any sense complete. Nonetheless, by showing some of the similarities and differences, and the strengths and weaknesses of each technique, we can come to a better understanding of how they may be applied. Figure 4 shows a hierarchy of these (and other) techniques; a further discussion of the hierarchy is given in Section 3.9.

3.1 Neural Networks

Artificial neural networks (ANNs) are statistical models based on an extremely simplified model of the brain, and consist of networks of nodes with weighted connections. In common with many intelligent systems approaches, ANNs must initially be trained, using a subset of the available data to optimise the free parameters (e.g. the connection weights). Two of the most common types of neural network are multi-layer perceptrons (MLPs) and radial basis function (RBF) networks, which use networks of sigmoid and Gaussian functions respectively. RBF networks have the advantage of faster optimisation techniques, such as combining clustering and linear optimisation, rather than the MLPs often inefficient back-propagation. See Bishop (1995) for more details.

Advantages: ANNs have been widely used in industry as well as academia for many years, and are well understood. They are flexible enough to model any system accurately (in theory, if given enough suitable data), and cope well with data that is noisy, non-linear and high dimensional. There is a variety of fast heuristic training algorithms available.

Disadvantages: ANNs are essentially black boxes, as a trained network is represented as just a (often large) matrix of parameters. There is also a strong risk of over-fitting, where the system models noise in the data, rather than just the underlying function as desired. This can be overcome to some extent by regularisation (which penalises complexity) or by applying Bayesian techniques (Section 3.5). ANNs generally need large data sets to learn the parameters correctly. Because the parameter search is a point-to-point iterative optimisation, there is also a risk of becoming trapped in local optima, and therefore failing to find the globally optimal model. There are no reliable rules for determining the ideal number of nodes in the network; this is often estimated after laborious cross-validation.

3.2 Fuzzy Logic

Fuzzy logic captures the human experience of making complex decisions based on uncertain or incomplete information, and produces exact results from imprecise data. In conventional set theory, sets are “crisp”, which means that for any given set, every element is either in the set or not. While this simplifies notions of set union and intersection, it does not correspond with our experience of the real world. For example, there is no hard division between the set of foods that taste good and the set of those that do not, although undoubtedly some food is good, and some food is not. (This is irrespective of personal preferences: even for a single consumer, there are no crisp divisions.) Fuzzy logic is based on the notion of fuzzy set membership, where the degree to which an element belongs to a set ranges from zero to one, as defined by the set’s membership function. Because the sets overlap, each item can belong to two (or more) sets simultaneously.

Once the fuzzy sets have been defined for each variable, rules are defined that link fuzzy sets together. Usually, an initial set of rules is created by an expert because of the difficulty of learning fuzzy rules from data. Several hybrid fuzzy systems have been developed to perform automatic knowledge acquisition, such as neuro-fuzzy approaches (Nauck 1994) and inductive learning of fuzzy rules (Kacprzyk & Iwanski 1992).

Advantages: Because fuzzy rules are very close in format to experts’ knowledge, representing this knowledge is relatively straightforward. Fuzzy systems also tend to produce fewer rules, and therefore simpler systems, than other rule-based systems. This simplicity means that they are easier to understand and maintain, and often generalise well. As with other rule-based systems, several rules may be triggered at once, leading to robust predictions.

Disadvantages: One of the biggest drawbacks to fuzzy systems, as with conventional expert systems, is that the rules must be created by hand. This introduces an information bottleneck, although this can be eased by the hybrid systems mentioned above. In multi-dimensional problems, the total number of fuzzy sets becomes very large, so that large data sets may be required to discover the rules.

3.3 Genetic Classifiers

Genetic classifiers are an extension of genetic algorithms, and both are inspired by Darwinian notions of evolution through inheritance, variation and survival of the

fittest. Genetic classifiers work by creating a population of sets of simple random production rules in the form: IF <condition> THEN <action>. These sets of rules undergo evolution, so that “useful” (i.e. accurate) rules tend to be developed. Pairs of rules are selected as parents, and used to produce offspring that are similar to, but not identical to the parents. The likelihood of becoming a parent (and therefore of genetically surviving), depends on the relative accuracy of the rules.

Each rule has a weight associated with it, so that if the conditions of more than one rule are met, selection between them can be made: the one with the strongest weight will be activated. The resultant rule sets will have a similar form to a decision tree (Section 3.6) or propositional rule set (e.g. CN2 rule induction (Clark & Boswell 1991)), but because they are derived by a different technique, they will tend to cope with noise in a different way, producing distinct results. See Goldberg (1989) for an introduction.

Advantages: Genetic classifiers carry out a population-based search, like all evolutionary systems, reducing the risk of getting stuck in poor local optima. They are faster than some alternatives because they do not attempt an exhaustive search of the hypothesis space, and unlike decision trees, they can represent complex, multivariate conditions straightforwardly. Genetic classifiers produce human-readable results.

Disadvantages: Genetic classifiers work best on discrete-variable problems. The evolutionary process includes a significant random element, so that multiple runs of the algorithm may be required to produce reliable results. The underlying theory of exactly how genetic algorithms and genetic classifiers find solutions, is still incomplete (Thornton 1997).

3.4 Support Vector Machines

Support vector machines (SVMs) (Burgess 1998) are a novel class of learning algorithms that have been applied to a range of problems including 3D object recognition (Roobaert & Hulle 1999) and customer modelling (Wu & Auslender 1998). This latter example builds decision trees (Section 3.6) using SVMs, to distinguish between “good” and “bad” customers, such as late payers.

Conventional modelling techniques (including those in described in Sections 3.1–3.3) perform empirical risk minimisation: they attempt to find a set of parameters that minimises the error observed on the training data. In contrast to this, SVMs approximate structural risk minimisation, which minimises a bound on the expected error on unseen data. This is related to regularisation techniques, which penalise complexity.

Consider the linearly-separable two-class classification problem shown in Figure 2: the aim is to discover a hyperplane that separates the two classes (dots and crosses in the figure). The SVM finds a subset of the data known as the support vectors (ringed in Figure 2). These are the points immediately adjacent to the separating hyperplane, i.e. those points on the edge of each class. These points (and their parameters) uniquely determine a hyperplane which optimally separates the classes, in the sense of finding a hyperplane that is as distant as possible from the data on either side. This distance is known as the margin, and hence SVMs are large-margin classifiers, as are boosting and bagging (Section 3.7). For a given em-

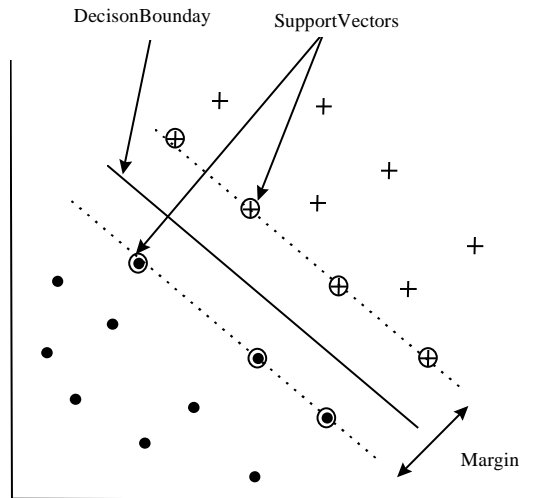


Figure 2: A support vector classifier. The decision boundary separates the two classes (dots and crosses) while maximising the margin.

pirical error rate, a larger margin improves generalisation to novel cases. In contrast, a neural network uses its weights to define a similar hyperplane, and the learning algorithms used move this hyperplane until the training error is zero, i.e. until the classes are separated completely. This may touch some data points, and so will not be an optimal decision plane.

In more complicated, realistic problems, the data may not be linearly separable. In these cases, the data are mapped into a higher dimensional space using kernel functions, such as the radial basis functions used in RBF networks (Section 3.1). In the higher-dimensional space, the classes become linearly separable. Slack variables may also be introduced to allow points to remain on the “wrong” side of the decision boundary, to allow a relatively simple model to fit noisy data.

Advantages: Numerous studies have found that SVMs tend to generalise better than comparable classifiers (Schölkopf, Sung, Burges, Girosi, Niyogi, Poggio & Vapnik 1997). The support vectors themselves define the smallest subset of the data that still completely defines the optimal solution, so can be used as a data summary. The parameters are found by solving a convex linear programming problem, which has a single global solution. This compares favourably with iterative optimisation techniques (such as typical neural networks learning algorithms), which often only find local optima.

Disadvantages: As in the case of neural networks, the results are often hard to interpret: a linear solution in high dimensional space is very non-linear in low dimensional space. Results are sensitive to parameter selection, which must be performed by the analyst before use, such as choosing the nature of the kernel.

3.5 Bayesian Learning

Bayesian learning has been applied to a wide range of problems, from robot control to fraud detection. Bayesian belief networks are graphical models that encode

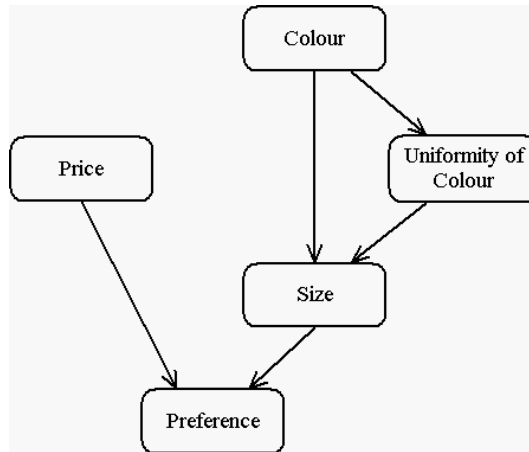


Figure 3: A Bayesian belief network

probabilistic relationships between variables of interest. They have become increasingly popular within the intelligent systems community since their inception in the late 1980s (Pearl 1988), due to their ability to represent and reason with uncertain knowledge. Figure 3 is a typical Bayesian belief network, showing how certain attributes combine to influence the consumers’ preferences. A recent paper by the author (Corney 2000) discusses Bayesian belief networks in some detail, and describes their application to a typical food design problem. The paper shows how the design profile of a hypothetical “ideal” product can be derived from limited data.

Bayesian pattern recognition is a mathematically rigorous technique used to discover the free parameters of a statistical model given the data. Non-Bayesian machine learning techniques have an unfortunate tendency to over-fit the data, leading to poor performance when presented with new, unseen cases. Rather than providing only a single “best estimate” of the parameter values, Bayesian techniques produce the probability distribution of each parameter, thus producing a confidence measure as well as the parameter itself.

Many intelligent systems (including those discussed in Sections 3.1–3.4) will always produce an output, with no indication of confidence. If a model is trained using data drawn from one particular space, and then used to make predictions about data from a different space, the output will be very unreliable. Essentially, the higher the density of the training data in a particular region, the more reliable the model will be in that region. This reliability is captured by the probability distributions produced by Bayesian models.

Advantages: Bayesian techniques inherently include regularisation (to prevent over-fitting the data), give confidence values, and can form committees, all of which have to be added as extras to non-Bayesian methods. Bayesian methods also allow the explicit inclusion of prior knowledge. All available data can be used to build the model, so no cross-validation is required. The underlying theory is based in a well-established field of mathematics.

Disadvantages: Bayesian models are mathematically complicated, as they produce distributions, rather than single values for the parameters. This means that modelling is slower, and that the final results may be harder to understand.

3.6 Decision Trees

Decision trees are classification tools that sort instances of data down a tree from the root to a leaf, with the terminal leaf specifying the classification of the instance. Every internal (non-leaf) node in the tree tests the instances against a particular attribute, with each branch descending from the node representing one possible value of this attribute. Each path from root to leaf is a conjunction of attribute tests, so the entire tree is a disjunction of conjunctions. This means that the tree can be re-written as a set of discrete rules, making it easy to understand. Of the various algorithms for building trees, one of the more widely used is ID3 (Quinlan 1986). This iteratively selects the attribute that best separates the classes, and uses this to split the data at each node. This algorithm is a precursor to Quinlan's proprietary C5.0 algorithm, which introduces pruning and boosting (Section 3.7) to improve generalisation performance (Quinlan 1993).

Advantages: Decision trees are clear models, in the sense that it is easy for a non-expert to understand what a tree is showing (provided it is not too large). Decision trees are widely used and well understood, and are reasonably robust to noisy and incomplete data.

Disadvantages: The basic algorithm only models discrete-valued functions, so continuous data has to be divided into bins or clusters first. This division loses information from the data, because the scales the data are measured on are transformed from interval (or possibly ratio) scales into ranked ordinal scales. The rules captured by the tree divide the data space into (hyper-)rectangles, which may be an unjustified bias.

3.7 Combining Classifiers

It is always easier to build a poor model than a strong one. Fortunately, this can be put to good use by combining a number of weak learners into a committee. A committee tends to outperform the average of its constituent members if the errors of the members are sufficiently uncorrelated (Perrone & Cooper 1993), which will be true if they have been trained using different (possibly overlapping) sets of training data. Bishop (1995) describes forming committees of neural networks.

Freund & Schapire (1996) describe a boosting algorithm, which creates a series of weak learners and combines them to form a committee. Each learner is trained on the entire training data set, and each point in the set is assigned a weight. During learning, data points that tend to be modelled inaccurately are given increasingly large weights, which forces future learners to model them more accurately. Thus when the committee of learners is formed, all of the data points should be modelled accurately by at least one weak learner, and therefore the committee's final vote should be accurate over the entire problem space.

Quinlan (1996) compares this boosting technique with bagging, where new sets of the training data are repeatedly created by selecting points (with replacement) from the main training set. This tends to produce distinct data sets, and hence different models, which between them cover the entire problem space.

Advantages: These techniques are simple to implement, and results show that they can produce considerable improvements in accuracy. One known problem with

modelling consumer preferences is that models tend to reflect the negative dislikes of certain products more accurately than they model the positive likes of other products. A version of boosting could be used to overcome this by biasing the models towards or away from certain “problem” data points.

Disadvantages: Each committee member has to be trained separately, so these techniques can be slow. Perhaps counter-intuitively, boosting and related techniques work best when the basic algorithm is weak, such as a decision tree. Boosting very sophisticated models tends to produce marginal improvements at best.

3.8 Cluster Analysis

Cluster analysis aims to discover distinct groups of similar points within a set of data, with applications ranging from biological taxonomy to market research. Green, Frank & Robinson (1967) describe the division of American cities into groups based on various socio-economic indicators, to select suitable sites for test marketing. A related application is consumer preference modelling, where the aim is to find groups of consumers with similar likes and dislikes. One form of this is collaborative filtering (Ungar & Forster 1998), which is increasingly popular in e-commerce. Consumers are grouped together based on past purchases, so new purchase suggestions can then be made. Wooding (1999) describes using self-organising maps as a visualisation tool to investigate properties of food and their impact on consumer preferences.

The techniques described earlier are examples of supervised learning, where the aim is to learn some function given labelled instances of data. Clustering is an example of unsupervised learning, where the data set is unlabelled (i.e. each record does not have a known target class or value). Such data can be viewed as having been generated by a series of clusters, but where the information about which cluster generated which data point is missing. The aim of cluster analysis is to rediscover this missing information, and therefore discover where the clusters are, how big they are, what shape they are, and so on. Everitt (1993) provides a comprehensive introduction to the field.

Advantages: Rather than treating all consumers the same, if distinct groups of consumers with similar preferences can be found, then different products can be manufactured and marketed for each group. These will match consumer preferences more closely.

Disadvantages: One of the major open problems in the field is in determining the number of clusters that exist within a set of data. Many heuristics have been suggested, but no rule works reliably. There is a very wide range of clustering techniques, each with different properties, and no clear way to select an appropriate method.

3.9 A Machine Learning Hierarchy

Sections 3.1–3.8 form an overview of some of the many intelligent systems techniques that are available, each with its own strengths and weaknesses. This leaves open the question of which method should be used to solve each particular problem. Given that a data-mining problem is defined primarily by the data, it seems sensible to choose a technique based on the features of the data at hand. Figure 4 is an

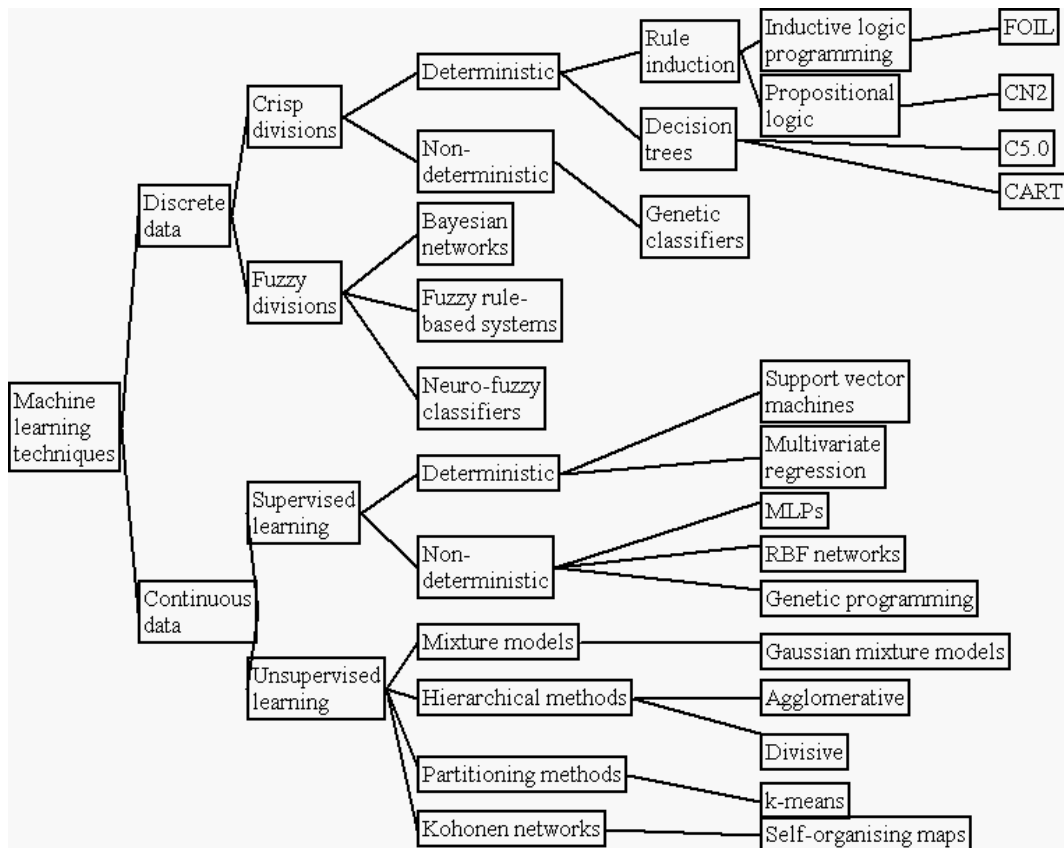


Figure 4: A hierarchy of intelligent systems techniques

attempt to show some of the relationships between these techniques, based on the characteristics of the data (as well as characteristics of the solution).

There are (at least) two caveats regarding this particular hierarchical classification. Firstly, it is incomplete: it shows less than 20 techniques, missing out case-based reasoning, factor analysis, and linear discriminants, to name but three. Even the techniques it does show often represent a group of related methods, such as the numerous genetic classifier algorithms, or the limitless regression models.

Secondly, the same list of techniques could have been grouped into many equally justifiable hierarchies, suggesting quite different relationships between them. This hierarchy must be enhanced and expanded, as must the FMCG hierarchy introduced earlier (Figure 1). Ultimately, it is hoped, these two trees can be merged into a single chart, showing how each problem can best be solved with particular machine learning techniques.

Some terms used in the diagram need to be clarified:

Discrete vs. continuous If a variable is discrete, it can only take on values drawn from a finite set. A continuous variable can take on any value, usually within a finite range. Several techniques outlined above require discrete data, such as decision trees and rule induction. Susmaga (1997) describes one technique for breaking continuous variables into discrete sets.

Crisp vs. Fuzzy A crisp division has a clear, well-defined boundary, whereas a fuzzy division has overlaps between each division. The gender of a consumer would be defined using crisp divisions, but typical food data sets consist of fuzzy divisions. For example, there is no clear boundary between good and bad food. This dichotomy only applies to discrete data.

Deterministic vs. non-deterministic Certain modelling techniques will always produce the same model, given a fixed set of data, whereas others rely on initially random parameters or on random processes, and so will tend to produce different models each time they are executed. Examples of the former are support vector machines and least-squares regression; examples of the latter are neural networks and genetic classifiers.

Supervised vs. unsupervised If data points have a specific output target, then supervised learning can be used to predict the target value from the input variables. Most of the listed techniques require targets, and perform supervised learning only. If there is no known, specified target, then unsupervised learning, such as clustering, can still be carried out.

4 The Future: Food Design

Returning to the case studies of applications of intelligent data analysis to FMCG problems (Section 2), we can see more clearly how they relate to the central problem of food design:

Cheese and Tomato Preference Modelling This is obviously closely related to the general problem of food design. Although this is a statistical method, rather than an intelligent system, the use of PCA and ANOVA at least provides a benchmark for evaluating alternative techniques.

On-line Beer Tasting In order to predict consumer preferences from product taste, one must have a way of measuring that taste reliably. Enhancing or replacing the expensive sensory panels with cheaper instruments applies equally to taste and to appearance.

Detecting Damaged Food Again, using instrumental image data is cheaper (and potentially more accurate) than sensory panels, and detecting damage can be seen as analogous to detecting low product quality, with quality defined by consumer preferences.

Coffee-blend Classification Using instruments to measure odours is applicable to many foodstuffs. Preferences for many foods, such as wine, cheese and fruit, are based largely on odour.

Other Studies The other three applications described (food sterilisation, package design and food sales forecasting) move beyond product design, but are firmly within the same industrial area.

Following from Corney (2000), related work is continuing to combine market segmentation and product design. One approach being considered is combining k -means clustering (to group consumers) with Bayesian belief networks (to model each group's preferences).

This paper has discussed current and future applications of intelligent systems to a wide range of food and drink issues, and has highlighted food design as a fruitful area of further work. Proposals have been made for hierarchical orderings of both problems and solutions, with the hope of combining such hierarchies to match existing problems with appropriate techniques.

Acknowledgements

The research was undertaken within the Postgraduate Training Partnership established between Sira Ltd and University College London. PTPs are a joint initiative of the DTI and the EPSRC. The work has been generously sponsored by Unilever Research plc.

References

- Bimbenet, J. & Trystram, G. (1992), 'Process control in the food industry', *Food and Bioproducts Processing* **70 C3**, 115–125.
- Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Clarendon-Press, Oxford.
- Burges, C. J. C. (1998), 'A tutorial on support vector machines for pattern recognition', *Data Mining and Knowledge Discovery* **2(2)**, 121–167.
- Clark, P. & Boswell, R. (1991), Rule induction with CN2: Some recent improvements, in 'Proceedings of the Fifth European Working Session on Learning', Springer, Berlin, pp. 151–163.
- Corney, D. (2000), Designing food with Bayesian belief networks, in I. Parmee, ed., 'Evolutionary Design and Manufacture – ACDM2000', Springer-Verlag, pp. 83–94.
- Everitt, B. S. (1993), *Cluster Analysis*, 3rd edn, Edward Arnold.
- Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996), From data mining to knowledge discovery: An overview, in U. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy, eds, 'Advances in Knowledge Discovery and Data Mining', AAAI Press/MIT Press, pp. 1–34.
- Financial Times Survey (1999), *Financial Times*. London.
- Freund, Y. & Schapire, R. E. (1996), Experiments with a new boosting algorithm, in 'Proceedings of the 13th International Conference on Machine Learning', Morgan Kaufmann, pp. 148–146.
- Gardner, J., Pearce, T., Friel, S., Bartlett, P. & Blair, N. (1994), 'A multi-sensor system for beer-flavour monitoring using an array of conducting polymers and predictive classifiers', *Sensors and Activators series B* **18-19**, 240–243.

- Gnanadesikan, R. (1997), *Methods for Statistical Analysis of Multivariate Observations*, 2nd edn, Wiley.
- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Massachusetts.
- Goonatilake, S. & Khebbal, S. (1995), *Intelligent Hybrid Systems*, Wiley and Sons.
- Green, P., Frank, R. & Robinson, P. (1967), ‘Cluster analysis in test market selection’, *Management Science* **13**, 387–400.
- Greene, D. P. & Smith, S. F. (1987), A genetic system for learning models of consumer choice, in J. J. Grefenstette, ed., ‘Proceedings of the 2nd International Conference on Genetic Algorithms and their Applications’, Lawrence Erlbaum Associates, Cambridge, MA, pp. 217–223.
- Gunasekaran, S. & Ding, K. (1994), ‘Using computer vision for food quality evaluation’, *Food Technology* pp. 151–154.
- Kacprzyk, J. & Iwanski, C. (1992), Fuzzy logic with linguistic qualifiers in inductive learning, in L. Zadeh & J. Kacprzyk, eds, ‘Fuzzy Logic for the Management of Uncertainty’, John Wiley and Sons.
- Nauck, D. (1994), A fuzzy perceptron as a generic model for neuro-fuzzy approaches, in ‘Proceedings Fuzzy-Systeme’94’, Munich.
- Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, California.
- Perrone, M. P. & Cooper, L. N. (1993), When networks disagree: Ensemble methods for hybrid neural networks, in R. J. Mammone, ed., ‘Neural Networks for Speech and Image Processing’, Chapman-Hall, pp. 126–142.
- Quinlan, J. (1986), ‘Induction of decision trees’, *Machine Learning* **1**, 81–106.
- Quinlan, J. R. (1993), *C4.5: Programs for machine learning*, Morgan Kaufmann.
- Quinlan, J. R. (1996), Bagging, boosting, and C4. 5, in ‘Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference’, AAAI Press / MIT Press, Menlo Park, pp. 725–730.
- Roobaert, D. & Hulle, M. V. (1999), ‘View-based 3d object recognition with support vector machines’, *IEEE Neural Networks for Signal Processing Workshop*.
- Rosandich, R. (1997), *Intelligent visual inspection : using artificial neural networks*, Chapman and Hall.
- Schölkopf, B., Sung, K.-K., Burges, C. J. C., Girosi, F., Niyogi, P., Poggio, T. & Vapnik, V. (1997), ‘Comparing support vector machines with Gaussian kernels to radial basis function classifiers’, *IEEE Transactions on Signal Processing* **45**(11), 2758–2765.

- Schonkopf, S., Naes, T., Baardseth, P. & Risberg-Ellekjaer, M. (1996), 'Computer-aided product development in the food industry', *Food Technology* pp. 69–75.
- Singh, R. & Ou-Yang, F. (1994), 'Knowledge-based fuzzy control of aseptic processing', *Food Technology* pp. 155–162.
- Singh, S., Hines, E. & Gardner, J. (1996), 'Fuzzy neural computing of coffee and tainted-water data from an electronic nose', *Sensors and Actuators B* **30**, 185–190.
- Stone, H. & Sidel, J. L. (1993), *Sensory Evaluation Practices*, 2nd edn, Academic Press.
- Thiesing, F. & Vornberger, O. (1997), Forecasting sales using neural networks, *in* B. Reusch, ed., 'Computational Intelligence: Theory and Applications - Proceedings of Fifth Fuzzy Days International Conference', Springer-Verlag, Dortmund, Germany, pp. 321–328.
- Thornton, C. (1997), 'The building block fallacy', *Complexity International*.
- Topolski, A. S. & Reece, D. K. (1989), Packaging Advisor TM: an expert system for rigid plastic food package design, *in* H. Schorr & A. Rappaport, eds, 'Proceedings of the IAAI-89 Conference on Innovative Applications of Artificial Intelligence', MIT Press, pp. 348–357.
- Ungar, L. & Forster, D. (1998), Clustering methods for collaborative filtering, *in* 'AAAI Workshop on Recommender Systems'.
- Van Gennert, L., Wolters, C. & Maarse, H. (1990), The relationships between sensory attributes of round and beef tomatoes, and consumer preferences, *in* Y. Bessiere & A. Thomas, eds, 'Flavour Science and Technology', J Wiley and Sons Inc.
- Wooding, S. (1999), On the application of self-organising maps to the exploration of product performance measures, *in* R. John & R. Birkenhead, eds, 'Soft Computing Techniques and Applications', Physica-Verlag, pp. 40–45.
- Wu, K. B. D. & Auslender, L. (1998), On support vector decision trees for database marketing, Technical Report 98–100, Rensselaer Polytechnic Institute, Troy, New York.
- Zhang, Q. & Litchfield, J. (1991), 'Applying fuzzy mathematics to product development and comparison', *Food Technology* **45**(7), 108–1152.