

Designing Food with Bayesian Belief Networks

David Corney

Computer Science Dept., University College London
Gower Street, London, WC1E 6BT
D.Corney@cs.ucl.ac.uk

Sira Technology Centre
South Hill, Chislehurst, Kent, BR7 5EH
David.Corney@ptp.sira.co.uk

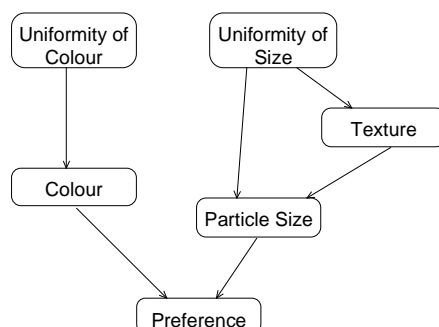
Abstract. The food industry is highly competitive, and in order to survive, manufacturers must constantly innovate and match the ever changing tastes of consumers. A recent survey [1] found that 90% of the 13,000 new food products launched each year in the US fail within one year. Food companies are therefore changing the way new products are developed and launched, and this includes the use of intelligent computer systems. This paper provides an overview of one particular technique, namely Bayesian Belief networks, and its application to a typical food design problem. The characteristics of an "ideal" product are derived from a small data set.

1. Introduction

Bayesian Belief Networks are graphical models that encode probabilistic relationships between variables of interest. They have become increasingly popular within the AI community since their inception in the late 1980's [2] [3], due to their ability to represent and reason with uncertain knowledge. They have been used successfully in expert systems, decision support systems and diagnostic systems, among others. Figure 1 shows a typical network, described in more detail in part 3.

Historically, one of the first applications of Bayesian networks was to medical diagnosis. For example, a Bayesian network system has been developed from a database containing descriptions of many symptoms and associated diseases [4]. By entering a brief description of a patient's symptoms, the system can deduce likely causes, i.e. diseases. The system was designed as a decision-support system for use by medical experts, and as a teaching aid.

Bill Gates recently described Microsoft's competitive advantage as being its expertise in Bayesian networks [5]. Microsoft have been actively recruiting experts in the field since the early 1990's, and have become a significant research force. They have also released software components using Bayesian networks, such as the Office Assistant and the grammar checker, both in Office 97. Further applications of



$p(\text{Size} \text{Texture}, \text{Uniformity of Size})$		
	$T=\text{low}$	$T=\text{high}$
$US=\text{low}$	(0.6, 0.4)	(0.7, 0.3)
$US=\text{high}$	(0.9, 0.1)	(0.8, 0.2)

Figure 1: A hypothetical Bayesian network

Table 1: A conditional probability table

Bayesian networks include robot guidance [6], software reliability assessment [7], data compression [8] and fraud detection [9].

A great deal has been achieved with Bayesian networks, and (the author believes) they can and will be applied to product design. Products are artefacts purchased and used because of their properties and functions [10]. They are designed to meet the end-users' requirements, whether this means a car must be fast, a mouse-trap must be "better", or a plate of food must taste nice. Because of their ability to learn, adapt and explain, intelligent systems such as Bayesian networks can aid product designers in their work.

The next section describes the nature of the data used in food design work. This is followed by an overview of Bayesian networks, and descriptions of how the models can be built and used. Finally, some experimental results are presented.

2. Food Data

When designing new food products, companies typically obtain data from three sources: sensory panels, preference panels, and instrumental data. The nature of instrumental data is product-specific, and is not covered here in detail, but may include digital images, acoustic imaging or chemical fingerprinting.

The *sensory panel* is a group of typically 10-20 people, selected and trained for several months. The panel derives their own descriptors of product attributes, which can then be systematically used to describe different varieties of the product. The panel typically produces between 8 and 20 descriptors after discussion and analysis. Members of the panel are then presented with a variety of different products, selected to represent a wide range of flavours, colours, etc. They then measure each sample by ranking it for each descriptor. The ideal sensory panel should produce absolutely consistent and uniform results, allowing the panel to be treated as an instrument. In practice, human perception is neither absolute nor constant.

The *preference panel* is a larger group of untrained people, typically 50-500 potential consumers, who are bought in "off the street" specifically for the trials. They are individually presented with a few samples and are then asked to rank each one on a simple preference scale. No training is given and no discussion between panellists is allowed, so the results will be entirely subjective, and vary from panellist to panellist. The relatively large panel size should smooth out any unwanted discrepancies.

Once the preference panel has classified the samples, the sensory panel data is re-examined, to determine which sensory attributes best distinguish the different preferences. For example, suppose the preference panel gave two samples significantly different grades. Then if the sensory panel gave both of them the same grade for some measure, e.g. shape, then this attribute is a poor predictor of quality. If a correlation can be found between the one or more of the sensory panel attributes and the preference panel scores, then this can be used to guide future product design and marketing.

The entire data-gathering process is very expensive and very time-consuming, and depends on human perception, which lead to the most striking and important features of the data sets: they are small and sparse, and contain uncertainty. The data used here is described in section 5 and included as an appendix.

3. Features of Bayesian Networks

Bayesian Belief Networks are graphical representations of the joint probability distributions over a set of discrete variables, and incorporate conditional independence assumptions. They consist of a directed acyclic graph (DAG) such as the simple model shown in Figure 1, and a set of conditional probability tables, such as Table 1. In the graph, each node represents a variable and the arcs between nodes specify the independence assumptions between the variables. More precisely, each variable is "conditionally independent of any combination of its non-descendants, given its parents" [8]. Thus Figure 1 shows, for instance, that given "colour", then "uniformity of colour" has no influence over any variable.

One conditional probability table is determined for each node, defining the probability of the variable being in each possible state, given each of the possible states of its parent node(s). If a node has no parents, the unconditional probabilities are used instead.

Table 1 shows the conditional probability distribution for the "particle size" node, conditional upon its parents, namely "uniformity of size" (US) and "texture" (T). Each cell in the table has two numbers, the probability that the particle size is low (i.e. "small") and the probability that it is high (i.e. "large").

Bayesian networks have a number of features that make them suitable for product design, as shown in Table 2 and discussed in the remainder of this section.

3.1 Explaining away observations

"Explaining away" can be defined as "a change in the belief in a possible explanation if an alternative explanation is actually observed" [11]. The standard example of explaining away is the lawn sprinkler: suppose we observe that the lawn is wet one morning. There are two possible causes: either it rained or the sprinkler was left on. Our belief in both of these explanations increases. We then observe that our neighbour's lawn is also wet, and so deduce that it rained last night. Because we now believe that the wet lawn was caused by the rain, we no longer have any reason to believe that the sprinkler was left on, so we should retract that belief [12].

In the case of food modelling, if we know that sweet foods are generally preferred, and we have a particular sample that is both sweet and popular, then our simple model gives us no reason to believe its colour will affect its popularity. More traditional rule-based expert systems fail to cope with this type of situation, because the systems are *modular*, meaning that the rules are fired with no reference to the

Explaining away	Make effective use of all available information
Bi-directional Inference	Can diagnose what causes high preference
Complexity	Can scale up to represent complex models
Uncertainty	Can deal with uncertainty in the data
Confidence values	Provide confidence measures on results
Readability	Produce graphical, transparent models
Prior Knowledge	Can incorporate expert knowledge

Table 2: Features of Bayesian Networks

context of other rules or the source of the data. The conditional probabilities in the Bayesian network models encapsulate the desired effect.

3.2 Bi-directional Inference

Many intelligent systems (e.g. feedforward neural networks, fuzzy logic) are strictly one-way in the sense that when a model is given a set of inputs it can predict the output, but not vice versa. The question one really wants to ask "What features would a product have, if it had a high preference score?" This inverse problem can be solved by bi-directional modelling, where inputs can be used to predict outputs, and outputs can be used to "predict" or diagnose inputs.

Bayesian networks can do this within a single structure because variables are *not* specified as being solely for input or for output. By applying Bayes' theorem, the direction of the relationship can be reversed. For example, given the rule "*If (product is sweet) then (product is preferred)*" and given the fact "*product is sweet*" we can obviously deduce that it is preferred. However, with a Bayesian system, we might observe (or hypothesise) that "*product is preferred*" and deduce that this preference must be caused by its sweetness, i.e. that "*product is sweet*". In other words, while many systems can perform induction, Bayesian networks can also perform *abduction*.

3.3 Complexity

The independence assumptions expressed by the graph mean that fewer parameters need to be estimated because the probability distribution for each variable depends only on the node's parents. This independence assumption allows us to factorise the network, considering each node and its parents in isolation from the rest of the model. This means that far fewer parameters are needed to fully specify the relationships between the variables, than would be required by a fully connected network, or any other global, "unfactorable" model.

Similarly, when learning the structure of the graphs, the search can be local, with the optimal set of parents for each node being selected independently of the rest of the model. Thus even very complex models can be discovered without suffering from a combinatorial explosion. These efficiencies are particularly important when only small data sets are available, as is often the case with food design. The K2 algorithm described later relies on this feature.

3.4 Uncertainty

There are many sources of uncertainty, such as distortion, incompleteness and irrelevancy [11]. Consider asking a group of preference panellists, "How much do each of you like product X?" However much time is spent defining or describing the word "like", there is no guarantee that any two subjects will actually use the same scale to measure the product on, irrespective of personal differences in taste. Furthermore, experimental results show that individual subjects will give the same product different scores at different times, depending on the context, their mood, etc. The same problem occurs with sensory panel data. In common with all Bayesian systems, Bayesian networks model "degrees of belief", equivalent to probabilities, rather than a crisp true/false dichotomy. This means that uncertainty can be handled effectively, and explicitly represented.

3.5 Confidence Values

The output of any Bayesian model is a probability distribution, rather than simple scalar or vector. For example, whereas a neural network might predict a scalar preference score of say, 0.753, a Bayesian belief network might give an output in the form: $p(\text{low}) = 0.28$; $p(\text{high})=0.72$. This sort of information can be used as a measure of confidence in the result, which is essential if the model is going to be used for decision support.

3.6 Readability

When a (human) designer produces technical drawings and reports, the aim is to aid manufacture, sales, marketing and so on. When computers are being used to generate the designs automatically, it is important that they are still *readable*. No one is going to invest a great deal of time, money and expertise developing a product if they cannot see why it will be good. Due to their graphical nature, Bayesian networks provide a transparent model, although very complex systems may require networks too large to be comprehensible.

3.7 Prior Knowledge

It is impossible to avoid the use of prior knowledge when building models. By defining the bounds of the solution space, the representation used, the scoring measure used and so on, the analyst will inevitably introduce biases. Bayesian approaches make these prior assumptions explicit and formal. The size of the data sets also influences the use of prior knowledge. Because food design data sets are typically small, little information is contained in them, so the use of alternative, non-electronic sources of information (i.e. experts) is significant. This could be in the form of selecting nodes, sub-graphs or even entire graphs, if these are known to be important.

4. Bayesian Belief Networks Theory

Having described many of the features of Bayesian networks, it is now time to describe some of the processes involved in building and using them. There are three problems that must be solved: defining the graphical structure (B_s), defining the parameters in the form of the conditional probabilities (B_p), and finally using the models to make predictions. Further details of learning both the structure and parameters can be found in [13], and making predictions (inference) is covered in [12].

4.1 Defining the Structure

The graph consists of two parts: a collection of nodes and a collection of arcs joining them. In graph theory, these are known as vertices and edges respectively. In some cases, suitable expert knowledge may be available to allow the entire structure to be defined by hand, with the expert stating which variables are relevant, and how they interact. More often however, such knowledge will be unavailable, or at best, imperfect.

The total space of all legal (i.e. directed, acyclic) graphs over a set of nodes is greater than exponential in the number of nodes. Therefore, in all but the simplest cases, an exhaustive search is impossible, requiring the use of heuristics. A number of search algorithms have been used, a selection of which are listed in Table 3.

Method	Comment	References
K2	Finds parents for each node via a greedy search.	[15]
Genetic algorithms	Constraints are similar to the Travelling Salesman Problem. Used to provide node ordering for K2.	[16] [17]
Branch-and-Bound	Often used in AI to limit the combinatorial explosion, e.g. during feature selection.	[18]
Structural EM	Learns structure and parameters with a modified Expectation Maximisation (EM) algorithm.	[19]

Table 3: Structure search techniques

With any search technique, we need some way of determining the quality, or fitness, of a Bayesian network. Given that we are trying to model some data, the direct way of considering this is to ask "How well does the data fit this model?" The Bayesian approach to this problem is to assume that the data was actually *generated* by the model, and then reverse the question to ask "How likely is it that this model produced the data?" This reversal is possible using Bayes' Theorem [14].

The experiments described later use the "K2" algorithm proposed by Cooper and Herskovits [15], and outlined here. Cooper and Herskovits show that the ideal model, i.e. that which maximises the posterior probability of the network structure given the data, $p(B_s|D)$, also maximises the joint probability, $p(B_s, D)$. This is easier to calculate, and they derive a polynomial-time function of this joint probability, using the frequency of variable instantiations in the data set. This gives a straightforward way of quantifying the goodness of fit between the model and the data, and therefore defines a fitness function for the models. We now have to search through the model space to find a good network structure.

To make the search tractable, the search space is limited by making a number of assumptions. K2 assumes that: all the variables are discrete; all the cases are independent given the model; there is no missing data; there is no prior knowledge regarding likely structures. In the current work, these present no problems: the variables can easily be discretised; there are no dependencies between the cases; the cases are complete; and there is no knowledge about the structures. K2 requires a fixed ordering of the nodes, such that each node will only be considered as a possible parent of nodes that appear later in the ordering. The algorithm also requires a maximum fan-in value, i.e. an upper bound on the number of parents any single node may have. Finally, it requires a complete database of cases.

By definition, nodes depend only on their parents; K2 makes use of this by searching for the optimum set of parents of each node independently, before finally constructing the network. The algorithm proceeds by considering each node in turn, and defining an initially empty set of parents for that node. Every possible parent is then considered, and the parent that maximises the K2 score is added to the node's parent set. Further parents are considered within the constraints of node ordering and maximum fan-in, until no further additions improve the fitness score. Then the parents of the next node are considered. The end result is a list of parents for each node. This list is sufficient to completely define the structure of a Bayesian network.

4.2 Defining the Parameters

Once the graph has been defined, the only remaining parameters are the conditional probabilities for each node. Remembering that each node depends only on its immediate parents, we need only estimate $p(v|\pi_v)$ for each node, where v is the variable (node), and π_v is the set of parents of the node. All the variables must be discrete in order for the propagation and inference algorithms to work (see below), so continuous data must be converted to discrete values prior to use.

The simplest way of estimating the probabilities is to use the frequency with which each configuration of variables is found in the data. As the number of data points observed increases, this frequency will tend towards the true probability distributions; however, the small data sets typical of food design studies tend to be very sparse when considered this way, as many configurations will not have been observed. An alternative approach therefore is to initially assume a particular distribution (e.g. uniform) and then update this to encapsulate the information contained in the data. This can be done using the Expectation Maximisation (EM) algorithm [20], optionally combined with an equivalent sample size [21].

4.3 Inference

Given a complete model, defining both the structure and the conditional probabilities, we can begin to make predictions. If the values of some variables are known ("observed"), then the probabilities of the remaining variables can be calculated. This is done by fixing the states of the observed variables, and then propagating the beliefs around the network until all the beliefs (in the form of conditional probabilities) are consistent. Finally, the desired probability distributions can be read directly from the network. The standard propagation algorithm is due to Lauritzen and Spiegelhalter [2].

5. Data, Experiments and Results

Two experiments are described here. The first uses the K2 algorithm to build Bayesian networks, and compares their accuracy at predicting preference scores against two alternative models. The second experiment uses one such Bayesian network to estimate the characteristics of the "perfect" product under consideration.

The data used throughout the remainder of this paper was provided by Unilever Research, and consists of a preference score ("P1") and eight sensory panel scores ("S1" to "S8"). A total of just 20 records were available, each record being a complete set of data for a single sample of the food. The exact nature of the food is commercially confidential, but the samples were carefully selected to represent the full range of varieties of the product.

The raw data is included in Appendix 1. Each value was converted to a binary score, by assigning the lowest ten scores of each attribute to the class "low" and the ten highest scores to the class "high". A more precise model could be obtained by discretising each attribute into more than two states, but the data would become extremely sparse.

5.1 Performance Measures

Two measures of performance were used: predictive accuracy and joint probability. To calculate the predictive accuracy of each model, the preference (P1) was treated as a target class, which the sensory scores (S1-S8) were used to predict. The

accuracy score is simply the proportion of records that were assigned to the correct preference class (high or low). However, if the same data is used to both build and test the model, the resultant score tends to underestimate the model's true accuracy [14]. This is particularly true when small data sets are used (as in the current work), because models will tend to overfit the data, fitting both the underlying distribution and the inherent noise. To avoid this, leave-one-out cross-validation is often used. Thus given 20 records, we construct 20 models, each being built using a different subset of 19 records, and each being tested on the remaining record. This produces 20 accuracy scores, the mean of which is a good estimate of the model's true accuracy.

The maximum likelihood model is that which maximises $p(\mathbf{B}_s|\mathbf{D})$, which [15] show is proportional to $p(\mathbf{B}_s, \mathbf{D})$, the joint probability of the network and the data. Therefore in this work, this joint probability was calculated for every Bayesian network and the naïve Bayes classifier (described below). The entire data set was used to build each model and thus to calculate this probability score without the need for any cross-validation. This is because when performing Bayesian inference, complex models have lower prior probabilities than simple models, giving Bayesian techniques a built-in safeguard against overfitting. Note that no equivalent score exists for standard neural networks.

5.2 Model Performance

As an initial study, three techniques were compared to see which could most accurately predict preference from the sensory scores. Table 4 summarises the results.

The three techniques used were a neural network, a naïve Bayes classifier, and a Bayesian belief network. The neural network was a standard MLP, with 8 inputs (the sensory data), 5 hidden nodes and one output (the preference score). Leave-one-out cross-validation was used to measure the neural networks' accuracy at predicting the preference score, so each evaluation cycle actually consisted of building and testing 20 neural networks. Of 100 cycles, the mean accuracy was 0.796. [14] describes neural networks in more detail, as well as several cross-validation techniques.

The naïve Bayes classifier is a special case of Bayesian network that treats one variable as a target class. It assumes that all the other variables depend only on this class, being conditionally independent of each other. Here, we treat preference (P1) as the target class and assume that the eight sensory scores are independent. This produces a network where every sensory node has exactly one arc, which leads from the preference node, as shown in Figure 2. Using leave-one-out cross-validation gave an estimated accuracy of 0.80 for the naïve Bayes classifier. The log

	<i>Accuracy</i>		<i>Log Joint Probability</i>	
<i>Neural Network</i>	0.796	(0.05)	<i>N/A</i>	
<i>Naïve Bayes Classifier</i>	0.800	(0.00)	-123.08	(0.00)
<i>Bayesian Belief Network</i>	0.800	(0.00)	-107.51	(1.03)

Table 4: Comparison of models.

"Accuracy" is the estimated accuracy with which the model predicts P1 given S1-S8.

"Log Joint Probability" is the log of the joint probability of the model and the data, $\ln p(\mathbf{B}_s, \mathbf{D})$. The standard deviation of each score is shown in brackets.

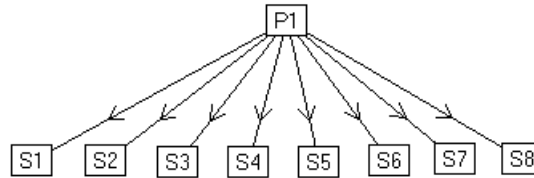


Figure 2: Naïve Bayes Classifier

joint probability was -123.08. Note that the corresponding variances shown in the table are zero, because the network structure is fixed, so the accuracy score has no variance.

The Bayesian belief networks used in this study were generated using the K2 algorithm, and Figure 3 shows one such network. The K2 algorithm was executed 100 times, with randomly generated node orderings. In each case, the joint probability $p(B_s, D)$ was calculated, and had a mean of -107.51. A separate experiment repeatedly used K2 with leave-one-out cross-validation. In every case, K2 selected the same two variables (S5 and S6) as parents, and so gave the same accuracy score of 0.80. This shows that (at least for this data set) the ordering of the nodes presented to K2 is not critical.

These results show that Bayesian belief networks, neural networks and naïve Bayes classifiers are equally effective at the specific task of predicting product preference from sensory panel scores.

Note that the Bayesian belief networks are constructed to model the *entire* data set, rather than just one relationship within it. In contrast, the other two techniques used here explicitly build models that are designed to predict preference. The final column of Table 4 shows that as well as making equally good preference predictions, the Bayesian network models the data more closely than the naïve Bayes classifier, as indicated by the higher log probability value. This suggests that the assumptions made by the naïve Bayes classifier are invalid, and therefore that the sensory panel variables are *not* independent.

5.3 Belief Propagation

If Bayesian networks are no more accurate than simpler alternatives, why use them? As outlined in section 3, Bayesian belief networks have many attractive features, including abduction: the ability to diagnose likely causes of an observed effect. In the current work, this is estimating the most likely characteristics of a hypothetically perfect product.

The Bayesian network shown in Figure 3 is used here to demonstrate how

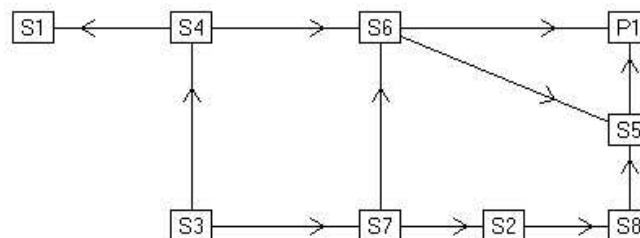


Figure 3: Bayesian network for sensory data

predictions can be made from limited observations. The parameters (probabilities) were defined using the data frequencies only, and the Lauritzen and Spiegelhalter algorithm [2] was used to propagate several observations and to make predictions.

Graph (a) in Figure 4 show the effect of observing, for some hypothetical sample, that the value of S1 is low. The chart shows the nine variables used in the model (Figure 3) with the first bar of each pair showing the *prior* probability of the variable having a "high" value, and the second bar showing the corresponding *posterior* probability, after the observation and belief propagation. For example, the prior belief that any given sample would have a high S1 measure is roughly 0.5, while the posterior is 0.0 - we are stating the level is low, so the probability of it being high is zero. The probability of a high S4 has increased from 0.2 to 0.5, suggesting that S1 and S4 are inversely correlated to some extent, so that a low S1 tends to "cause" a high S4. Finally, the probability of a high preference score (P1) decreases from 0.4 to 0.2, suggesting that the preference panel dislike products with a low S1 score.

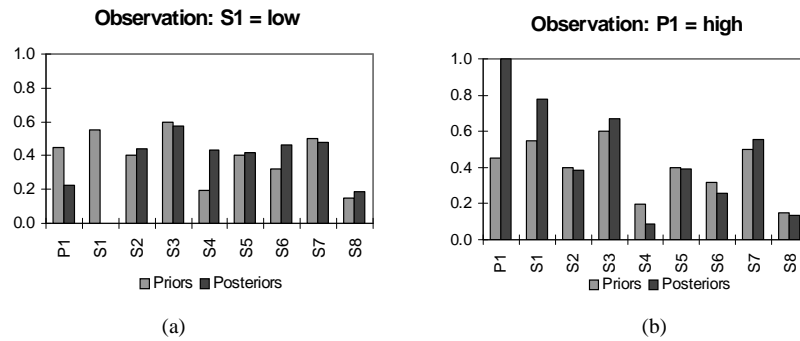


Figure 4: Single variable observations

Chart (b) shows the effect of observing a high preference score (P1). Here we are asking, "What features must a product have in order to be preferred?" The observation leads to an increase in the belief that the sample will have a high S1 score and a low S4 score. The other variables are largely unchanged, suggesting they have little direct influence over preference. The posterior probabilities here describe the "perfect" product according to the model.

6. Conclusions

Bayesian Belief Networks are a valuable addition to the product designers' toolkit. They are powerful tools for developing graphical models from a combination of data and expertise. They can be built from modest data sets, with or without background knowledge, and yet are scalable because they are afford local optimisation. The results here show that they are as accurate as neural networks, but with the advantage of being reversible. This allows probabilistic predictions of optimal designs to be made, and these models are now being used to aid consumer preference modelling.

7. References

- [1] AAFC 1991. "A Profile of the Canadian Speciality Food Industry" Market Report produced by the Canadian Department of Agriculture and Agri-Food.
- [2] Lauritzen, S L, Spiegelhalter, D J 1988. "Local computations with probabilities on graphical structures and their application to expert systems" *Journal of the Royal Statistical Society*, Vol. 50 No. 2 pp.157-224.
- [3] Pearl, J 1988. *Probabilistic Reasoning in Intelligent Systems: networks of plausible inference* Morgan Kaufmann.
- [4] Barnett, G O, Famiglietti, K T, Kim, R J, Hoffer, E P, Feldman, M J 1998. "DXplain on the Internet" in *American Medical Informatics Association 1998 Annual Symposium*
- [5] Helm, L 1996. "Improbable Inspiration" *Los Angeles Times*, October 28, 1996.
- [6] Berler, A, Shimony, S E 1997. "Bayes Networks for Sonar Sensor Fusion" in Geiger, D, Shenoy, P (eds) 1997. *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence* Morgan Kaufmann.
- [7] Neil, M, Littlewood, B, Fenton, N 1996. "Applying Bayesian Belief Networks to Systems Dependability Assessment" in *Proceedings of Safety Critical Systems Club Symposium, Leeds, 6-8 February 1996* Springer-Verlag.
- [8] Frey, B J 1998. *Graphical Models for Machine Learning and Digital Communication* MIT Press.
- [9] Ezawa, K J, Schuermann, T 1995. "Fraud/Uncollectible Debt Detection Using a Bayesian Network Based Learning System: A Rare Binary Outcome with Mixed Data Structures" in Besnard, P, Hanks, S (eds) *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* Morgan Kaufmann.
- [10] Roozenburg, N F M, Eekels, J 1995. *Product Design: Fundamentals and Methods* Wiley.
- [11] Krause, P, Clark, D 1993. *Representing Uncertain Knowledge: An Artificial Intelligence Approach*, Intellect Books.
- [12] Jensen, FV 1996. *An Introduction to Bayesian Networks* UCL Press.
- [13] Heckerman, D 1995. "A Tutorial on Learning With Bayesian Networks", Microsoft Research report MSR-TR-95-06..
- [14] Bishop, C M 1995 *Neural Networks for Pattern Recognition* Oxford University Press
- [15] Cooper, G F, Herskovits, E 1992. "A Bayesian Method for the Induction of Probabilistic Networks from Data" *Machine Learning* Vol. 9 pp. 309-347.
- [16] Larranaga, P, Kuijpers, C M H, Murga, R H, Yurramendi, Y 1996. "Learning Bayesian network structures by searching for the best ordering with genetic algorithms" *IEEE Trans on Systems, Man and Cybernetics-A* Vol. 26 No. 4 pp.487-493.
- [17] Etxeberria, R, Larranaga, P, and Pikaza, J M 1997. "Analysis of the behaviour of the genetic algorithms when searching Bayesian networks from data", *Pattern Recognition Letters* Vol. 18 No 11-13 pp 1269-1273
- [18] Narendra, P M, Fukunaga, K 1977. "A Branch and Bound Algorithm for Feature Subset Selection" *IEEE Transactions on Computers*, Vol. 26, No. 9, pp. 917-922.
- [19] Friedman, N 1998. "The Bayesian Structural EM Algorithm" in Cooper, GF, Moral, S (eds) *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* Morgan Kaufmann.
- [20] Dempster, A P, Laird, N M, Rubin, D B 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm with discussion", *Journal of the Royal Statistical Society, Series B*, Vol. 39 pp.1-38.
- [21] Mitchell, T M 1997 *Machine Learning* McGraw-Hill

8. Appendix: Data Set

The table below contains the raw data used in this work. Each row represents a record for a single sample. Column "P1" is the preference score, the remaining columns being eight sensory scores.

This data is also available in ASCII format from this URL:
<http://www.cs.ucl.ac.uk/staff/D.Corney/FoodDesign.html>

	P1	S1	S2	S3	S4	S5	S6	S7	S8
1	3.329	6.050	2.560	6.373	2.649	3.587	1.670	6.230	1.012
2	3.700	5.659	2.577	4.579	3.377	5.278	3.119	2.457	1.206
3	4.004	5.442	1.495	8.175	2.384	4.315	2.133	8.669	0.930
4	2.400	6.185	1.607	7.763	1.948	1.646	3.435	8.374	0.940
5	3.109	4.391	1.916	6.748	3.628	4.220	2.206	6.995	0.982
6	8.253	7.848	2.687	8.258	1.482	9.606	0.992	4.881	1.165
7	5.160	5.834	1.536	8.588	2.348	7.116	1.228	8.902	1.025
8	4.240	6.506	1.854	6.325	2.267	4.370	1.866	5.510	1.010
9	1.784	1.854	3.400	3.739	7.736	1.378	8.607	6.603	1.046
10	6.262	8.248	0.848	8.857	2.042	7.375	1.087	8.896	0.935
11	2.087	1.920	4.680	2.627	8.152	2.797	5.017	4.837	1.068
12	5.287	8.073	1.016	8.598	2.189	4.294	1.329	8.635	0.951
13	6.180	7.023	2.279	6.321	2.148	6.683	1.213	5.041	1.016
14	2.538	5.836	2.033	6.886	2.227	2.735	2.982	7.033	1.070
15	7.987	8.312	2.535	8.848	1.273	8.034	1.132	5.984	1.029
16	3.587	7.289	1.411	7.863	1.510	2.969	1.508	7.854	0.987
17	5.131	3.586	6.424	1.943	2.106	8.186	1.018	5.313	2.731
18	2.211	5.210	1.765	7.862	2.087	1.512	2.466	9.283	1.022
19	7.298	7.147	3.889	6.666	2.109	8.161	1.022	3.699	1.028
20	5.318	7.501	1.667	7.670	1.338	4.870	1.075	6.958	1.053

Acknowledgements

Unilever Research Ltd. have generously sponsored this work, and have provided data and advice throughout. The research was undertaken within the Postgraduate Training Partnership established between Sira Ltd and University College London. Postgraduate Training Partnerships are a joint initiative of the Department of Trade and Industry and the Engineering and Physical Sciences Research Council.