

Chapter 5

Future work

5.1 Introduction

The complexities of analysing food design data raise many questions, of which we have attempted to answer only a few. In this chapter, we describe four areas of potential future work. In Section 5.2, we discuss several ways in which the same data sets that we have been using could be analysed using the same techniques discussed in previous chapters, but applied in different combinations. In Section 5.3, we discuss alternative methods, broadly classified as intelligent data analysis, which could be applied to the same class of data sets. In Section 5.4, we briefly outline alternative sources of data, which could benefit from similar approaches to data analysis. Finally in Section 5.5, we propose changes to the current data gathering process, which would aid analysts and improve analysis. Each of these proposals would lead to alternative models, new results, and potentially, to increased consumer satisfaction.

5.2 Alternative routes to intelligent food design

Given the tools that we have introduced in the previous chapters, we can now consider different ways of analysing the data. Below, we summarise seven possible routes towards food design, starting with one introduced earlier.

5.2.1 Virtual consumers

- Cluster preference scores
- Treat centre of each cluster as a “virtual consumer”
- Perform feature selection and regression to predict preferences of novel products

This allows us to understand what drives each market segment, and to design near-optimal products for each group. We consider the mean of a group of consumers as a sensible summary of individual preferences. This approach assumes that all consumers who express similar preferences are driven by the same sensory features. This information is not provable with the type of data currently available, because we have no data about the preference panellists except for their preference scores. In particular, we do not have access to their sensory experiences when tasting the foods.

If we perform regression (including feature selection) separately for each cluster, we can produce a combined performance estimate by weighting the errors of each regression model according to the size of the cluster (i.e. the number of consumers modelled). This is the subject of Sections 3.9 and 4.6.8.

Alternatively, we could use a model with multiple outputs, such as an RBF network. In this case, the input to the model would be the sensory features of a product, and the output would be a vector of preferences, such that each element is associated with a different consumer cluster. The hidden nodes of such a network model would contain information shared across several (separately identified) clusters, and it has been argued that such distributed representations lead to improved generalization (e.g. Hinton et al. [65]), and therefore to better predictions of consumer behaviour than has been achieved in this work.

5.2.2 Consumer committees

- Cluster preference scores
- Treat each cluster as a committee of voters
- Perform feature selection and regression to predict preferences of novel products

This is similar to the approach in section 5.2.1, but with a more natural interpretation of treating preference panellists as focus groups instead of as “virtual consumers”. We treat each preference record as an committee member who votes on the importance of each feature and on the optimum values of each feature. If every consumer is given an equal vote, the results will be identical to those of section 5.2.1. However, if we allow each consumer to have a different weight, then we would have a more flexible model. For example, if we used weights inversely proportion to the distance to their nearest neighbour, then we could bias the models towards consumers that are more representative, and reduce the influence of outlying consumers within each cluster. This would lead to predictions that match target consumers more closely, rather than trying to model all (potential) consumers with equal accuracy. Coupled with marketing information, this could lead to increase satisfaction amongst key consumers.

5.2.3 Grouping consumers drivers

- Treat each preference panellist individually
- Perform feature selection to identify each individual's drivers
- Cluster the drivers
- Find “virtual consumers” in terms of shared drivers

To design food, we need to identify drivers, which is made explicit with this approach. Selecting features for a single panellist is difficult in the sense that with such a small sample (one record), the results will be unverifiable. However, the data sets each have several hundred panellists; if most feature sets are approximately correct, then the clustering should be robust to a few outliers, especially if we incorporate an outlier detection routine into the analysis. Ideally, the “virtual consumers” found via this approach should tally with those found by other methods outlined here, giving a further check on model validity. This gives us an alternative method for performing dimension reduction, while retaining interpretability about key drivers.

5.2.4 Grouping product drivers

- Treat each product individually
- For each product, perform feature selection to identify drivers
- Compare drivers across products

This treats all consumers identically, without any attempt at clustering. We could enhance it further, by clustering consumers at the second step, and identifying different drivers for each group. The key here is to discover why consumers like or dislike each product individually, rather than assume that the same drivers apply across the entire product range. If very different drivers are found to be important for each product, this calls into question attempts at finding a single group of drivers for all the products (e.g. section 5.2.1).

5.2.5 Virtual products

- Cluster products in sensory feature space
- Treat the centre of each cluster as a “virtual product”
- Cluster consumers, perform feature selection and regression as in section 5.2.1

By replacing a group of similar products with a single “virtual product”, we can reduce the number of dimensions in the preference data set. Given the sparseness of the preference data, reducing the dimension is very appealing. However, we are starting with a small set of products, so each cluster will contain very few, making the centre of each cluster rather ill defined. Further, it assumes that the mean of two (or more) products is a sensible concept, which may or may not be the case. Note that the mean is in feature space rather than ingredient space.

5.2.6 Separate products

- Treat each product individually
- For each product, cluster the consumers in a univariate preference space
- Compare consumer clusters found for each product

Because the clustering takes place in a one-dimensional space, the data is much more dense than the clustering described in Chapter 3 (in a m -dimensional space, with m products). This gives greater confidence in the results, and allows us to use information criteria, such as MDL, to estimate the number of clusters more reliably.

However, there is likely to be little or no correlation between each cluster. If we project a set of clusters in a high dimensional space onto a space with far fewer dimensions, the clusters are unlikely to remain clearly defined. A more standard approach to dimension reduction (such as feature selection or PCA) may be more effective.

5.2.7 Separate products and virtual consumers

- Treat each product individually
- Cluster preferences in univariate preference space
- Define “virtual consumers” at centre of each cluster
- Perform sensory feature selection and regression to predict preferences
- Compare drivers and models for each product and for each virtual consumer

This combines ideas from Sections 5.2.1 and 5.2.6, and therefore combines their weaknesses as well as their strengths. The aim is to increase our confidence in clustering consumers, by using a one-dimensional space, while recognising that each segmentation is likely to be different for the different products.

5.3 Alternative methods for data analysis

In this section, we discuss several machine learning methods that we have yet to investigate. Some are extensions to methods that we have described already; others are introduced here for the first time.

5.3.1 Semi-supervised feature selection

In Section 2.5.2, we introduced a semi-supervised feature selection algorithm based on Schuurmans' ADJ test [115]. We used this approach to improve forward sequential feature selection in Section 2.8, and demonstrated that it produces models with fewer features and the same error as standard FSS. The same semi-supervised feature selection could be used to enhance other feature selection methods, such as simulated annealing (Section 2.4.3), which may be more robust than FSS. The only requirement for ADJ is a sequence of models of increasing complexity, and this can easily be achieved if we constrain SA to use a specified number of features.

5.3.2 Extending cross model validation

The cross model validation algorithm was presented in Section 2.5.1, p. 51, as a means to simultaneously control the number of features to be used, to select the feature subset. In Section 2.11.2 p. 72, we noted difficulties in selecting other model parameters, such as the number of nodes in a RBF network.

Figure 5.1 shows an extended version of cross model validation, which allows three sets of parameters to be chosen simultaneously, namely the number of features, the feature subset, and the number of RBF nodes. In principle, this algorithm could be extended to optimise any number of parameters, while still avoiding overfitting. One weakness of this approach is that for each extra parameter set we wish to optimise, the central model-building loop of the algorithm will use one fewer record. With a small data set, and a large number of parameter sets, we may have very few records to estimate each model's parameters, and so end up choosing very carefully between a large number of very poor models. Conversely, with a large data set, this approach could become very slow, due to the nesting of the loops.

5.3.3 Fuzzy Rand index

We described the Rand index in Section 3.5.1 as a method to measure the similarity of two segmentations of a data set. The Rand index is calculated by counting the number of agreements and disagreements between two segmentations. We noted earlier (p. 101) that one drawback of the Rand index is that each point is regarded as belonging to exactly one cluster, which makes counting agreements straightforward. However,

Given $n \times m$ data set, D , a feature selection routine FS, and a maximum number of basis functions r , do:

- 1 Initialise CMV, a m -dimensional vector of zeros
- 2 For $i = 1$ to n
 - 2.1 For $j = 1$ to m
 - 2.1.1 Initialise CMV' , a r -dimensional vector of zeros
 - 2.1.2 For $k = 1$ to r
 - 2.1.2.1 $h_s = \text{FS}(D_{ij}, j, k)$
 - 2.1.2.2 $CMV'(k) = CMV'(k) + \text{err}(h_s, d_j)$
 - 2.1.3 $OBFS = \min_k CMV'$
 - 2.1.4 $h_s = \text{FS}(D_i, j, OBFS)$
 - 2.1.5 $CMV(j) = CMV(j) + \text{err}(h_s, d_i)$
 - 2.2 next j
- 3 next i
- 4 $OMS = \min_j CMV$
- 5 $h_o = \text{FS}(D, OMS, OBFS)$

where:

- $D_i = D \setminus d_i$, i.e. data set with i^{th} record removed;
- $D_{ij} = D \setminus (d_i, d_j)$, i.e. data set with i^{th} and j^{th} records removed;
- OMS is the estimated optimum model size;
- $OBFS$ is the estimated optimum number of basis functions;
- $\text{FS}(D, S)$ is a function that performs feature selection using data D constrained to use exactly S features, and returns a single regression model;
- $\text{FS}(D, S, R)$ is a function that performs feature selection using data D constrained to use exactly S features, and returns a single RBF regression model which contains exactly R basis function;
- $\text{err}(h, d)$ is the error for model h estimated from test set d ; and
- h_o is the final model selected.

Figure 5.1: Modified cross model validation algorithm, including RBF network size selection and feature selection

clustering algorithms such as GMMs and fuzzy c -means assign points fractionally to several clusters at once. If the sum of the assignments is constrained to be one, then it would be straightforward to extend the Rand index to allow fractional assignments. This would give a more accurate indication of the similarity between two segmentations.

Consider a point fractionally assigned to two different clusters in one segmentation, with weights $(0.3, 0.7)$. In the second segmentation, this point is given weights $(0.4, 0.6)$. We can define the disagreement between these as the absolute difference in these values, giving a score of $|0.3 - 0.4| = |0.7 - 0.6| = 0.1$ in this case. Subtracting this disagreement score from one gives us the agreement score: 0.9, in this case. If we sum these agreement scores, and then divide by the total number of records, we get a normalised score between zero and one. A score of one corresponds to perfect agreement, and zero corresponds to no agreement, as with the conventional “crisp” Rand index. The standard “crisp” version forces us to discard information from cluster models when points are fractionally-assigned. The proposed fuzzy Rand index retains this information, as recommended by Marr’s “Principle of Least Commitment” [77].

5.3.4 Analysis of GMM components

We discussed Gaussian mixture models in Section 3.4.5, and noted there that analysis of the elements of the covariance matrices could provide useful information. A small variance score on leading diagonal of a GMM component’s covariance matrix suggests that the members of the corresponding cluster show consistent preferences. It is more important to model these preferences accurately than it is to model other, more inconsistent, preferences. Thus we could introduce a cost term into regression, where the accuracy with which the preferences of different products are predicted is penalised according to the importance of modelling that product. This cost term could vary between clusters or between products, providing us with greater control over the analysis, and leading to predictions of greater precision.

5.3.5 Catalogue design

Kleinberg et al. [74] argue that the aim of data mining is to produce actionable results, rather than just finding patterns, an idea which extends data mining into decision analysis. One standard problem in decision analysis is the “Knapsack Problem”. Suppose a thief has a knapsack of a certain capacity. He can steal items of various sizes and various values. He would like to choose items that have the greatest possible value yet still fit in his knapsack.

A related problem is also defined by Kleinberg et al. [74], namely the “catalogue segmentation” problem. Here, we must choose r_k items to put into each of k catalogues. Each consumer is then sent one catalogue, from which they select their preferred item,

thus leading to a high consumer satisfaction. There is a cost associated with producing each catalogue. The goal is to maximise the total satisfaction of the consumers, while minimising the number of catalogues produced.

In the case of food design, each catalogue corresponds to a distinct market segment, which is to be targeted through marketing and advertising with a particular group of products. Consumers are free to choose any product within the catalogue. Realistically (and perhaps unfortunately), competitors' products would also appear in the same catalogues.

This formulation allows us to move beyond simply producing one product for each market segment. We would want to minimise the number of catalogues and the number of products in each catalogue, while still maximising total customer satisfaction. We would no longer be attempting to produce a perfect product, but rather to produce a range of sufficiently good products with which to satisfy our customers.

5.3.6 Sequential outlier rejection for classification and regression

The sequential outlier rejection algorithm was introduced in Section 4.6 for detecting outliers while clustering. The same approach could be applied to classification and regression (c.f. Section 4.6.8, where we stored the regression error, but rejected points according to the cluster model error).

As we outlined in Section 4.4.2, Tax and Duin [125] use a linear classifier to produce a decision boundary, and via bootstrap sampling, measure the stability of classification to detect outliers. Instead, we could build a classifier using all the available data, then locate the data point that is least well modelled, and remove that data point. By iterating this process, a sequence of rejections can be produced similar to the clustering rejection sequences described earlier.

Consider a single row from our data, $x_i \in X$. In clustering, we treat this as a point in the data space, and minimise some error criterion as a function of x_i . In regression (or classification), we split the vector x_i into two components, to be treated as “input” and “target”, respectively:

$$\begin{aligned}x_i^{input} &= x_{i,1..(m-1)}, \\x_i^{output} &= x_{i,m}\end{aligned}$$

This takes advantage of the fact that we can separate the problems of outlier detection and data modelling, so we can use regression (or classification) techniques for the former, and clustering techniques for the latter. It also gives us a wide range of further algorithms that can be applied, such as support vector machines, RBF networks, decision trees, etc.

However, an extreme point may lie exactly on a regression line, or be a long way

on the correct side of a classification boundary, but in either case, be so far removed from the rest of the data that it should be regarded as an outlier. The SOR algorithm would not reject these points, and neither would Tax and Duin’s classifier instability method [125]. Combining one of these techniques with a nearest-neighbour method might improve the results.

5.3.7 Sequential outlier rejection for feature selection

Rather than iteratively removing the least well modelled data point, we could remove the least well modelled *feature*. Consider the algorithm shown in Figure 5.2. We start with a data set, X , which is an $n \times m$ matrix, with n records and m features. We temporarily remove one data point, and use the remaining data to estimate the “missing” value. By repeating this for every data point in X , we can produce an $n \times m$ matrix of errors, E . This measures how consistent each point is with the rest of the data. If we sum these values along each row, giving n scores, and choose the largest, we can reject outlying data points as before, i.e. the point that is least consistent with the others. But if we sum each *column*, given m scores, we can measure how consistent each feature is, with respect to the rest of the data. Columns with large sum error scores correspond to features that are poorly modelled by the remaining features. This provides us with an alternative wrapper method for evaluating features (Section 2.3).

If one feature cannot be predicted by the remaining features, it does *not* necessarily follow that the feature is useless. In fact, given that strongly correlated features are redundant, we might want to remove those features that given the *lowest* errors, i.e. those features that can be predicted from the others.

With a linear regression model, this feature selection would be equivalent to the correlated based feature selection method, described in Section 2.3. However, a non-linear regression model, such as an RBF network, may reveal more complex feature redundancies. When we build a non-linear regression model, it makes sense to combine it with a non-linear feature selection method. Chapter 4 showed promised results using SOR, suggesting that further analysis is justified.

5.3.8 Replacement machine learning methods

Chapter 2 considered linear regression and RBF regression models. We could use many other regression techniques, such as support vector machines (Vapnik [132]), decision trees (Breiman [20]) or multi-layer perceptron neural networks (Bishop [15]). Given that the main problem with food design is the lack of data, it seems unlikely that these would produce substantially better results than the two approaches examined. These alternative methods all have many parameters to estimate, and that may require more

<p>Given data X, an $n \times m$ matrix, consisting of n records with m features, repeat:</p> <ol style="list-style-type: none"> 1. for $i = 1 \dots n$ 2. for $j = 1 \dots m$ 3. Define $X' = X \setminus x_{ij}$ 4. Build model $f_{ij} : X \rightarrow x_{ij}$ 5. Calculate error, $e_{ij} = (f(X') - x_{ij})^2$ 6. end 7. end 8. Rejection feature $j = \max_j \Sigma e_{.j}$ <p>$\Sigma e_{.j}$ is the sum of the j^{th} column of e</p>

Figure 5.2: Sequential Feature Rejection Algorithm

data than we have available. However, some methods provide alternative means of performing feature selection, such as decision trees, as we mentioned in Section 2.3.

Similarly, Chapter 3 compared two clustering techniques (k -means and Gaussian mixture models); these could be further compared with hierarchical models, fuzzy c -means clustering, graphical methods and so on (e.g. Everitt [42]). The different performances of these methods may reveal further information about the nature of clusters, following the discussion in Section 3.9.1. For example, many hierarchical clustering methods are known to be very sensitive to noise, although they may be otherwise useful for detecting outliers [42].

5.4 Alternative applications

In this section, we consider some alternative approaches to improving customer satisfaction, and therefore one hopes, sales of consumer goods. The intelligent systems used to aid food design could equally be applied to perfume design, clothes design and any other fast-moving consumer goods. Consumers' preferences of these products could still be measured with the same sort of preference panel used in food studies. Sensory panels could also be retained, albeit in different forms (e.g. replacing flavour and odour for food, with material and cut, for clothing). In the following sections, we move away from preference panels and sensory panels, to discuss alternative forms of consumer goods design.

5.4.1 Mass customisation of consumer goods

“Mass customisation” refers to the mass production of a product (with the beneficial economies of scale) combined with some level of customisation (to maximise customer satisfaction). One example is buying a new car: having chosen the make and model, customers then choose whether to have air-conditioning, a sunroof, alloy wheels, and so on, in sharp contrast to Ford’s maxim quoted in the introduction (Section 1.2.1).

Suppose that with the packaging of a processed food product there was a brief questionnaire with questions such as “was the product too salty?” or “was the product sweet enough?” The consumer ticks the relevant boxes and returns it to the manufacturer. A batch of the product is then produced which more closely matches this consumer’s preferences. One box is then delivered to the consumer’s local food retail outlet, ready for them to collect next time they go shopping.

Of course, the profit margins on new cars are rather larger than those on foods; consumers will “happily” pay an extra few hundred pounds for a customised car, but would balk at paying that much for a tin of baked beans. But given that mass customisation has been applied to children’s toys (customers can choose doll’s hair colour and skin tones from www.Barbie.com) and clothing (such as personalised Nike trainers), perhaps food is the next area. In fact, simple examples of customised food are already available. Customers can choose the colour of “M&M” chocolates, and Hershey’s offer chocolates with personalised messages. The next logical step is surely customised recipes.

5.4.2 Collaborative filtering for food

One aspect of data mining that has grown with the internet is collaborative filtering, also known as recommender systems. Service providers, such as Amazon.com, maintain a database of the preferences of a number of people using the system, and use this to make predictions for new (or existing) users. The preferences are usually stored as votes, which can be either implicit (e.g. frequency of purchases) or explicit (e.g. expressed a like or dislike of the product), equivalent to the direct and indirect measurement of preferences outlined in Section 1.4.2.

Breese et al. [18] give an empirical comparison of several such methods, applied to television, film, and web site preferences, which are typical applications of collaborative filtering. The same idea could be applied to food recommendations. Suppose Mr. John Doe likes both Granny Smith’s and Cox’s Orange Pippins apples. A second consumer is known to like Granny Smith’s, so (lacking further information) we may predict that this person will also like Cox’s Orange Pippins. One danger of this approach is making too many false-positive predictions, leading to customer dis-satisfaction. Nonetheless, this approach has been applied by retailers with some success, so application to man-

ufacturer's seems sensible.

5.5 Data gathering

In this section, we propose improvements to the data gathering process. Preference panels are composed of untrained consumers, who will suffer from taste-fatigue if presented with too many samples. The resultant data set will always be limited in size therefore.

Sensory panels are trained and used over a period of time, and so don't tend to suffer from taste-fatigue. This means that they can be presented with many more product samples than the preference panels. This would provide us with many more unlabelled records, which would improve the semi-supervised learning models, as discussed in Section 2.11.5 and in the Appendix A.3. The results in these sections show the potential benefits of using unlabelled data, although further research is required to estimate how much extra is useful.

As we discussed in Section 2.12, accurate models can be built more easily when only a small set of features are available. We therefore suggest that sensory panels are encouraged to select fewer features than is currently the norm. This would ease feature selection, and allow more accurate regression models to be built. It would also make better use of the expertise present in these panels.

5.6 Conclusions

In Chapter 1, we noted that there are many paths through the food design process (Section 1.5). In this chapter, we have considered several alternative paths by treating products, consumers and drivers in different ways, while using the same statistical and machine learning methods described in Chapters 2–4. We have now also been considered several further approaches, and further related applications. Given the complexities of food data, and of the food industry, the possibilities are endless. Several of the techniques proposed in this chapter are also applicable to more general data analysis, such as applying SOR to regression (Section 5.3.6), and the alternative “fuzzy” Rand index (Section 5.3.3). People will always want food, and manufacturers will always face strong competition, so it is vital that manufacturers continue to improve design methods. The results presented in Chapters 2–4 merit the further analyses proposed in this chapter.