LATENT VARIABLE METHODS: CASE STUDIES IN THE FOOD INDUSTRY

LATENT VARIABLE METHODS: CASE STUDIES IN THE FOOD INDUSTRY

By EMILY NICHOLS, B.Sc. Eng., P.Eng

A Thesis Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree Master of Applied Science

Master of Applied Science (2011) McMaster University

(Chemical Engineering) Hamilton, Ontario

TITLE: Latent Variable Methods: Case Studies in the Food Industry

AUTHOR: Emily Nichols, B.Sc. Eng., P.Eng

SUPERVISOR: Dr. John F. MacGregor

NUMBER OF PAGES: x, 102

Abstract

Accommodating changing consumer tastes, nutritional targets, competitive pressures and government regulations is an ongoing task in the food industry. Product development projects tend to have competing goals and more potential solutions than can be examined efficiently. However, existing databases or spreadsheets containing formulas, ingredient properties, and product characteristics can be exploited using latent variable methods to confront difficult formulation issues. Using these methods, a product developer can target specific final product properties and systematically determine new recipes that will best meet the development objectives.

Latent variable methods in reformulation are demonstrated for a product line of frozen muffin batters used in the food service industry. A particular attribute is to be minimized while maintaining the taste, texture, and appearance of the original products, but the minimization is difficult because the attribute in question is not well understood. Initially, existing data is used to develop a partial least squares (PLS) model, which identifies areas for further testing. Design of experiments (DOE) in the latent variable space generates new data that is used to augment the model. An optimization algorithm makes use of the updated model to produce recipes for four different products, and a significant reduction of the target attribute is achieved in all cases.

Latent variable methods are also applied to a difficult classification problem in oat milling. Process monitoring involves manually classifying and counting the oats and hulls in the product streams of groats; a task that is time-consuming and therefore infrequent. A solution based on near infrared (NIR) imaging and PLS-discriminant analysis (PLS-DA) is investigated and found to be feasible. The PLS-DA model, built using mixed-cultivar samples, effectively separates the oats and groats into two classes. The model is validated using samples of three pure cultivars with varying moistures and growing conditions.

Acknowledgements

The projects described in this thesis were made possible by many individuals and organizations. Dr. John F. MacGregor took me on as a graduate student even as he was attempting to reduce his academic workload, and I cannot thank him enough for his teaching and guidance. This degree is the realization of a longtime dream.

Many thanks go out to my colleagues and friends at PepsiCo Foods Canada Inc. It was a pleasure to collaborate with them on two very interesting projects. I am extremely grateful for all of the experimentation and data collection that they undertook on my account, and for the opportunity to contribute to such practical industrial efforts.

I wish to thank McMaster University, the department of Chemical Engineering, and the McMaster Advanced Control Consortium for the support I received over the past two years. Faculty, staff and students offered valuable input and technical assistance as I completed my coursework and thesis.

Members of the Acurum development group at DuPont Canada provided access to their Acurum instrument on several occasions and were generous with their time.

The team at ProSensus, Inc., were also very helpful. I am especially grateful to their software developers who responded in a timely fashion to my many queries and requests for new features.

My sincere appreciation goes out to so many friends and family who were supporters of this endeavor. Some even got their hands dirty helping with the sorting and photography of thousand of oats and groats. Thank you, thank you, thank you. And to my husband, Marcel Verner, your endless support and encouragement means the world to me.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

PLS and PCA
Even PLS-DA
All will be discussed in time
Through studies, tables, plots and rhyme.

This thesis is comprised of two distinct case studies, namely (i) the rapid reformulation of frozen muffin batters and (ii) the applications of near infrared (NIR) imaging in oat milling. Both studies demonstrate the application of latent variable methods to specific problems in the food industry. Rapid reformulation makes use of partial least squares (PLS) regression, design of experiments (DOE) in the latent variable space, and optimization. Principle components analysis (PCA) and PLS-discriminant analysis (PLS-DA) are both used to analyze the NIR images collected in the oat milling case study.

The ability to rapidly develop new or reformulated products having specific properties is an economic benefit in many industries. The food industry in particular must respond to taste and lifestyle trends, nutrition research, corporate objectives and government regulation with a steady stream of new and/or improved products. Product development seeks to achieve competing goals such as improved nutrition, appealing flavour, and reduced cost. Typically advances are made in an ad-hoc fashion based on past successes.

Latent variable methods can unite several types of data, often from disparate sources, and produce a comprehensive model that describes many facets of a product or family of products. Often, only a few designed experiments are needed to augment data that already exists in various production or laboratory databases and spreadsheets. The resulting model can be used in an optimization framework to target specific, desired properties for a new or modified product. Chapter 2 describes the general procedure and tools required for rapid product development: PLS regression, DOE in the latent variable space, and optimization.

Chapter 3 demonstrates the application of these methods to the reformulation of frozen muffin batters, in a real-time collaboration with an industrial partner. The goal of this reformulation is to minimize a specific quality attribute which is poorly understood and for which there is no first principles model. Any impact on other product properties such as appearance and taste must be minimized. Initially, baseline data is used to build a latent

variable model that identifies key ingredients and areas for further experimentation. The use of ingredient properties as additional model variables is explored and experimentation in the latent variable space is demonstrated. Model inversion via constrained optimization generates modified muffin recipes which are evaluated against their corresponding original products, both for the quality attribute being studied and for appearance and taste.

Chapter 4 describes how process monitoring is impractical in oat milling due to the manual classification required to assess a key quality metric, which is the number of intact oats or hulls in the finished product stream. There is a tradeoff to be made between yield and quality, but more frequent monitoring is needed to achieve the ideal balance between the two.

The case study assesses the feasibility of a machine vision solution for the classification task. Colour imaging is explored briefly but the main focus is NIR imaging. Unsupervised classification (PCA) and supervised classification (PLS-DA) of the NIR images is demonstrated, and the PLS-DA model is validated using samples of three different cultivars of oats.

Chapter 2 Rapid Reformulation Using Latent Variable Methods

Your customers want something new.
The government has new rules too.
How to make them salivate?
In a word, reformulate.

This chapter introduces the techniques involved in rapid reformulation using latent variable methods. The goal of these techniques is to achieve a product having specific final properties by using existing data and executing a minimum number of additional experiments. The target product could be a polymer blend, a high performance polymeric coating, a food item, or any other product involving many raw materials in its formulation. The same procedure can be employed to develop new products, but to be consistent with Chapter 3 the term reformulation is used throughout Chapter 2.

Reformulation for the purposes of this thesis is defined as the process of modifying specific properties of a product while maintaining some defining features of the original product. By this definition, reformulation is an activity common to many industries and may be driven by consumer demand, government regulation, or corporate objectives such as cost containment or environmental impact. These drivers change often so the ability to bring improved products to market quickly is a significant economic advantage.

The complexity of product reformulation arises from the fact that there are usually many competing goals and many potential solutions to investigate. “In the context of healthier food choices, food reformulation might be defined as reformulating existing foods to remove (e.g. trans fatty acids) or reduce (e.g. sugars, saturated fat, salt) certain food components while maintaining characteristics such as flavour, texture, and shelf life.” (van Raaij, Hendriksen and Verhagen 2008). Such a reformulation involves choosing among many potential ingredients, recipes and process conditions. Satisfying all of the reformulation objectives can be a daunting task when only the traditional methods of experimentation are considered.

In this thesis, the term recipe is synonymous with 'ingredient proportions' and describes the weight percentages of all of the ingredients in a product. The term formula encompasses the recipe and process conditions for a given product.

This chapter demonstrates how latent variable methods can handle the features of a reformulation problem (competing goals, many potential solutions) and provide a direct path to successful new formulas. In general, there are four phases in the rapid reformulation process; model building, augmentation of the model with designed experiments, optimization, and testing. Interaction with the model is necessary in every phase, therefore PLS, the modeling technique, is discussed first. The rest of the chapter proceeds with the four phases of rapid reformulation, in sequential order.

2.1 PLS Models

PLS is the methodology employed to build models suitable for rapid product development. (PLS is also useful in many other applications). The PLS acronym has two meanings: projection to latent structures and partial least squares. Projection to latent structures describes the concept behind PLS modeling; that is, taking many non-independent variables and projecting them down into a smaller (lower dimensional) latent variable space. Each latent variable is a linear combination of the original variables, and all of the latent variables are orthogonal to each other. PLS's other meaning, partial least squares, refers to the method for calculating model parameters. This brief introduction will focus on the conceptual aspects of PLS models and their interpretation. (Wold, Sjostrom and Eriksson 2001) is a good reference that contains more of the mathematical and algorithmic details.

As a regression method, the power of PLS stems from the fact that it models the underlying structure of two matrices **X** and **Y**, as well as the relationship between them. This means that new values for either **X** or **Y** can be tested for their consistency with past observations, and can be used to predict new values for **Y** or **X**, respectively. The matrix form of the equations that describe the projections of **X** and **Y** into latent variable space are

$$\begin{aligned} \mathbf{X} &= \mathbf{T} \cdot \mathbf{P}^{\mathrm{T}} + \mathbf{E} \\ \mathbf{Y} &= \mathbf{T} \cdot \mathbf{C}^{\mathrm{T}} + \mathbf{F} \end{aligned} \qquad \textbf{2.1}$$

where **T** contains the latent variables, $\mathbf{P}^{\mathrm{T}}$ and $\mathbf{C}^{\mathrm{T}}$ relate the latent variables to the original variables in **X** and **Y**, and **E** and **F** are residual matrices.

Each column of **T** is a latent variable; a direction in latent variable space. The values in **T** are called scores, but they can be thought of as coordinates. Each row of **T** contains the coordinates of an observation's location in latent variable space. The values in **T** are calculated by an iterative method that also produces a matrix $\mathbf{W}^{*}$, such that

$$\mathbf{T} = \mathbf{X} \cdot \mathbf{W}^{*} \qquad \textbf{2.2}$$

where $\mathbf{W}^{*}$ contains weights for projecting observations from the higher dimensional **X**-space to the lower dimensional **T-**space, or latent variable space. This projection (the values in $\mathbf{W}^{*}$) takes into account the correlation structure of both **X** and **Y**.

The first latent variable, $\mathbf{t}_1$ (the first column of **T**) is the direction explaining the largest covariance between **X** and **Y**; subsequent latent variables have less importance. For this

reason, it is sometimes sufficient to show just the first two latent variable dimensions in a score plot. A score plot is a two-dimensional window into the latent variable space; it describes the distribution of observations in a plane of the model.

Figure 2.1 is an example of a score plot, taken from a model describing the denseness and moistness of different types of cakes. This model has only two dimensions; Figure 2.1 shows the score values in $\mathbf{t}_1$ plotted against the score values in $\mathbf{t}_2$. The model was built using **X**-data from (Joachim, Schloss and Handel 2008), shown in Table 2.1. The **X**-data describes the approximate percentages by weight of ingredients required to make each type of cake. The **Y**-data are numerical interpretations of the descriptive cake characteristics given in the source. Notice how the cakes are distributed in the latent variable space. Cake types that are more similar, such as Sponge and Chiffon, fall closer together while cakes of very different textures are spread apart, such as Angel Food and Pound cake.

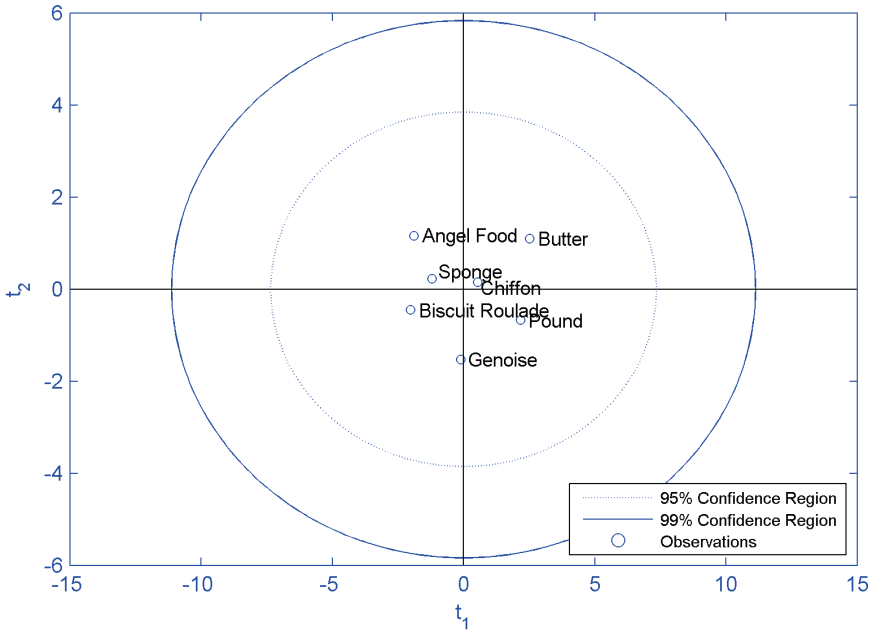

Figure 2.1 Score plot for example cake model, showing both latent variables plotted against one another

		X-data (% by weight)					**Y**-data	
	Cake Type	**Liquid**	**Egg**	**Flour**	**Sugar**	**Fat**	**Denseness**	**Moistness**
Training Set	Angel Food	6	47	13	34	0	-1	0
	Sponge	4	45	20	31	0	-1	0
	Biscuit Roulade	0	59	14	27	0	-1	-1
	Genoese	0	46	23	23	8	0	-1
	Chiffon	14	35	18	24	9	-1	1
	Pound	12	22	22	22	22	1	1
	Butter	24	10	27	27	12	0	1

Table 2.1 Data for example cake model (Joachim, Schloss and Handel 2008)

The **P** and **C** in equation 2.1 are called loadings and the $\mathbf{W}^*$ in equation 2.2 are called weights; they relate the original variables to the latent variables. In other words, they contain the values that translate each observation into the latent variable space. In interpreting a PLS model, $\mathbf{W}^*$and **C** are often visualized as bar plots, which illustrate how much each original variable contributes to a given latent variable. Figure 2.2 shows the loadings for the first dimension, or component, ($\mathbf{t}_1$) of the example cake model. It is evident that Egg and Fat contribute more to the first component than the other variables in **X**; a cake recipe with a relatively high amount of Fat and low amount of Egg will have a high value of $\mathbf{t}_1$. Denseness and Moistness both have positive coefficients for the first latent variable; so a cake that is moist and dense relative to other cakes in the data set will also have a relatively high value of $\mathbf{t}_1$.

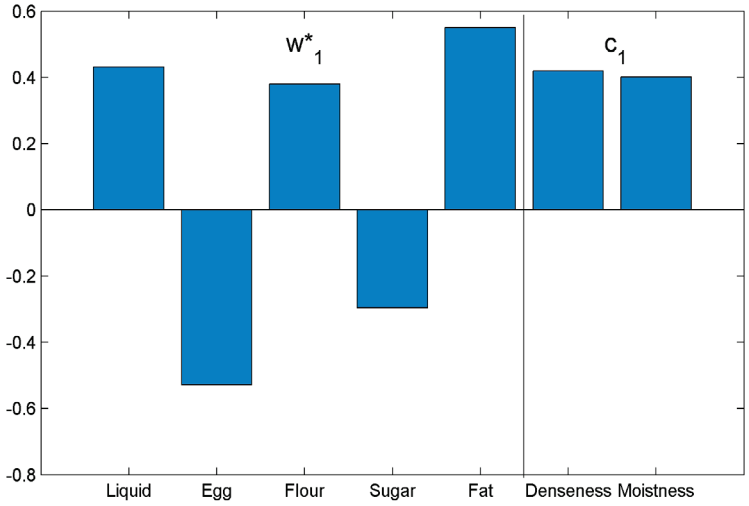

Figure 2.2 Loading plot for the first dimension of the example cake model

The vectors of weights and loadings can also be visualized in pairs, as a scatterplot. When overlaid with a score plot, they offer a compelling visual representation of why certain observations fall where they do in the latent variable space. This is called a loading biplot. Correlated variables are located near to each other and in the same direction from the origin. Figure 2.3 is a biplot of $\mathbf{w}^*_1$ and $\mathbf{c}_1$ versus $\mathbf{w}^*_2$ and $\mathbf{c}_2$ superimposed on the score plot from Figure 2.1. It shows that the **Y**-variable Moistness is correlated with the **X**-variable Liquid, which makes sense. Both are negatively correlated with the **X**-variable Egg, located diagonally across the origin. Looking at the relative positions of the observations, Butter cake and Pound cake are both more moist and dense than the other cakes, with Butter cake tending towards Moistness and Pound cake tending towards Denseness. The **Y**-variable Denseness is correlated with the **X**-variables Flour and Fat.

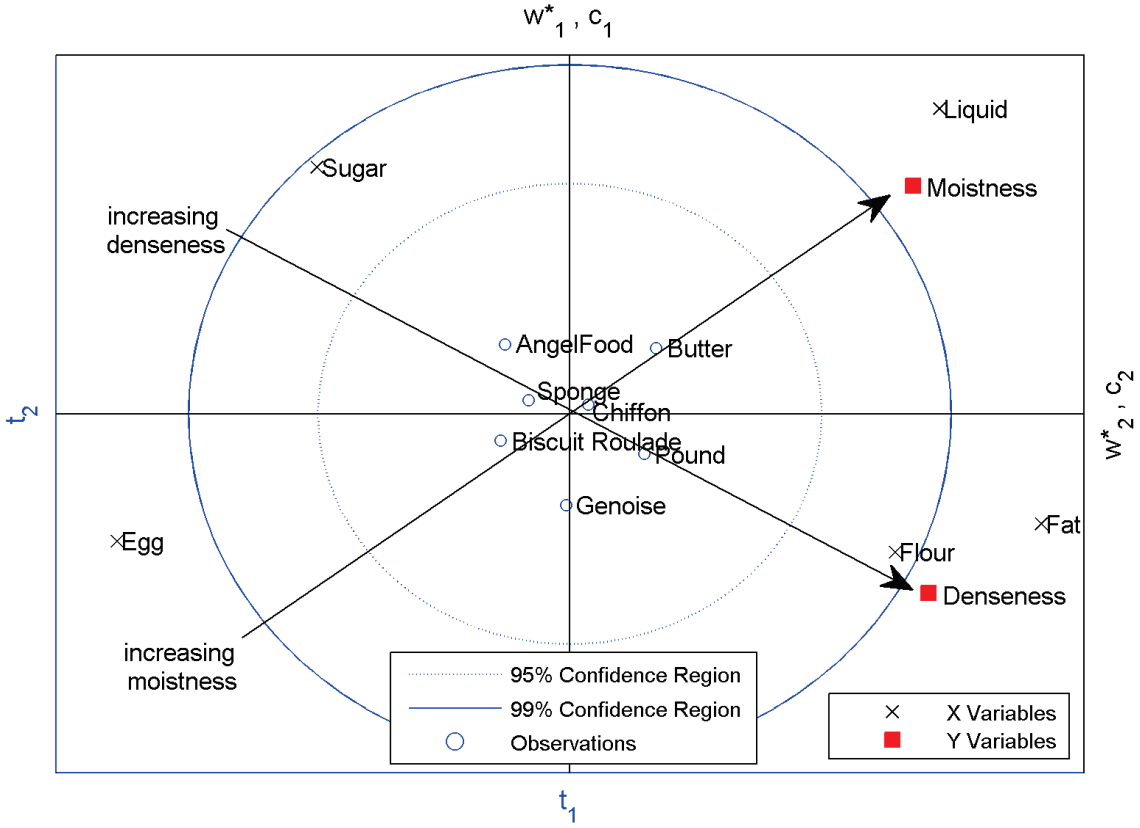

Figure 2.3 Loading biplot for example cake model

2.1.1 Quality of Fit and Quality of Prediction

One important consideration is the number of latent variables that should be included in a PLS model, in other words, the number of independent directions in a data set. This is judged based on two measures, R^2Y and Q^2Y. R^2Y measures the quality of fit of the model, that is, the fraction of the sum of squares of all of the **Y**-variables that is explained by the model. This number increases as more components are added. Q^2Y measures the quality of prediction of the model, and is calculated by cross-validation, detailed in Appendix to Chapter 3: Cross-Validation. Unlike R^2Y, it usually plateaus and then decreases if too many components are added. In model building, the goal is to choose a number of components that strikes a balance between the highest values of R^2Y and Q^2Y. Figure 2.4 displays the values of R^2Y and Q^2Y for the example cake model. It also displays R^2X, which is the fraction of the sum of squares of all of the **X**-variables that is explained by the model. R^2X becomes important when inverting a model to predict new values of **X** for desired values of **Y**. If R^2X is low, then the correlation structure in **X** is inadequately modeled, and the results of the inversion are less likely to be successful.

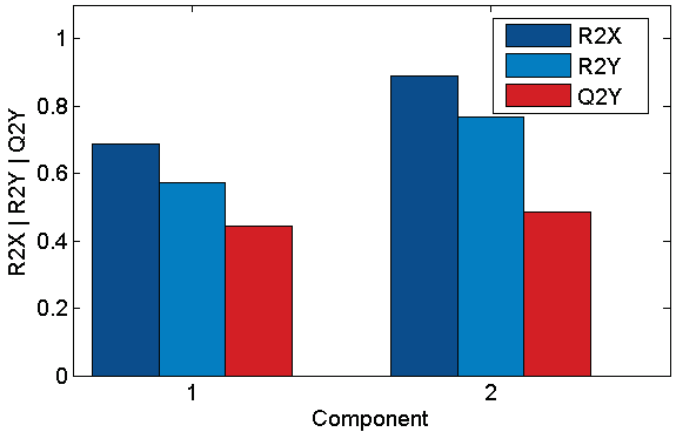

Figure 2.4 Model summary plot for example cake model

2.1.2 Assessing New Observations

As mentioned earlier, one important property of PLS is that new observations can be assessed as to their consistency with past data. For a new observation, $\mathbf{x}_{\text{new}}$, the latent variable scores are found by the following

$$\boldsymbol{\tau}_{\text{new}}^{\text{T}} = \mathbf{x}_{\text{new}}^{\text{T}} \cdot \mathbf{W}^{*} \qquad \textbf{2.3}$$

where $\boldsymbol{\tau}_{\text{new}}^{\text{T}}$ is a vector containing the coordinates of the new point in the latent variable space. It can be thought of as a new row in the $\mathbf{T}$ matrix.

In a score plot, the location of new observations indicates their relationship to past data points. The score plot shows whether new observations belong to existing clusters of points or not, and whether they are within the model's confidence ellipses. As examples of new observations, a pound cake recipe and three variations, plus a pastry recipe from (Fance 1966) were used as a prediction set for the cake model. The data is shown in Table 2.2. Figure 2.5 shows that Fance's pound cake recipe and its three variations all fall near the Pound cake in the training set which makes sense; they are all variations on a pound cake. Fance predicts that Variation #1 will be similar to a Genoese due to the additional Egg and Flour. The model confirms this hypothesis, as Variation #1 falls near the Genoese from the training set. The pastry falls much farther away, beyond the 95% confidence ellipse, which is not surprising because it is not a cake.

		X-data (% by weight)					**Y**-data	
	Cake Type	**Liquid**	**Egg**	**Flour**	**Sugar**	**Fat**	**Denseness**	**Moistness**
Training Set	Angel Food	6	47	13	34	0	-1	0
	Sponge	4	45	20	31	0	-1	0
	Biscuit Roulade	0	59	14	27	0	-1	-1
	Genoise	0	46	23	23	8	0	-1
	Chiffon	14	35	18	24	9	-1	1
	Pound	12	22	22	22	22	1	1
	Butter	24	10	27	27	12	0	1
Prediction Set	Pound (Fance)	5	25	25	25	20		
	Variation #1 (Fance)	4.4	27.8	27.8	22.2	17.8		
	Variation #2 (Fance)	17.6	16.3	32.7	20.4	13.1		
	Variation #3 (Fance)	22	12.5	37.5	20	8		
	Pastry (Fance)	5.4	8	52	13	21.6		

Table 2.2 Data for example cake model (Joachim, Schloss and Handel 2008), including prediction set (Fance 1966)[1]

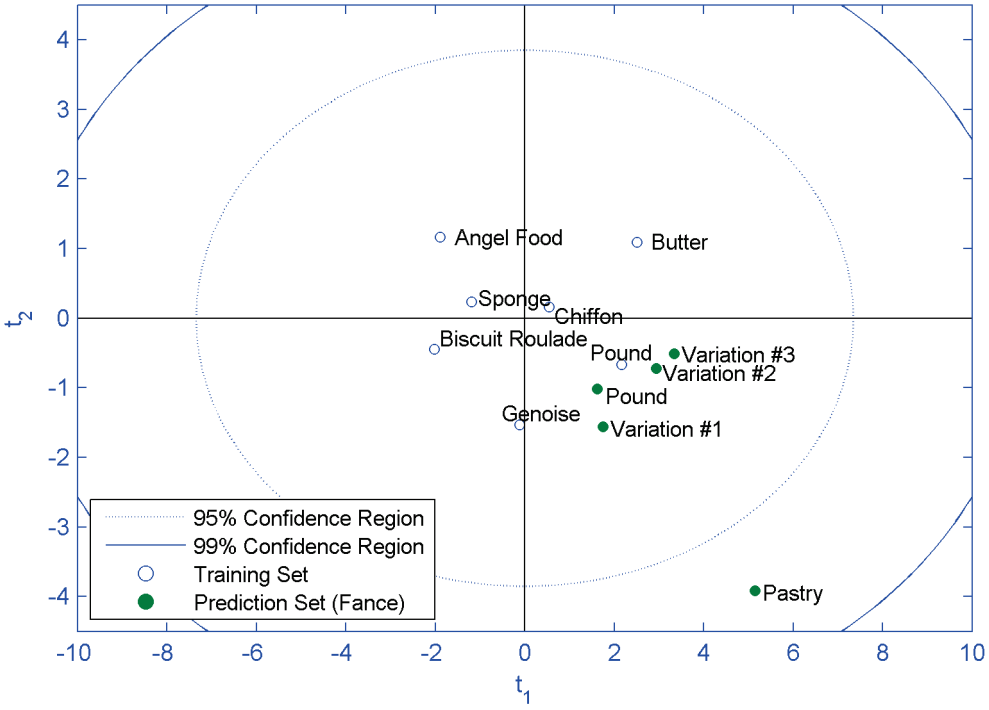

Figure 2.5 Score plot for example cake model, with new observations

Comparison of a new point with past data can also be made by calculating two specific parameters, Hotelling's T-squared (T^2) and the squared prediction error (SPE). Conceptually, T^2 measures an observation's distance from the centre of the latent variable

[1] Fance's recipes have been adjusted to match the format in (Joachim, Schloss and Handel 2008), i.e. that the moisture in the fat is reported as liquid. The adjustment assumes that the fat contains 20% moisture.

space. If it is very large, then that observation is beyond the range of the initial data. T^2 is calculated as

$$T^2_{\text{new}} = \sum_{a=1}^{A} \frac{\tau^2_{\text{new,a}}}{s^2_a} \quad \textbf{2.4}$$

where A is the number of latent variables in the model, and s^2_a is the variance of the a^{th} latent variable. The example cake model has only two latent variables so Figure 2.5 gives a good indication as to each observation's T^2 value. The ellipses are the 95% and 99% confidence limits for T^2. However, T^2 is the distance to the origin over all of the latent variable dimensions; therefore when there are more than two dimensions a different plot is needed. Figure 2.6 shows a plot of T^2 for the training observations and the prediction set (Fance's recipes).

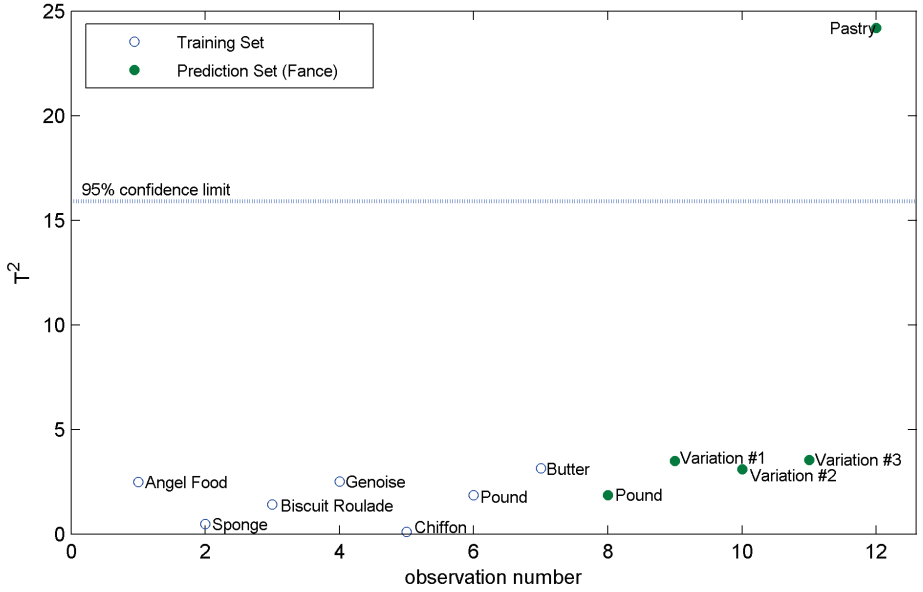

Figure 2.6 Hotelling's T^2 for the example cake model

SPE measures the perpendicular distance between $\mathbf{x}_{\text{new}}$, which is the new observation's actual location in its original, higher-dimensional space and $\hat{\mathbf{x}}_{\text{new}}$, which is its location on the latent variable plane as predicted by the model. A large SPE means that the new observation falls well off the plane that defines the model, meaning that it does not adhere to the correlation structure of past data. After calculating $\boldsymbol{\tau}^{\text{T}}_{\text{new}}$ in equation 2.3, $\hat{\mathbf{x}}^{\text{T}}_{\text{new}}$ is calculated as

$$\hat{\mathbf{x}}^{\text{T}}_{\text{new}} = \boldsymbol{\tau}^{\text{T}}_{\text{new}} \cdot \mathbf{P}^{\text{T}} \quad \textbf{2.5}$$

and SPE is calculated as

$$\mathrm{SPE}_{\mathbf{x}_{\mathrm{new}}} = \sum_{k=1}^{K} (\hat{\mathbf{x}}_{\mathrm{new}} - \mathbf{x}_{\mathrm{new}})^2 \qquad \textbf{2.6}$$

where K is the number of **X**-variables in the model. Figure 2.7 shows that Fance's Pound cake and Variation #1 are the most similar to the training data in terms of its correlation structure. Variation #2 and Variation #3 are both off the model plane for the same reason; they have more Flour and less Fat than the training data on average. This can be determined by looking at a contribution plot, not shown. The pastry recipe has an extremely high SPE; again this makes sense as it is not a cake. The model may not predict **Y**-values accurately for observations with SPE values beyond the confidence limits. An accurate prediction is even less likely for observations with high SPE and high T^2, such as the pastry recipe.

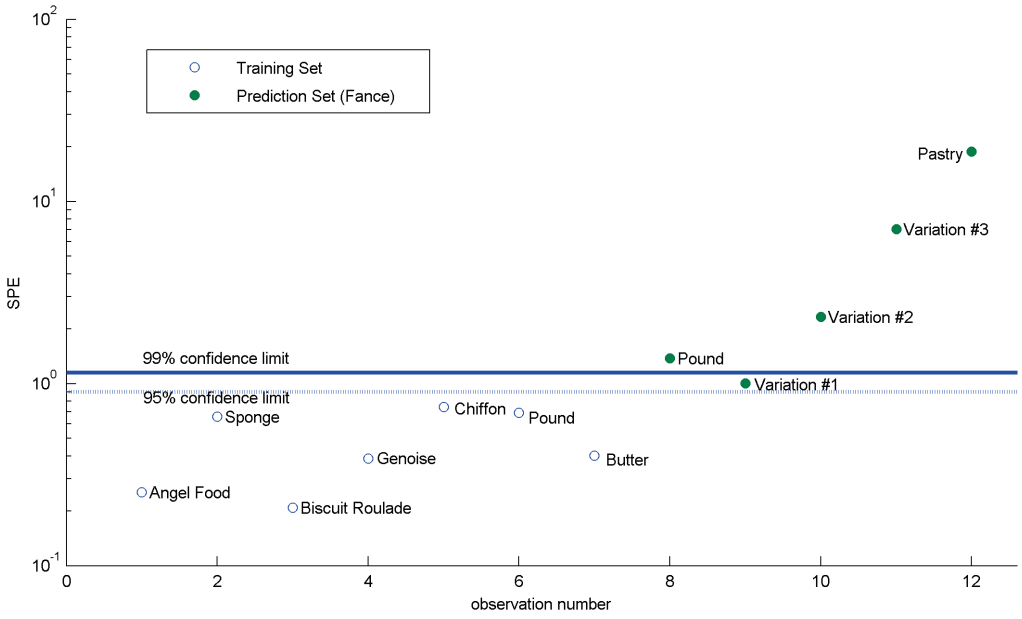

Figure 2.7 SPE for the example cake model

If T^2 and SPE are acceptable (i.e. below the 95% or 99% confidence limits), then the predicted outcomes of $\hat{\mathbf{x}}_{\mathrm{new}}$, called $\hat{\mathbf{y}}_{\mathrm{new}}$, can be calculated as

$$\begin{aligned} \hat{\mathbf{y}}_{\mathrm{new}} &= \boldsymbol{\tau}_{\mathrm{new}}^{\mathrm{T}} \cdot \mathbf{C}^{\mathrm{T}} \\ &= \hat{\mathbf{x}}_{\mathrm{new}}^{\mathrm{T}} \cdot \mathbf{W}^{*} \cdot \mathbf{C}^{\mathrm{T}} \\ &= \hat{\mathbf{x}}_{\mathrm{new}}^{\mathrm{T}} \cdot \boldsymbol{\beta} \end{aligned} \qquad \textbf{2.7}$$

where $\boldsymbol{\beta}$ contains the PLS regression coefficients. The coefficients are not usually used for predictions directly, but are useful to view as a bar plot because it illustrates which **X**-

variables influence a particular **Y**-variable. The coefficients for Moistness and Denseness are shown in Figure 2.8. The confidence intervals are calculated by jackknifing during the cross-validation step, as described in (Martens and Martens 2000). Cross validation is covered in Appendix to Chapter 3: Cross-Validation.

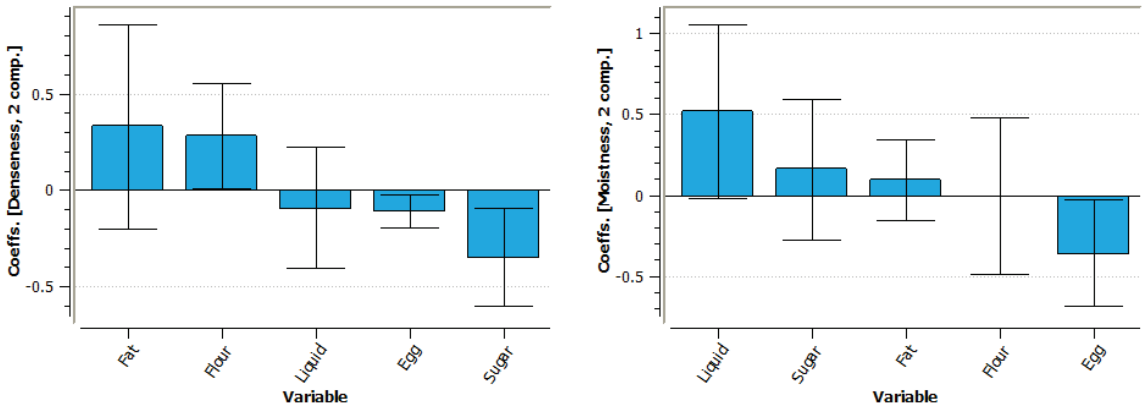

Figure 2.8 PLS coefficients for Denseness (left) and Moistness (right) for the example cake model

These are the main concepts of PLS as required for the purposes of rapid product development. PLS has been covered extensively in the literature; one very accessible source is (Umetrics AB 2006)[2]. (Wold, Sjostrom and Eriksson 2001) offers a succinct introduction and mathematical summary.

[2]Freely available online at http://books.google.ca/books?id=B-1NNMLLoo8C

2.2 Model Building

The procedure for rapid product reformulation could be applied to any mixture product. The defining feature of a mixture product is that its properties are "assumed to depend only on the proportions of the ingredients present in the mixture and not on the amount of the mixture" (Cornell 2002). In addition, the ingredient proportions are expressed as percentages, and naturally, the sum of the ingredient proportions must be 100%. (Kettaneh-Wold 1992) calls this a 'true' collinearity and goes on to explain that there can be many other 'near' collinearities in data due to other constraints on ingredient proportions. 'Near' collinearities in a food product model can also arise from the fact that some flavours go well together (e.g. chocolate and bananas) while others are not commonly put together (e.g. chocolate and carrots).

Mixture Models

Historically, mixture models were considered a class unto themselves, requiring special techniques to account for the lack of independence in the mixture data.(Cornell 2002) is a good reference for these techniques. (Kettaneh-Wold 1992) demonstrates that PLS is appropriate for mixture modeling because it can handle the collinearities inherent in many practical mixture applications and because process variables can be included in the same model. That is, a PLS model can be built using process conditions and ingredient proportions as **X**-data and any measured product properties as **Y**-data. The data structures used in a PLS mixture model are shown in Figure 2.9. For each formula, **Z** contains the process conditions, **R** contains the recipe, and **Y** contains the final ingredient properties. **Z** and **R** together are referred to as **X**-blocks, or **X**-data.

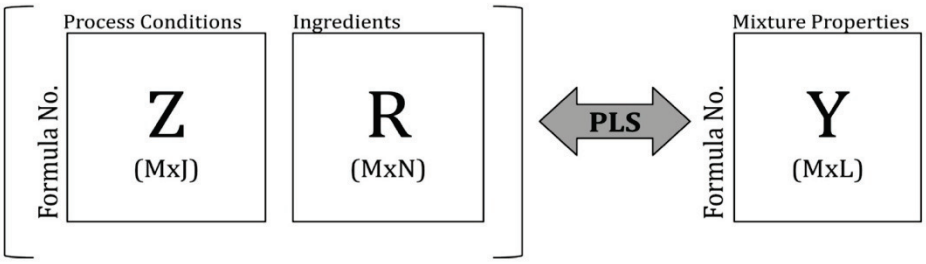

Figure 2.9 Data structures for PLS mixture models

Mixture-Property Models

(Muteki, MacGregor and Ueda 2006) introduced the concept of mixture-property PLS models and showed that they are more powerful than PLS models built using mixture data alone. Whereas a mixture model describes the relationship between recipes and final product properties, a mixture-property model describes the relationship between the properties of the ingredients and the properties of the final product. There are several benefits of mixture-property models as compared to mixture models; they include (1) the discovery of which material properties affect which final product properties; (2) better estimates of the final product properties; (3) estimates of product property differences that would result from the use of a new ingredient, even if that ingredient has never been used in previous production or experiments; (4) the potential for experimental designs based on ingredient properties and ingredient proportions; and (5) the potential to simultaneously determine which ingredients to include and in what proportions to blend them, using optimization techniques.

Figure 2.10 shows the matrices required to build a mixture-property model. The values in **R**, **Z** and **Y** may come from laboratory experiments, pilot-plant trials, or data collected during regular production. **D** (not shown) is a database of ingredient properties for all available ingredients. $\mathbf{D}_R$ is the subset of **D** that pertains to the ingredients in **R**. The values in **D** may be provided by suppliers or measured in-house, and may not necessarily contain values of all properties for every ingredient. Missing ingredient property data is addressed in (Muteki, MacGregor and Ueda 2005).

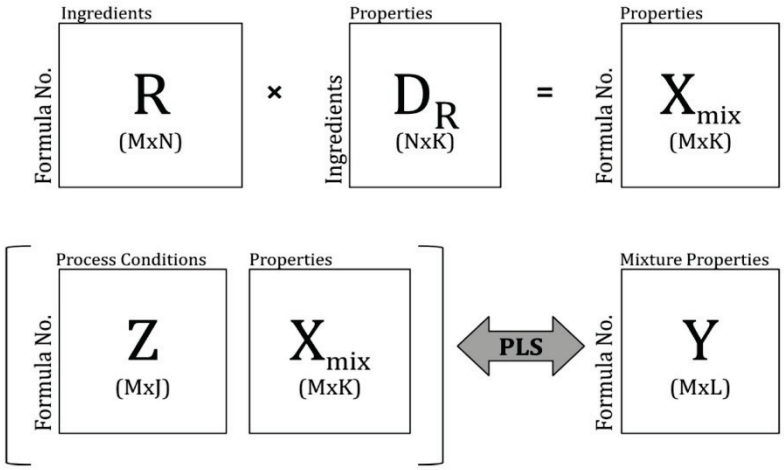

Figure 2.10 Data structures for PLS mixture-property models

Two assumptions must be made in order to use the data structures as shown. First, the ideal mixing rule (Grassmann 1971) must be approximately valid for the ingredients and properties contained in **D**. To illustrate the concept, consider the amount of iron in a food product. Suppose the product has only two ingredients, and that the recipe is 30% ingredient A and 70% ingredient B (by weight). Then by the ideal mixing rule,

$$\frac{\text{mg iron}}{\text{100g final product}} = 30\% \cdot \frac{\text{mg iron}}{\text{100g ingredient A}} + 70\% \cdot \frac{\text{mg iron}}{\text{100g ingredient B}} \quad \textbf{2.8}$$

Other mixing rules can be used if they are known (Muteki, MacGregor and Ueda 2006).

The second assumption is that the properties available in **D** must correlate well with the properties being measured on the final product, otherwise a mixture-property model based on those properties cannot be expected to be superior to a mixture model.

2.3 Augmenting a Model with Designed Experiments

In univariate statistics, it is accepted that designed experiments, where experimental factors are varied together in a prescribed manner, is far superior to one-variable-at-a-time experimentation (Box, Hunter and Hunter 2005). "Design of experiments can also be seen as a tool to make data balanced and representative for a given system or process."(Wold, Josefson, et al. 2004).

The same logic applies to latent variable spaces, i.e. it is necessary to vary more than one latent variable at a time to achieve a good model. Because the latent variable directions are independent, as opposed to the original variables which may be correlated, the latent variable space is the only space in which a truly orthogonal design can be accomplished. Each latent variable is a linear combination of the original variables, therefore choosing levels of latent variables for a factorial experiment corresponds to moving several of the original variables up and down together.

Whether a mixture model or a mixture-property model is being used, the model may need to be augmented with designed experiments. The need for designed experiments is highly application-dependent, based on how much initial data is available, and how well that data is distributed in the latent variable space. If there are holes in the latent variable space, i.e. regions where there are no observations, then it will be beneficial to execute designed experiments located in those regions. Similarly, if there is a high concentration of data points in a region, it may be beneficial to select just a few of those data points for inclusion in the model, to ensure that areas with less data will be adequately represented in the model. In Figure 2.1 (page 6), the observations are clustered near the centre; this model will be more widely applicable if additional points are added further from the origin in both dimensions. In Figure 3.11 (page 38), there are some obvious empty areas where additional points could be added.

The need for designed experiments also depends on whether the goal is to build a general predictive model over the range of the existing data, or to target a product with specific final properties. If the former, experiments should be added as required to balance the existing data as explained above. If the goal is to target specific final properties, then one can check the feasibility of the desired values of those properties using PCA.

2.3.1 Feasibility of the Desired Product Properties

(Jaeckle and MacGregor 1998) explain that because some of the variables in **Y**, the matrix of final product properties, may be correlated, the desired values for each **Y**-variable cannot necessarily be chosen independently. In other words, the desired values must match the correlation structure of the past **Y**-data. To evaluate this, a principle components analysis (PCA) model can be built using the **Y**-data, which will determine how many independent dimensions are spanned by **Y**. PCA is a projection method similar to PLS, but it applies to just one block of data; in this case, the **Y**-block. It describes the correlation structure found in the **Y**-block, and therefore enables the testing of new observations (i.e. desired values) as to whether or not they conform to the correlation structure of past observations. PCA is briefly described in section 4.3, or the reader can refer to (Umetrics AB 2006).

The desired properties $\mathbf{y}_{\text{des}}^{\text{T}}$ can be projected into the PCA model as

$$\boldsymbol{\tau}_{\text{PCA}}^{\text{T}} = \mathbf{y}_{\text{des}}^{\text{T}} \cdot \mathbf{P}_{\text{PCA}} \qquad \textbf{2.9}$$

where **P** is the loadings matrix that relates the original **Y**-data to the PCA model space and $\boldsymbol{\tau}_{\text{PCA}}^{\text{T}}$ contains the score values, or coordinates, of the new point in PCA model space that corresponds to $\mathbf{y}_{\text{des}}^{\text{T}}$. T^2 for the new point can then be calculated from equation 2.4, and compared with the T^2 of the data used to build the model.

Similarly to equation 2.5, $\hat{\mathbf{y}}_{\text{PCA}}^{\text{T}}$ can be calculated as

$$\hat{\mathbf{y}}_{\text{PCA}}^{\text{T}} = \boldsymbol{\tau}_{\text{PCA}}^{\text{T}} \cdot \mathbf{P}_{\text{PCA}}^{\text{T}} \qquad \textbf{2.10}$$

and similarly to equation 2.6, the distance between $\mathbf{y}_{\text{des}}^{\text{T}}$and $\hat{\mathbf{y}}_{\text{PCA}}^{\text{T}}$ can be calculated as

$$\text{SPE}_{\mathbf{y}_{\text{des}}^{\text{T}}} = \sum_{\text{k=1}}^{\text{K}} \left(\hat{\mathbf{y}}_{\text{PCA}}^{\text{T}} - \mathbf{y}_{\text{des}}^{\text{T}}\right)^2 \qquad \textbf{2.11}$$

If T^2 and SPE value are small relative to the values calculated for the data used to build the model, then the desired final properties are feasible according to the PCA model on **Y**. These requirements are discussed in (Kourti and MacGregor 1996).

Suppose that a PCA model of one component is calculated on the **Y**-data for the example cake model outlined in Table 2.1. The resulting R^2 is 67% and Q^2 is 41%. Because there is only one component, this suggests that the two properties Moistness and Denseness are not independent, and therefore it is not possible to choose desired values for both properties.

Further suppose that there are two new types of desired cakes, with the properties shown in Table 2.3.

	Denseness	Moistness
$\mathbf{y}_{des1}$	1	0
$\mathbf{y}_{des2}$	1	-1

Table 2.3 Desired characteristics for two new types of cakes

These are the only two combinations of **Y**-values that do not appear in the **Y**-data used to build the model. Their values of SPE and T^2 are shown in the plot below. Notice that a cake which is dense and neither moist nor dry ($\mathbf{y}_{des1}$) falls just above the 95% confidence limit on SPE, meaning that it may be marginally feasible. A cake which is dense and also dry ($\mathbf{y}_{des2}$) falls far above the 99% confidence limit; it is infeasible according to this PCA model.

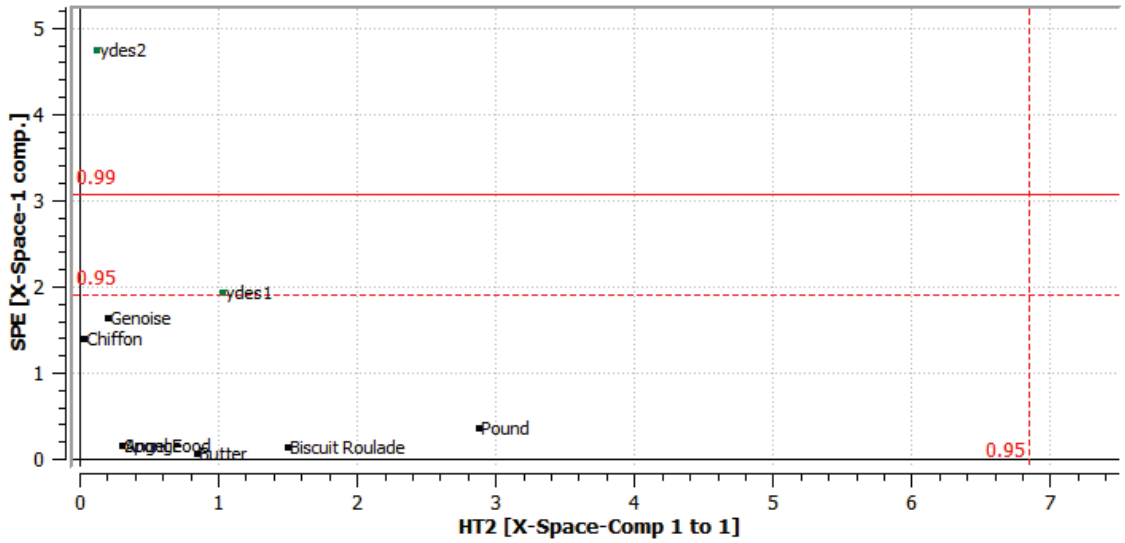

Figure 2.11 SPE vs T² for example cake model training set and prediction set

If $\mathbf{y}_{des}^{T}$ is judged to be infeasible according to the PCA model, then a sequential DOE can be undertaken, following the methodology of (Muteki and MacGregor 2007). This is an iterative process, where the model is updated with the new DOE points and the feasibility of the desired final properties is re-checked against the updated model in each iteration. As the iterations progress, the preferred outcome is that the desired properties become more feasible; i.e. T^2 and SPE are reduced. If however, T^2 and/or SPE remain large after several iterations, then the desired final properties may be unachievable using the current ingredients and process (Muteki and MacGregor 2007).

At the point where one has achieved a model with balanced and representative data, and according to which the desired final properties are feasible, one can proceed with model inversion.

2.4 Model Inversion and Optimization

PLS models are often used for assessing new observations and predicting their outcomes as described in section 2.1. In product development, one wishes to do the opposite, that is, specify the desired outcomes (product properties) and then invert the model to find the right ingredients, recipes and process conditions to produce such a product.

In terms of the relative numbers of variables in **X** and **Y**, (Jaeckle and MacGregor 1998) describe the three situations which may be encountered, and how each situation can be handled with regards to model inversion. The most common situation is that the number of product properties (**Y**-variables) in the PLS model is less than the total number of **X**-variables. In other words, there are some degrees of freedom in the inversion, and the number of solutions is infinite.

Optimization is used to choose the best solution from among the many possibilities, and there are several considerations that limit the choice of solutions. For example, SPE and T^2 should be less than their respective 95% or 99% confidence limits to ensure that the chosen **X**-values obey the range and correlation structure of past data. Preference is usually given to lower-cost solutions and to formulas with less complexity (i.e. fewer ingredients). These requirements can be accounted for in the optimization problem either as soft constraints (penalty terms in the objective function) or as separate hard constraints.

(Muteki, MacGregor and Ueda 2006) demonstrated the effectiveness of an optimization formulation (for a case study in which process conditions were constant) as shown in equation 2.12. Also refer back to Figure 2.10 (page 15) which displays the data structures for a mixture-property model. The objective function seeks a recipe, $\mathbf{r}_{\text{new}}$, that has final product properties as close as possible to the desired values $\mathbf{y}_{\text{des}}$. The properties can be given unequal weights if some are more important than others; these weights are stored on the diagonal in $\mathbf{W}_1$. The objective is restrained by two penalty terms, one for the cost of the recipe, and another for the number of ingredients used. w_2 and w_3 are the weights for cost and complexity respectively.

$$\min_{\mathbf{r}_{new}} \left(\mathbf{y}_{des} - \boldsymbol{\beta}_{PLS}^{T}\mathbf{x}_{mixnew}\right)^{T} \cdot \mathbf{W}_1 \cdot \left(\mathbf{y}_{des} - \boldsymbol{\beta}_{PLS}^{T}\mathbf{x}_{mixnew}\right) + w_2 \sum_{j=1}^{NN} \mathbf{r}_{new,j} \cdot c_j + w_3 \sum_{j=1}^{NN} \delta_j$$

such that:

Ideal Mixing Rule	$\mathbf{x}_{mixnew} = \mathbf{r}_{new}^{T} \cdot \mathbf{D}$
PLS Model Constraints	$SPE_{new} = \sum_{k=1}^{K} (\mathbf{x}_{mixnew} - \hat{\mathbf{x}}_{mixnew})^2 \leq \epsilon$ $T_{new}^2 = \sum_{a=1}^{A} \frac{\boldsymbol{\tau}_{new,a}^2}{s_a^2} \leq T_{max}^2$
Mixture Constraint	$\sum_{j=1}^{NN} \mathbf{r}_{new,j} = 1, \qquad 0 \leq \mathbf{r}_{new,j} \leq 1$
Binary Variable Constraint	$\delta_j = \begin{cases} 1, & \mathbf{r}_{new,j} < 0 \\ 0, & \mathbf{r}_{new,j} = 0 \end{cases}, \mathbf{r}_{new,j} \leq M_j \cdot \delta_j$

2.12

N.B. NN is the total number of ingredients listed in the ingredient property database, **D**
M_j is an upper limit on the amount of ingredient j in the final product.
A is the number of latent variables in the PLS model
K is the number of **X**-space variables

In this formulation, the hard constraints include the ideal mixing rule (Grassmann 1971), the PLS model constraints, the mixture constraint, and the binary variable constraint. The ideal mixing rule enforces the relationship between a recipe and its ingredient properties, the PLS model constraints ensure that the new recipe is consistent with the data used to build the model, and the mixture constraint ensures that the new ingredient proportions sum to 100%. The binary variable constraint works together with the third term in the objective function. δ_j is set to one if an ingredient is used in $\mathbf{r}_{new}$ and zero if it is not, and the sum of these values (the number of ingredients used in $\mathbf{r}_{new}$) is penalized in the objective function.

This formulation is flexible; for example, cost, SPE, or T^2 may appear in the objective function or as constraints depending on the specific goals of the project at hand. (Garcia-Muñoz, et al. 2006) further discuss the case where multiple solutions exist, including methods for the selection of the most desirable solution.

*Lack of data in the **Y**-space*

Ideally, data is available for all of the desired final product properties, and they are all included in the PLS model and optimization. When this is not the case, then the PLS model is only taking into account a subset of the desired properties, that is, the model isn't as 'smart' and the SPE and T^2 terms may not do enough to ensure that optimization results are practical. Extra constraints may be added to the optimization, such as specific min/max constraints for each ingredient.

If the lack of measured **Y**-variables is severe, a practitioner may choose to make 'model-informed' recipes rather than using a formal optimization. In other words, one may choose to make specific formula modifications based on the knowledge gleaned from the model as to which ingredients advance and hinder the reformulation goals.

Although these two situations are not desirable from a modeling perspective, they have been overcome in the Chapter 3 case study, and will be discussed in sections 3.5 and 3.6.

Chapter 3 Rapid Reformulation of Frozen Muffin Batters

Would you, could you, minimize
An unknown trait called AOI?
How could you find an answer, quick?
PLS should do the trick.

This chapter describes the application of rapid reformulation techniques to a product line of frozen muffin batters for the food service industry.

Initially, prepared muffins for each formula were analyzed for a specific quality attribute, which is referred to throughout this thesis as the Attribute of Interest (AOI). The AOI is an analytically measured quantity. A reduction in AOI was desired for several of the formulas, but a first-principles model was not available for this attribute.

In this case study, latent variable models are used in discovering which variables are related to high values of AOI and how AOI can be minimized. Designed experiments in the latent variable space are used to augment existing data, and optimization based on the latent variable model is used to generate reformulated recipes.

Baseline Data Set

Some existing data, referred to as the baseline data set, was available from the industrial collaborator at the outset of the project. The baseline data set contained eight replicate observations of AOI in 26 different muffin formulas (the **Y**-data), as well as the recipes and process conditions (the **X**-data) for those formulas. Actual weights of ingredients used were unknown; only the proportions dictated by the recipe were given. (Independence of these observations will be discussed in section 3.2 and in Appendix to Chapter 3: Cross-Validation). Figure 3.1 shows the ranges of AOI for each muffin formula. It illustrates that the variation in AOI is proportional to its magnitude, which is why it makes sense to use a logarithmic transformation (discussed in section 3.2) for this data.

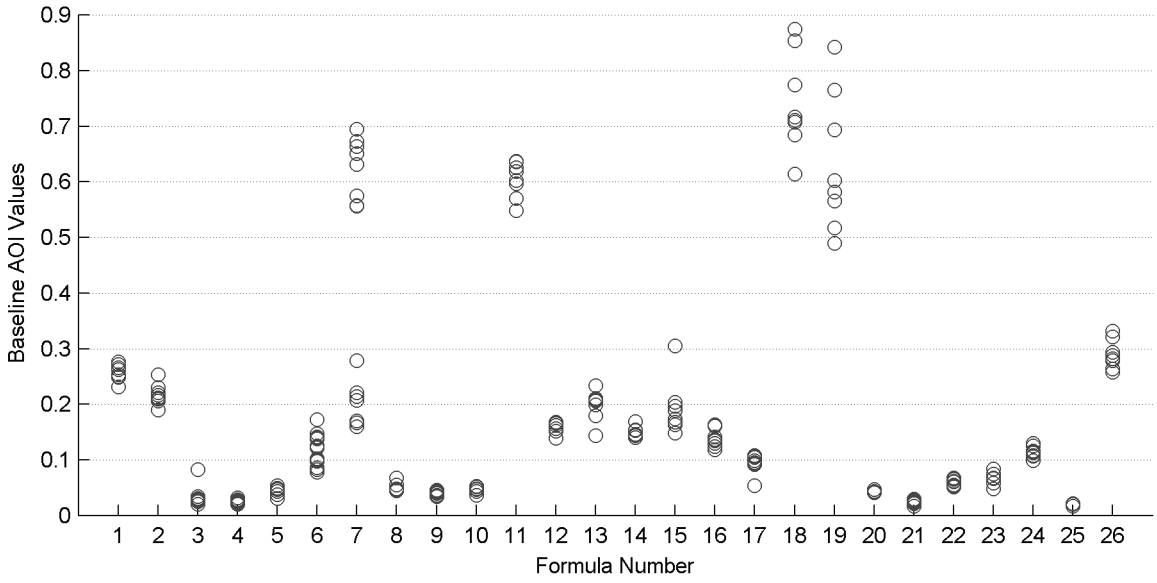

Figure 3.1 Baseline AOI values

Reformulation Constraints

Constraints on the modified formulas include taste, texture, aesthetics, functionality, nutritional properties, and cost. A reformulated product must be as similar as possible to the original in terms of taste and texture. The batter must also rise sufficiently and uniformly to produce an aesthetically pleasing muffin shape; muffin caps that are somewhat oblong are considered unacceptable. Once thawed, a pail of batter may be used over a period of several days, and must maintain its functionality through this period. Nutritional properties and cost are fairly simple constraints; they can be calculated from the recipe. Nutritional constraints vary by formula; minimum values for dietary fibre is an example. Added cost, if required, must be minimized.

Potential Solutions

Three types of modifications were considered. The process conditions could be changed, a New Ingredient E (NIE) could be added, or the proportions of existing ingredients (i.e. the recipe) could be adjusted. At the outset of the project it was unknown whether one or more of these would provide the best solution.

Some experimentation was completed on process conditions but for strategic business reasons, changing the process conditions was determined to be an impractical solution. Early testing incorporated both the NIE and recipe proportions as experimental factors together in the same experiments, but no interactions were discovered. The NIE was found

to be so effective that it dwarfed the ingredient effects. For this reason and for business reasons as well, it made sense to pursue NIE testing separately from recipe adjustment.

The addition of NIE is appealing because NIE is a minor ingredient, so there is minimal change to the recipe. The baseline data set contained no observations containing NIE, so its effect on AOI could not be determined until some experimentation was undertaken. In initial experiments, the deviation from original taste, texture and appearance was shown to be small, and AOI was significantly reduced. Addition of NIE therefore seemed to be an excellent reformulation option. However, there are specific business advantages to finding a solution without introducing a new ingredient.

Modifying a recipe carries the risk of changing the taste, texture or appearance of the current product, and with more than 50 different ingredients there are many potential combinations to explore. The pursuit of a solution by adjusting recipe proportions effectively illustrates rapid reformulation techniques and will be the main focus of this chapter. The baseline data set provided sufficient data to build an initial PLS model, using the recipes as **X-**data and the baseline AOI values as **Y-**data. Section 3.2 describes this model.

Experimentation on the NIE solution and the recipe solution proceeded in parallel, results of both types of experiments were added to the data set as they became available, and the model was repeatedly updated. Over time, the number of variables also increased to account for additional sources of variation. Therefore this chapter contains snapshots of the continually changing model at different points in time.

Chapter Organization

To protect the confidential nature of the project, observation and variable names are coded, and some variable values will be transformed for the purpose of plotting. Naming conventions are discussed in section 3.1. Section 3.2 describes a mixture model using the baseline data only. Section 3.3 illustrates the differences between experiments designed in **X**-space and experiments designed in latent variable space. In section 3.4, the effect of adding ingredient properties to the model is shown. Sections 3.5 and 3.6 discuss the optimization formulation and results. Finally, aspects of cross-validation as it applies to this case study are addressed in Appendix to Chapter 3: Cross-Validation.

3.1 Naming Conventions

Observation Naming

In this case study, an observation is batch of muffin batter that was mixed, baked and analyzed for AOI. Although each batch yields multiple muffins, there is only one observation per batch; a composite sample is sent for analysis. Each observation is primarily identified by a unique observation ID number, prefixed by "Obs".

Each observation belongs to a formula group; formula numbers 1 through 26 are the original formulas present in the baseline data. Recipe variations are grouped with their original formula even though their actual **X-**data is different from the rest of the group. Over the course of this project, a separate research effort resulted in the development of formulas for new muffin flavours; they were included in the model as they became available and new formula groups were defined to accommodate them.

Each observation also belongs to an experiment group; that is, a group of observations that were performed during the same time period using the same lots of ingredients. Experiment groups are denoted by the prefix "Expt".

Variable Naming

The **Y**-variable in this case study is referred to as the Attribute of Interest (AOI). The **X**-variables are not referred to by name; instead they are grouped loosely by ingredient type, and their code names consist of a prefix and a number. Table 3.1 shows the groups and prefixes. New Ingredient E is simply referred to as NIE.

Variable Name Prefix	Variable Group
Spice	Spices & Flavours
Leaven	Leaveners
Dairy	Dairy & Eggs
VegFru	Vegetables & Fruits
Grain	Grains,Flours & Starches
Misc	Other
Proc	Process Conditions
Nutrit	Nutritional Properties

Table 3.1 Prefixes for variable names

As mentioned earlier, varying the process conditions is not a practical solution from a business standpoint; however there are some process condition variables in the data set. Each reformulated product will use the same process conditions as its original formula.

3.2 Baseline Mixture Model

The goals of this model were to (1) fit a PLS model to the baseline data, (2) determine the number of latent variables needed to describe the baseline data, (3) discover how well a PLS model would predict AOI values, based on cross-validation, and (4) identify ingredients positively and negatively correlated with AOI.

Baseline Data Set

The baseline data set contained 26 muffin formulas, with a total of 58 ingredients and 1 process variable; this data was used as the **X**-data, and its structure can be seen in Figure 3.2. Gray squares indicate that an ingredient is present in a formula and the white area represents zeros, indicating the absence of an ingredient. Note that there are several ingredients used in only one formula. The **Y**-data contained four values; the AOI and three others. The three others did not exhibit any correlation with AOI and were therefore excluded from the model.

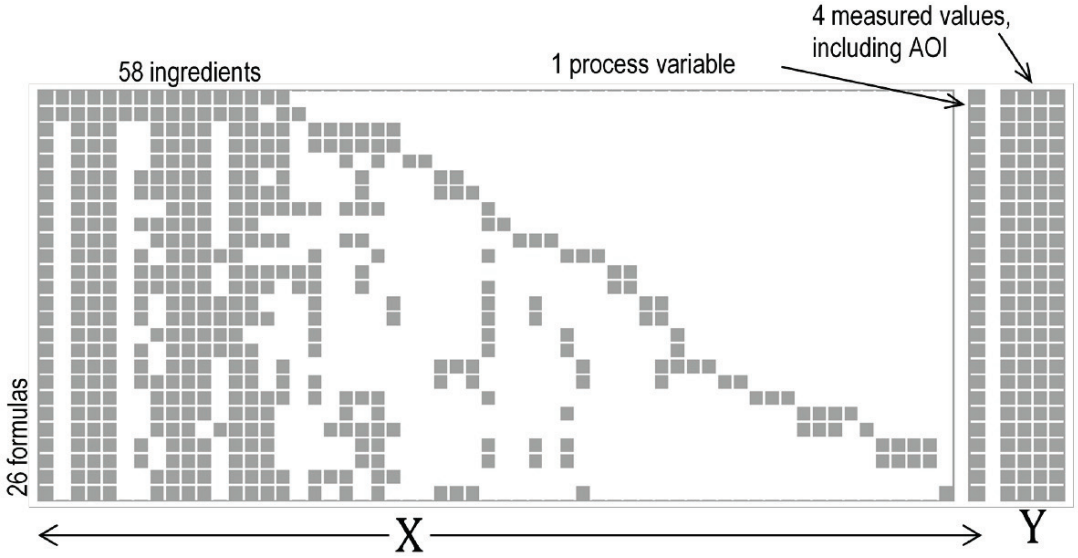

Figure 3.2 Matrices used in the baseline recipe model

The baseline **X** and **Y** matrices were much taller than shown; for each formula, there were at least eight replicate observations. Samples were taken in the manufacturing plant, during regular production. Within a formula group, the observations were taken far enough apart to each be from a unique batch. Within some formula groups, all observations were taken during the same production run, whereas other formula groups have observations across two or three different production runs. The within-group variance for each formula is underestimated, because the ingredient lots are common within a production run. As an

example, see Formula 7 in Figure 3.3, which exhibited a large variation between production runs.

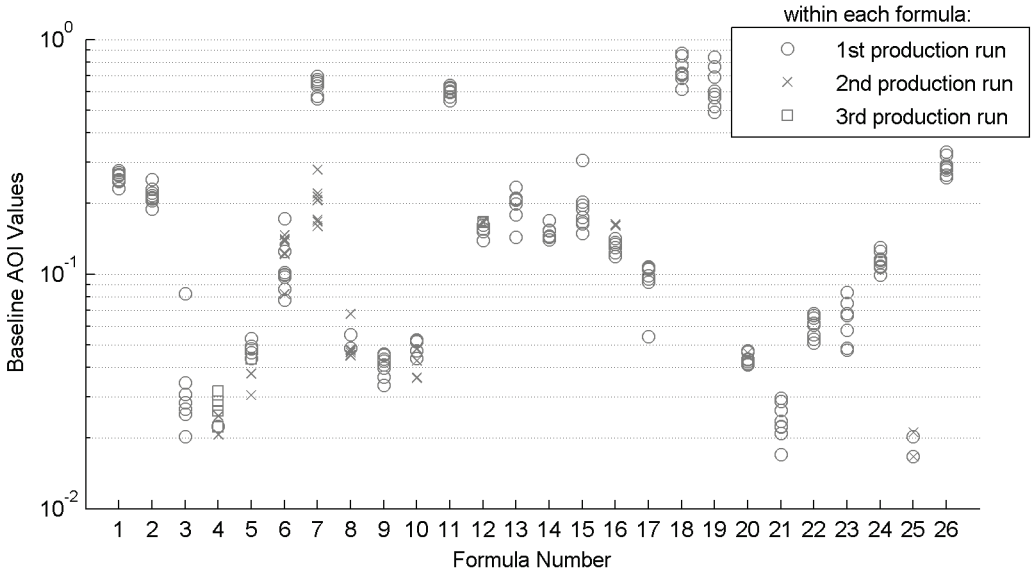

Figure 3.3 Baseline values of AOI for each of the 26 original formulas

Data Preprocessing & Model Building

All data was mean-centred and scaled to unit variance, and a logarithmic transformation was applied to AOI. Using a logarithmic transformation essentially models the percentage changes in the AOI rather than the absolute changes. This improves linearity in the model, and achieves a more constant variance within the formula groups as shown in Figure 3.4.

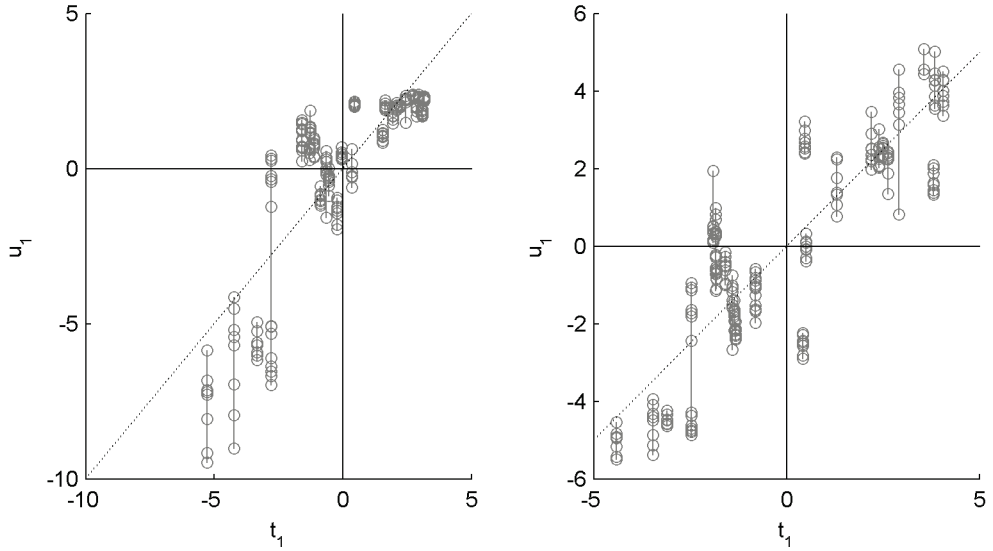

Figure 3.4 Score plots for a model without a log transform on AOI (left) and with a log transform on AOI (right) Each group of replicate observations is shown with a line connecting all points in the group.

Some ingredients were only used in 1 of the 26 formulas and were excluded from the model until further experimentation, leaving 43 variables to be included in the baseline model. Because the **X**-data was identical for all observations of a given formula, and because all of the replicates were not truly independent, cross-validation was done by formula groups. Appendix to Chapter 3: Cross-Validation contains further discussion on cross-validation for this type of data.

Model Results

The model built from the baseline data has 14 components, with very good quality of fit (R^2X=85%, R^2Y=96%) and quality of prediction (Q^2Y=94%). The component summary is shown in Figure 3.5, page 33 (top left). The number of components was decided by cross-validation. Model building was completed using ProSensus MultiVariate[3], in which by default, a component is deemed significant if it contributes at least 1% to total Q^2Y or if it contributes at least 5% Q^2Y for any one **Y**-variable. Since there is only 1 **Y**-variable in this model, the threshold is 1%. Following this rule, the fourth component is insignificant and the auto-fit function stops at three components, however, more can be added manually. Although the fourth component is insignificant as a contributor to Q^2Y, it contributes 2.7% to R^2Y and 5% to R^2X. The 5th through 14th components are again significant according to the cross-validation rules. The model summary plot shows these values graphically; see Figure 3.5 (top left). The resulting 14-component model exhibits excellent predictive ability as shown in the observed vs predicted plot in Figure 3.5 (top right).

The biplot shown in the centre of in Figure 3.5 contains a loading plot superimposed on a score plot for the first two components. The score plot (blue points, axes, and confidence ellipses) shows the distribution of observations in the first two model dimensions, along with the corresponding confidence regions. Observations located in close proximity to each other indicate that their formulas are similar. The nature of the similarity is illustrated by the loading plot (black points and labels), superimposed. It shows the weights of the **X**-variables for the first two model dimensions. Leaven2 and Leaven3 have high weights in the first model dimension; they are located farthest from the origin. Observations to the right of the origin will tend to have more Leaven3 and less Leaven2, and the opposite is true for

[3] Supplier: ProSensus, Inc.

observations to the left of the origin. An observation located at the origin would have average values of every ingredient.

The location of the red square shows the weights of the **Y**-variable, AOI. Its location relative to the **X**-variables is meaningful, for example, AOI is positively correlated with Spice7, Misc10, and Leaven2, located nearby, and negatively correlated with VegFru7 and Dairy1, located diagonally across the origin. Because there are 14 dimensions in the model, there are many combinations of latent variables that can be shown as biplots. Since the first two components explain more of the covariance between X and Y than any other two components, they give the best window into the latent variable space and so are shown in Figure 3.5. However, this plot can be slightly misleading because there are 12 more components to consider.

To look at the total influence of each variable over all 14 components, consider the coefficients plot in Figure 3.5 (bottom). This plot shows the magnitude and direction of each **X**-variable's correlation to the **Y**-variable AOI, and their confidence intervals. Note that the magnitude of the coefficients is largest for those variables that appear near the extremes of the biplot. Also note that the location of an **X**-variable relative to AOI in the biplot is an indicator of both the magnitude and direction of its correlation with AOI shown in the coefficients plot. For example, Spice7 and Misc10 are nearby to AOI and in much the same direction from the origin as AOI, and they both have large positive coefficients. In contrast, VegFru7 and Dairy1 are located diagonally across the origin from AOI, and they have large negative coefficients. There is not an exact correlation between a variable's location on the biplot and on the coefficients plot because the coefficients plot takes into account all 14 model dimensions.

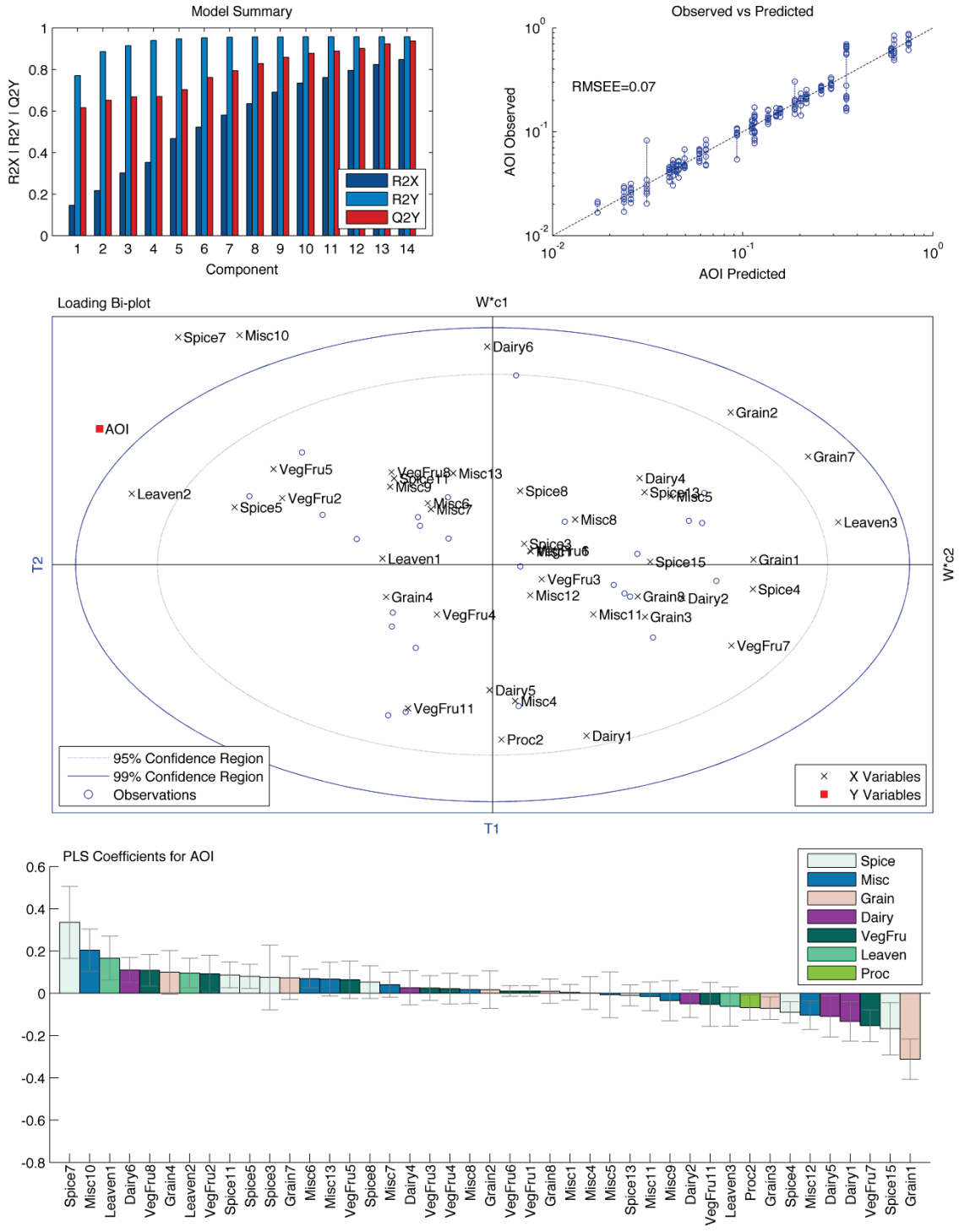

Figure 3.5 PLS plots for the baseline mixture model

Standard Deviation Model

A second model was built using the same **X**-data but using the (log-transformed) standard deviation of AOI per flavour group as the **Y**-variable. Some of the key variables as identified by the baseline model are also key contributors to the variation in AOI. Figure 3.6 displays the model summary plot and observed vs predicted plot for this model. Because there is only one latent variable, the biplot is omitted. The coefficients plot displays the coefficients for AOI (left) side-by-side with the coefficients for standard deviation (right). Most of the variables influence AOI and the standard deviation of AOI in the same direction. Variables that have a large positive or a large negative coefficient for both, are the best ingredients to decrease and increase, respectively, to achieve the goals of this project.

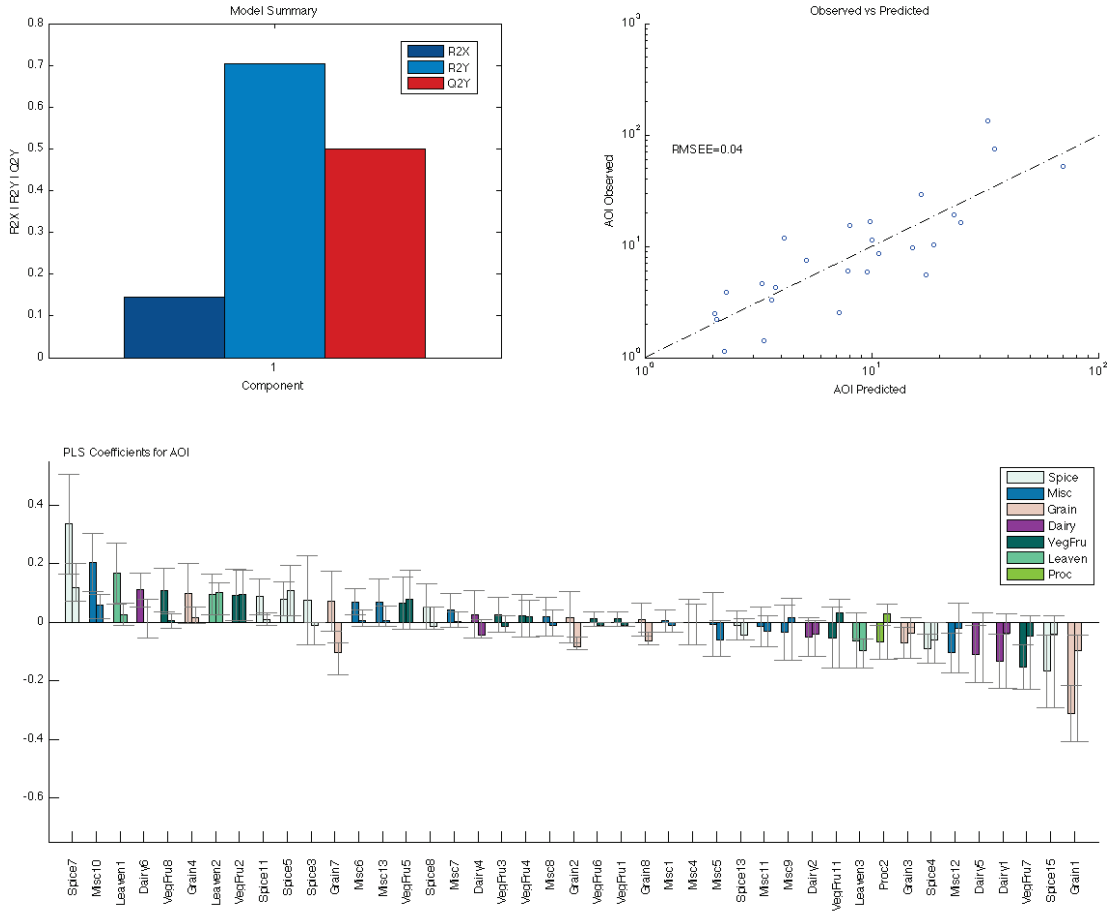

Figure 3.6 PLS plots, standard deviation model

3.3 Experimental Design Spaces

Section 2.3 discussed the importance of considering the latent variable space when designing experiments; that is, using the latent variables as experimental factors. This abstract concept is illustrated here by two examples related to the case study of frozen muffin batters. The first is a traditional two-level full factorial in the **X-**space (i.e. the factors are ingredients) and the second is a design in the latent variable space (i.e. the factors are the latent variables of the PLS model).

3.3.1 DOE in X-space

Following the development of the baseline AOI model as presented in section 3.2, a traditional full factorial design was used to test the effects of three ingredients on Formula11. Misc10, Leaven1 and Leaven2 were the chosen ingredients, all of which had large positive AOI coefficients in the baseline model. For each factor, its proportion in the original recipe (per Formula 11) was used as the high level, with half that amount used as the low level. Note that the latent variable model includes recipes with even lower levels of the three factors and several recipes with no Leaven2 or Misc10 at all. The ingredients chosen as factors are fairly minor (<5% by weight); therefore the rest of the recipe was left unchanged. Three replicates were completed due to expected batch to batch variability.

Results

Figure 3.7 illustrates that the results agree with a general trend seen in figure 3.1, namely that recipes with low levels of AOI also tend to have lower variances. It also shows the coefficients of Misc10, Leaven1 and Leaven2, calculated by an ordinary least squares regression using the DOE data. Misc10 and Leaven1 were found to have significant positive coefficients for AOI and Misc10 is the largest of the three effects; these results agree with the PLS model. Contradicting the PLS model, Leaven2 was not shown to be significant, although two interaction terms involving Leaven2 were found to be significant.

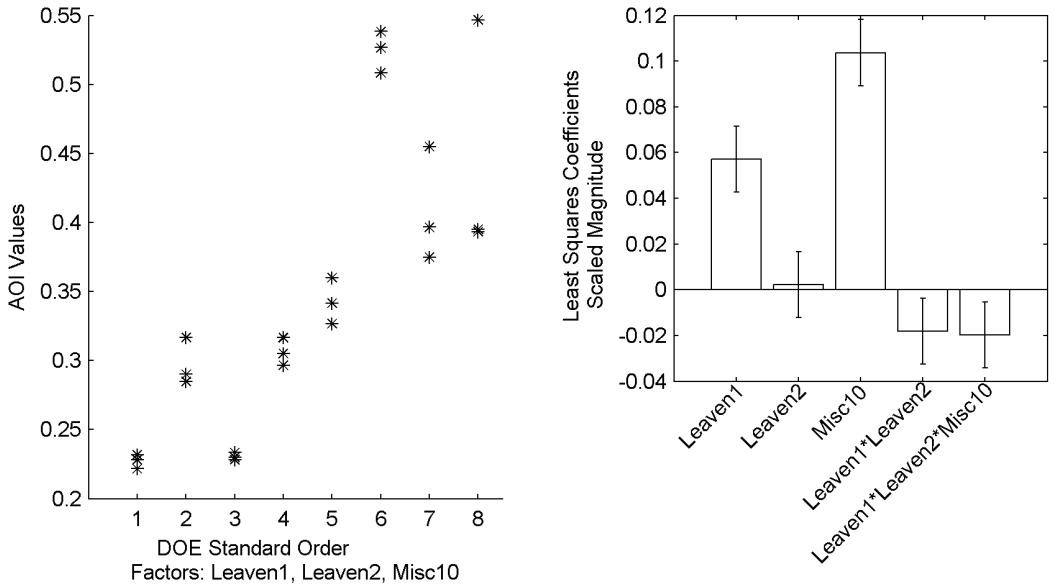

Figure 3.7 Results of X-space DOE on Formula 11: AOI Values (left) and least squares coefficients (right)

Figure 3.8 shows how the factorial DOE loses its orthogonality when projected into the latent variable space of the baseline PLS model. The distances between the two levels of Leaven1 and Leaven2 are very small when seen in the context of the latent variable space containing all of the muffin formulas.

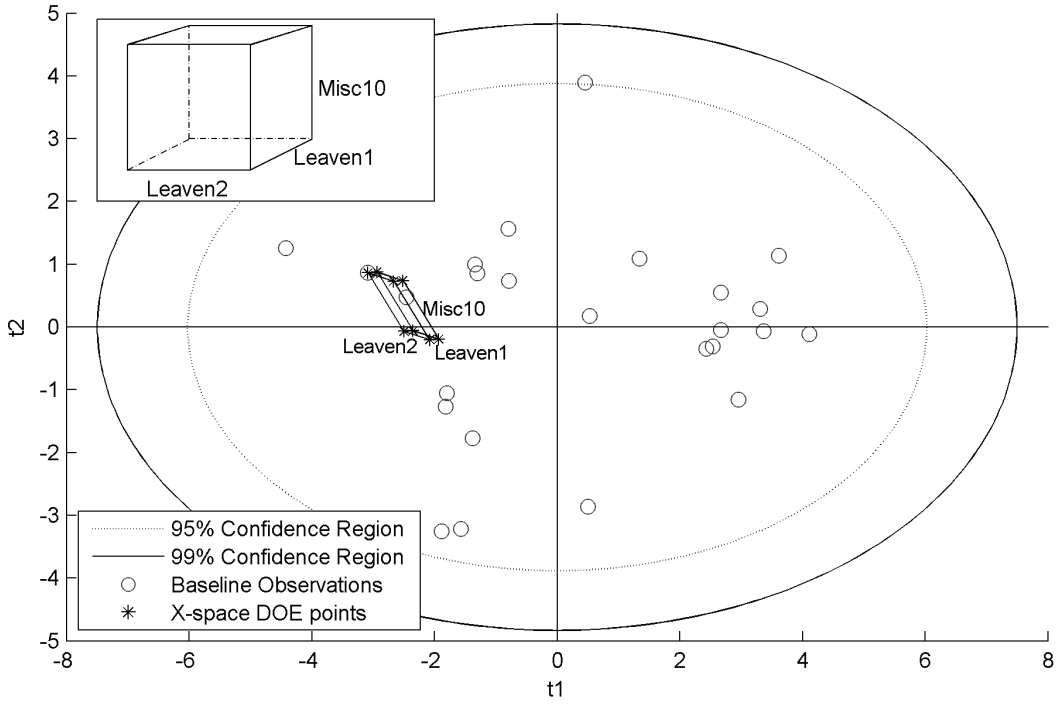

Figure 3.8 The X-space DOE on Formula 11 as seen in the first two latent variable dimensions, as compared to the cubic representation of a full factorial DOE in three factors (inset).

Figure 3.9 shows that all of the experimental points were quite close to the model plane and within the 95% confidence region in terms of distance from the centre of the model.

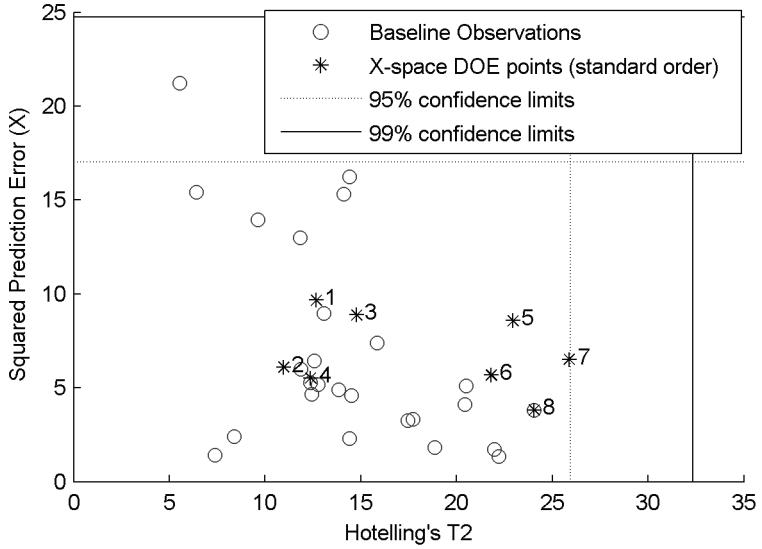

Figure 3.9 SPE vs T^2 for baseline observations and X-space DOE points (labels coincide with Figure 3.8)

Figure 3.10 shows the results of the DOE trials against their values as predicted by the baseline PLS model. Recall that the baseline model does not contain interaction terms. The two-way interaction that was found to be significant in this DOE (Leaven1*Leaven2) was added to the latent variable model but was found to be insignificant and therefore it was removed.

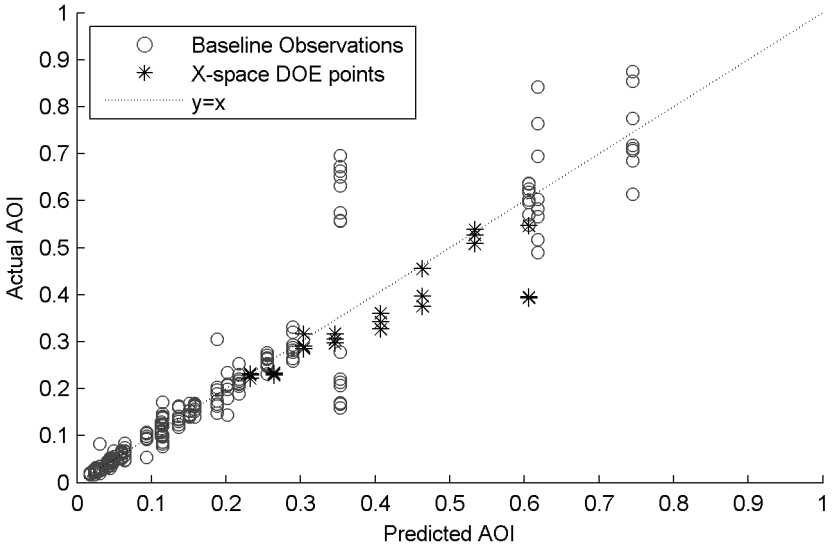

Figure 3.10 Observed vs Predicted for X-space DOE points

Although the results of this experiment on Formula 11 seem to contradict the results of the baseline PLS model in terms of the significance of Leaven2, the PLS model is more widely

applicable because it includes all of the baseline muffin formulas. The replacement of Leaven2 with Leaven3 was later found to be effective in reducing AOI values, and will be discussed in section 3.6.

3.3.2 DOE in Latent Variable Space

Later in the project, further experiments had been performed and ingredient properties had been added. (Section 3.4 discusses the inclusion of ingredient properties in detail.) The number of variables had increased to 111, and the number of observations was 166. Some of the baseline data had been excluded due to concerns over its lack of independence[4]. The updated model had 8 components, with a total R^2Y of 92%, and a total Q^2Y of 75%. The first three components captured most of the variation, contributing 82% to R^2Y and 60% to Q^2Y.

One result of the inclusion of new data and assumptions was that the observations were less evenly distributed across the latent variable space. They became more clustered, leaving some 'holes', which are especially apparent in the $\mathbf{t}_1$-$\mathbf{t}_2$ plot in Figure 3.11. Before performing an optimization, it is beneficial to explore those areas to understand whether recipes located there are practical and so that the model will be representative of the entire space.

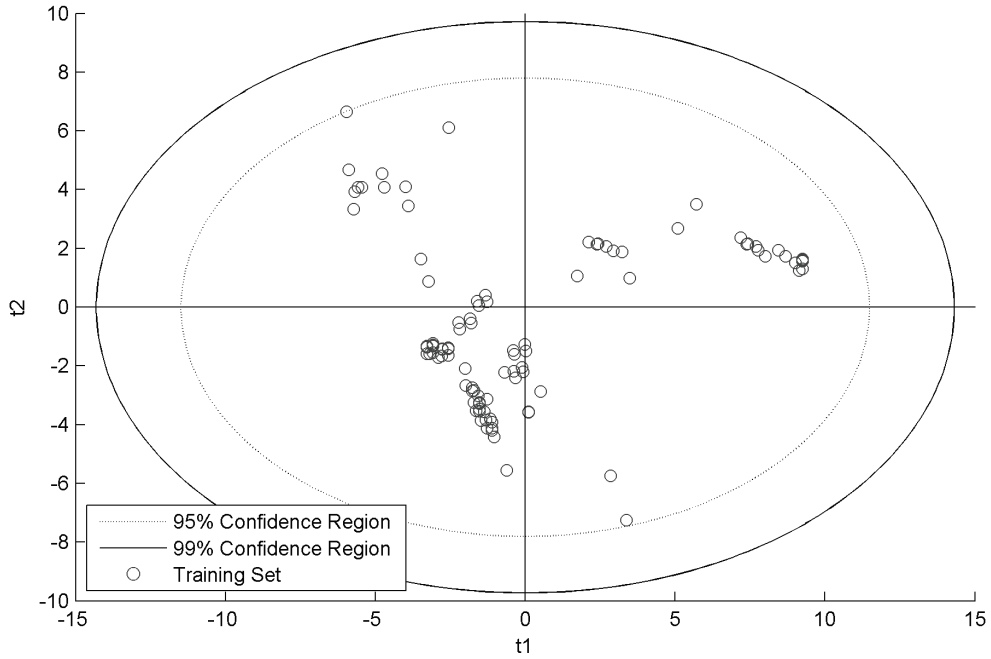

Figure 3.11 Score plot of the first two components in the updated model

[4] Excluded data was re-included (just prior to the optimization step) after clarification was received.

First, an approximate two-level full factorial was executed in the latent variable space. Because they accounted for most of the variation in the model, the first three components were chosen as the experimental factors. The high and low levels were chosen such that they would span most of the space inside the 95% confidence region. Recipes at each location were calculated via optimization. The optimization was configured to select a recipe whose SPE (distance from the model plane) would be at a minimum, and whose **T**-scores would fall within a small bounding box drawn at each experimental design point. There were no ingredient constraints used, nor any constraints on the number of ingredients used, except that the sum of the ingredients must equal 100%. As a result, the ingredients were present in combinations not seen before in the data set; i.e. flavour combinations that would not necessarily be desirable. The locations of the latent variable ('LV') space experimental design points are shown in Figure 3.12.

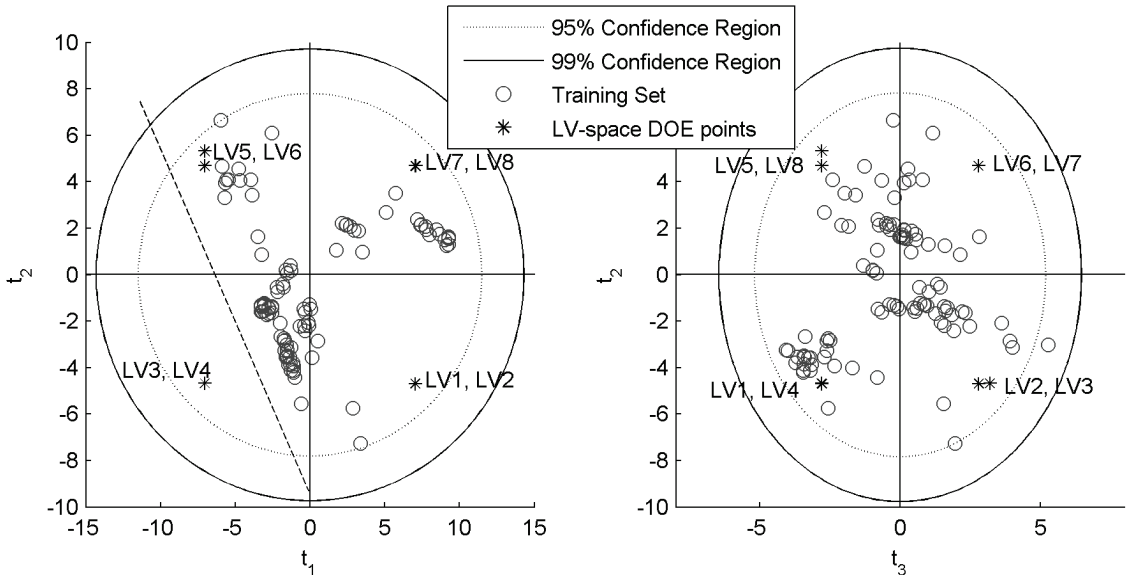

Figure 3.12 Locations of the latent variable space experimental design points in t_1-t_2 (left) and t_2-t_3 (right)

Notice that there seems to be some kind of natural boundary appearing in $\mathbf{t}_1$-$\mathbf{t}_2$, as drawn in Figure 3.12. The two recipes to the left of that boundary, LV3 and LV4, had by far the highest SPE values of all of the DOE recipes, as shown in Figure 3.13. Both experiments were relatively unsuccessful; they were not muffin-like in texture or appearance. This indicates that perhaps the observed boundary in Figure 3.12 delineates a 'feasible muffin boundary', where recipes to the left of that boundary in $\mathbf{t}_1$-$\mathbf{t}_2$ will not exhibit muffin-like qualities.

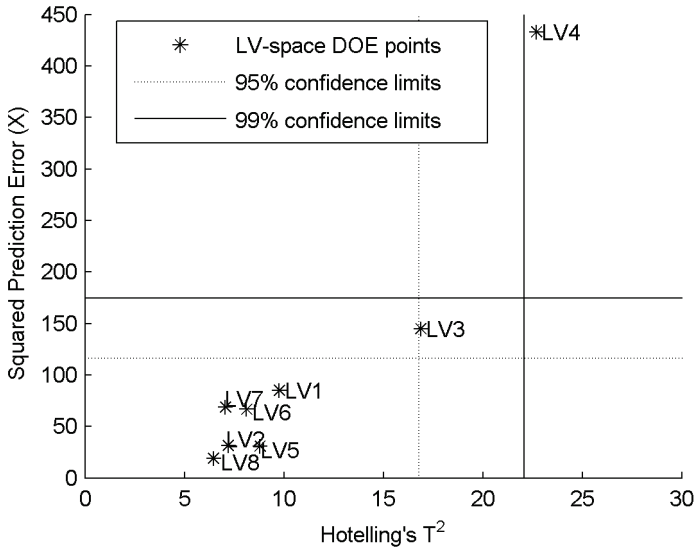

Figure 3.13 SPE vs T^2 for latent variable space DOE points

Next, an additional five experimental points were added, to fill in the 'holes' in $\mathbf{t}_1$-$\mathbf{t}_2$. These additional points are shown in Figure 3.14. Notice that the model has been updated to include the first eight experimental points from the approximate factorial, and that as a result, the locations of the scores shifted slightly.

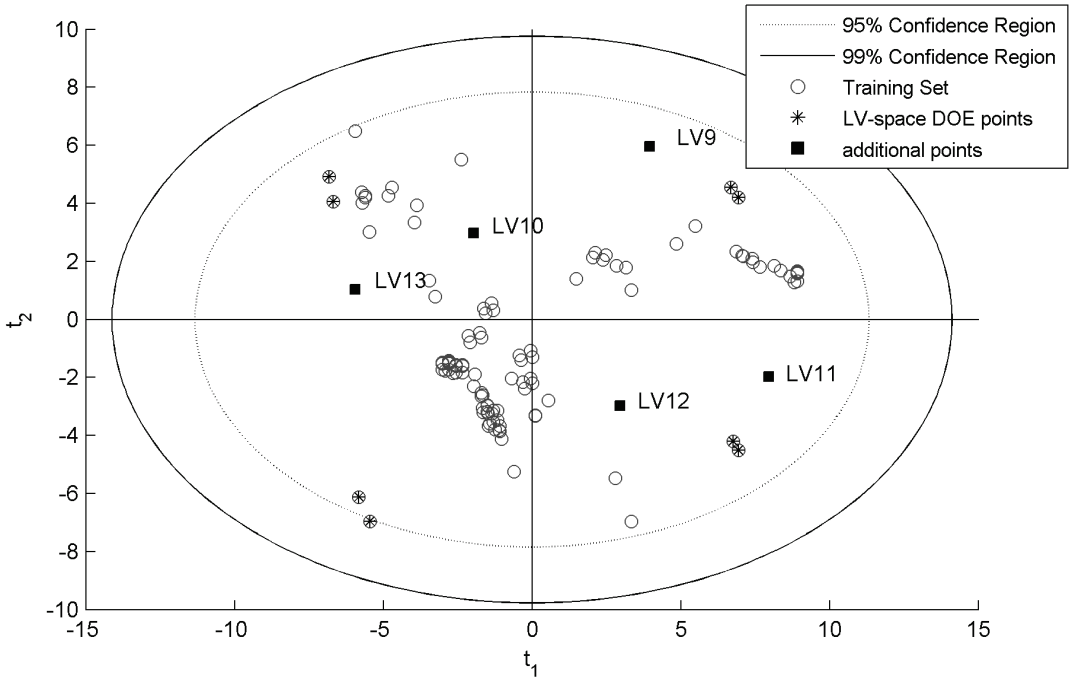

Figure 3.14 Experimental design points added to fill 'holes' in t_1-t_2

All five additional recipes were somewhat feasible in terms of muffin appearance and texture. LV13, located nearest to the 'feasible muffin boundary', was more successful than LV3 and LV4 in that its appearance was more muffin-like. However, it was very dry and dense. When the model is updated with these five points, the observations don't shift very much, and the $\mathbf{t}_1$-$\mathbf{t}_2$ space exhibits a better distribution of observations, very similar to Figure 3.14.

3.4 Modified Mixture-Property Model

Section 2.2 introduced the concept of a mixture-property PLS model, which extends the traditional mixture model into the ingredient property domain. Such a model enables an optimization algorithm to select among many potential ingredients based on their properties, even if some of those ingredients have never yet been used in an experiment. The usefulness of this feature hinges on the availability of a database of potential ingredients and in this case study, such a database was not available.

However, a database was available that contained some properties of the existing ingredients, such as vitamin and mineral content. These properties follow the ideal mixing rule but it was unknown whether or not they would be correlated with the final product property, AOI. In addition, food ingredients exhibit some natural variation between lots and the degree of variability was unknown.

Even though the full benefit of a mixture-property model could not be realized in this situation, the available data was used to build a mixture-property model. Its predictive ability was less than the mixture model built from the same observations, so a modification was conceived. The modification is shown in Figure 3.15; essentially the $\mathbf{X}_{mix}$ matrix is appended to the **R** matrix, rather than replacing it. The purpose of this model was to determine which ingredient properties are correlated with AOI, and whether or not the addition of $\mathbf{X}_{mix}$ would improve the predictive ability of the model.

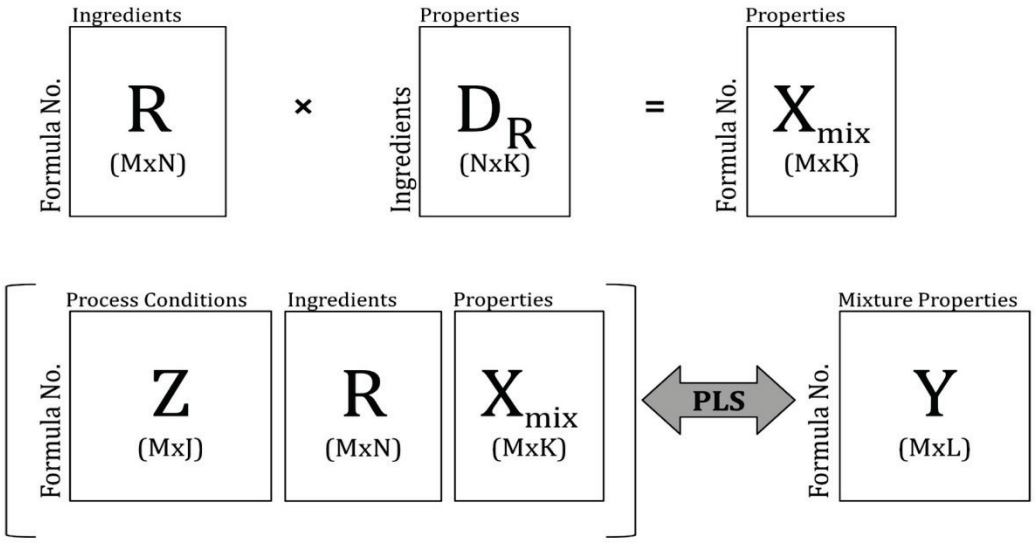

Figure 3.15 Matrices used in modified mixture-property model

As a result of including the ingredient properties in the model, Q^2Y increased from 70% to 80% as shown in Figure 3.16.

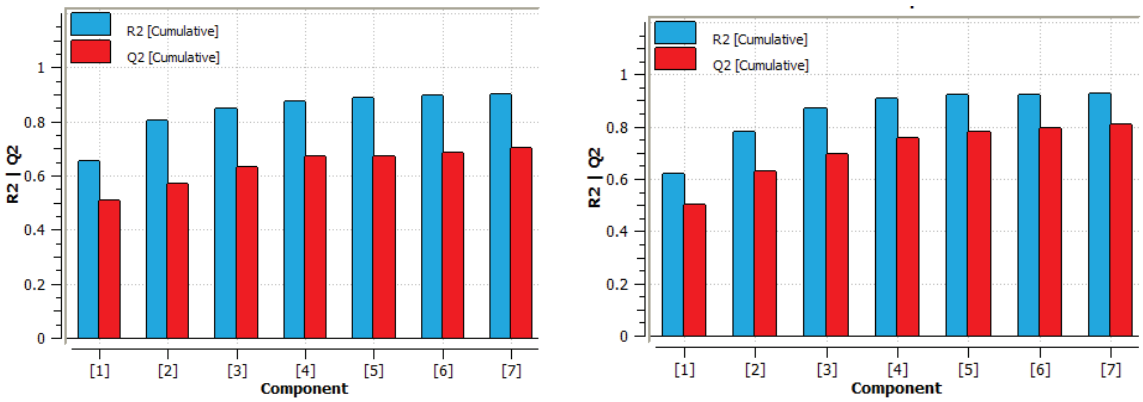

Figure 3.16 Comparison of models before the addition of ingredients properties (left) and after (right)

Several of the ingredient properties (prefix Nutrit) are highly correlated with AOI and these (Nutrit47 and Nutrit49 for example) can be seen near the far left and far right of the coefficient plot shown in Figure 3.17.

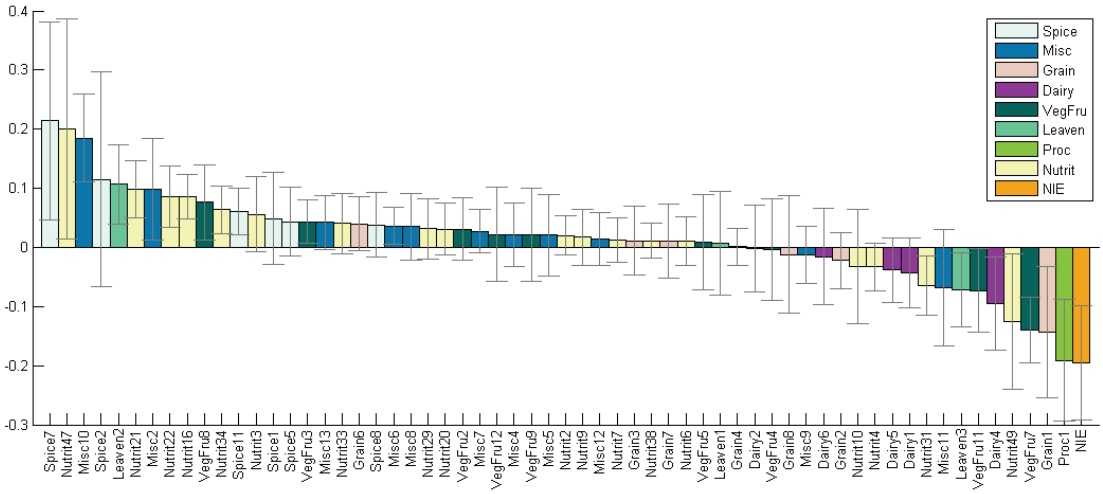

Figure 3.17 Coefficients plot for modified mixture-property model

Based on the increase in Q^2Y and the evidence that several ingredient properties are highly correlated with AOI, the modified mixture-property model structure was adopted for the remainder of the case study.

3.5 Optimization

In section 2.4, general approaches to model inversion and optimization were introduced, and Muteki's optimization formulation (Muteki, MacGregor and Ueda 2006) was presented. This section will discuss the specifics of optimization as it applies to the frozen muffin batter reformulation.

There are some key differences between Muteki's general approach and this case study, namely, the absence of the binary variables, the inclusion of SPE as part of the objective function, and the specification of cost as a constraint. The binary variables are omitted because a database of alternative ingredients is not available and therefore ingredient selection based on their properties is not being explored. This eliminates the third term in the objective function, and simplifies the optimization problem. The optimization formulation used in this case study is shown in equation 3.1.

Objective Function

The case study objective is to minimize AOI while maintaining the original taste, texture and appearance as much as possible. The first term in the objective function seeks to minimize AOI, assuming that the desired AOI is set to an acceptably low value. Note that there is no need for a weighting matrix ($\mathbf{W}_1$ in equation 2.12) because there is only one **Y**-variable (AOI). If measurements or even relative rankings were available for taste, texture and appearance, then those properties could also be included in the first term of the objective function, and they could all be optimized together as in (Muteki, MacGregor and Ueda 2006). Because this data is lacking, the objectives for taste, texture and appearance are addressed indirectly by including additional constraints on the **X**-space.

The second term in the objective function is a penalty term on the SPE, which ensures that the optimization result (a new recipe) will be as close to the model plane as possible, and therefore consistent with past recipes. This term also prevents recipes from wandering beyond the 'feasibile muffin boundary' as shown in Figure 3.12.

Constraints

The ideal mixing rule applies to the ingredient properties used in this case study, and is used to generate $\mathbf{x}_{\text{mixnew}}^{\text{T}}$, which is appended to $\mathbf{r}_{\text{new}}^{\text{T}}$ to obtain $\mathbf{x}_{\text{new}}^{\text{T}}$. Any new recipe must have a lower value of T^2 than the 95% confidence limit, and the sum of its ingredient

proportions must total 100%; these are taken into account via the PLS model constraint and the mixture constraint respectively. Unlike Muteki's formulation, there are specific minimum and maximum values for each ingredient. The model covers many different formulas, each with their own flavour profile, so these ingredient constraints are unique to each formula. They are needed to ensure that critical flavours are present in, and unwanted flavours are omitted from, new recipes for each formula. New ingredient E (NIE) is constrained to zero; the optimization only considers ingredients from the original 26 muffin recipes. Constraints are also imposed on certain nutritional measures such as fat, fibre, sugar, and salt. Cost, while definitely a concern, is not the primary objective of the project, so cost containment is implemented as a constraint rather than as a term in the objective function.

$$\min_{\mathbf{r}_{new}} \left(\mathbf{y}_{des} - \boldsymbol{\beta}_{PLS}^{T}\mathbf{x}_{new}\right)^{T} \cdot \left(\mathbf{y}_{des} - \boldsymbol{\beta}_{PLS}^{T}\mathbf{x}_{new}\right) + \mathbf{w}_2 \cdot \mathrm{SPE}_{new} \qquad \textbf{3.1}$$

subject to:

Ideal Mixing Rule	$\mathbf{x}_{new}^{T} = [\mathbf{r}_{new}^{T} \vert \mathbf{r}_{new}^{T} \cdot \mathbf{D}_{R}] = [\mathbf{r}_{new}^{T} \vert \mathbf{x}_{mixnew}^{T}]$
SPE Calculation	$\mathrm{SPE}_{new} = \sum_{k=1}^{K} (\mathbf{x}_{new} - \hat{\mathbf{x}}_{new})^2$
PLS Model Constraint	$T_{new}^2 = \sum_{a=1}^{A} \frac{\boldsymbol{\tau}_{new,a}^2}{s_a^2} \leq T_{max}^2, \alpha = 0.05$
Mixture Constraint	$\sum_{j=1}^{NN} \mathbf{r}_{new,j} = 100\%, \qquad 0 \leq \mathbf{r}_{new,j} \leq 1$
Ingredient Ranges	$LB_j \leq \mathbf{r}_{new,j} \geq UB_j$
Nutrition Constraints	$LB_j \leq \mathbf{x}_{mixnew,j}^{T} \geq UB_j$
Cost Constraint	$\sum_{j=1}^{NN} \mathbf{r}_{new,j} \cdot c_j \leq cost_{max}$

N.B. *NN* is the total number of ingredients listed in the ingredient property database, $\mathbf{D}_R$

Feasibility

Section 2.3 included a short section on assessing the feasibility of the desired final product properties by creating a PCA model on **Y**. This is unnecessary here because there is only one **Y**-variable, and the objective is to minimize it for specific formulas in the product line. Because there are already lower values of AOI in the dataset, it is known that achieving lower values is possible; what is unknown is whether it is achievable while maintaining the original taste, texture, and appearance of those formulas.

Model-Informed Recipe Improvements

A model-informed recipe is one that results not from a formal optimization but rather from a practitioner's use of process knowledge in combination with the new knowledge gleaned from a PLS model. In general, this is not a recommended approach but due to the lack of **Y**-variables, model-informed recipe improvements were undertaken alongside optimization. The lack of **Y**-variables created some uncertainty as to whether the PLS model and the optimization constraints would provide the optimization algorithm with enough guidance to come up with successful recipes.

3.5.1 Updated PLS Model

The PLS model was updated just prior to optimization to include the latest available observations. It contained 356 observations, 61 variables, and had seven components.

R^2X

It is important to look at R^2X when determining whether a model is suitable for inversion. R^2X indicates the percentage of the variation in **X** that is explained by the model, and if this value is low, then the optimized recipes are not as likely to produce the desired results. Figure 3.18 shows a component summary of the model used for optimization. Only seven components were deemed significant by the auto-fit rules, but four more were added manually to improve the R^2X from 60% to 74%. This also helps to compensate for the fact that there is only one **Y**-variable to optimize and yet several unmeasured **Y**-variables such as texture, taste and appearance that need to be consistent with the original recipes. Because SPE is penalized in the objective function, adding dimensions to the model further constrains the solution. In the absence of more constraints (i.e. product properties) in **Y**, the extra constraints on the **X**-variables help enforce consistency with past data. The addition of

the four extra components reduces Q^2Y slightly, by 0.4%. In general, the number of components for a model that will be inverted should be chosen to make a reasonable tradeoff between R^2X and Q^2Y.

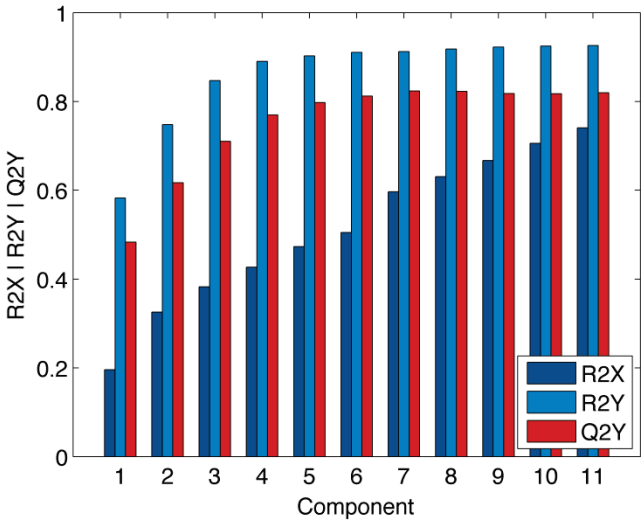

Figure 3.18 Summary plot for optimization model

Figure 3.19 shows the effect of the four additional components on the PLS coefficients.

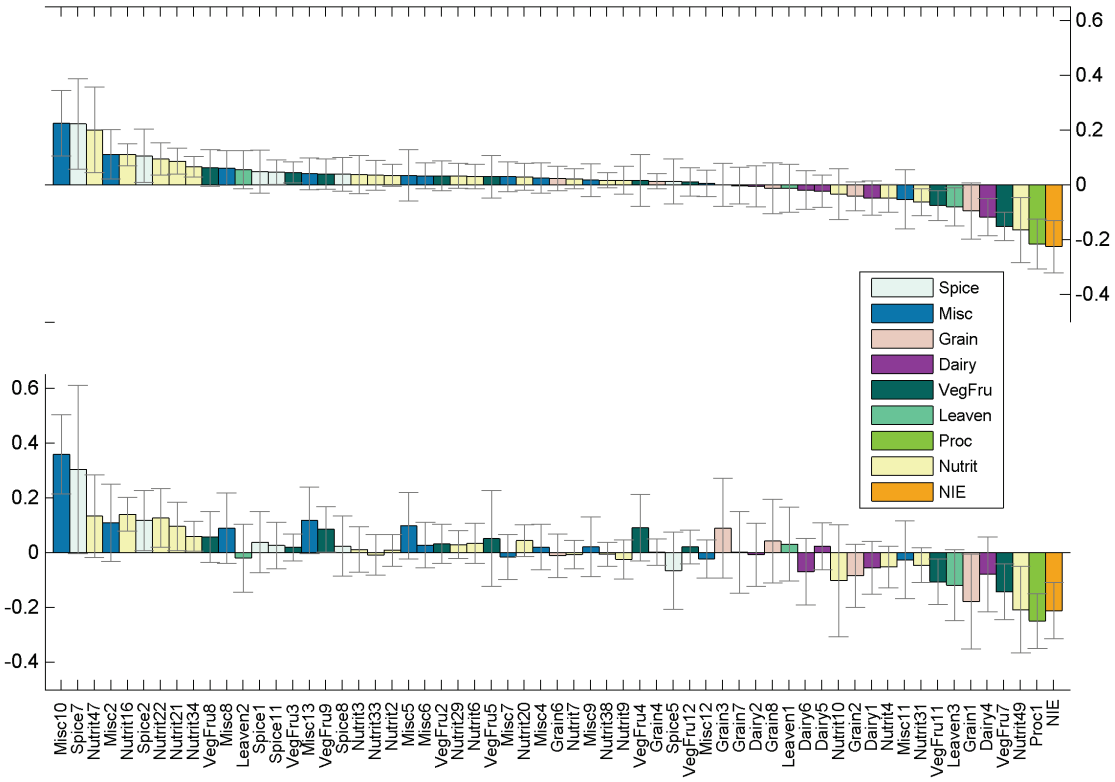

Figure 3.19 Coefficients for the updated PLS model, with 7 components (top) and 11 components (bottom)

3.6 Results

In this case study, four formulas were targeted for reduction of AOI; Formula 11, Formula 7, Formula 6, and Formula 18. In each case, optimization, model-informed ingredient substitutions, or both were used to generate new recipes. The new recipes were made up in the laboratory and were tested for AOI. They were also evaluated qualitatively regarding taste, texture and appearance, relative to the original recipes.

The plots in this section refer to experiment numbers ("Expt"), where an experiment is defined as a group of trial recipes made up at the same time using the same lots of ingredients. Some experiments include recipe variations on more than one of the original formulas. A control batch is made up for each formula in each experiment, which is important because some of the variation in AOI is attributed to variation in the ingredients, and the control batch provides a direct comparison. It also provides fresh muffins for comparison of appearance, taste, and texture.

Formula 11 was the first formula to undergo optimization. Recipes based on the 7-component model and the 11-component model were presented to the industrial partner for evaluation. The recipes based on the 11-component model were judged by the product developers to be more likely to produce desirable muffins, so from that point forward the optimization algorithm was configured to use the 11-component model exclusively. The four extra components increase the value of R^2X, but they slightly decrease Q^2Y. Since only the first seven components are significant for predicting new values of **Y**, the results in this section are presented against the 7-component model.

For each optimized formula, the product developers were presented with several alternative recipes and asked to choose one or more to execute. To generate these alternatives, the cost constraint and the weighting on SPE in the objective function were varied. In some cases, constraints on individual ingredient ranges were modified at the collaborator's request and the optimization was re-run to generate updated alternatives.

Formula 11

The baseline data for this formula was all collected during a single production run. This data underestimates the variance of the formula, as can be seen Figure 3.20 (left) by comparing it to the AOI values from 'Other Control Batches' which were produced in the laboratory as

part of several other experiments. Three optimization results were executed in Expt19 and all three muffins were easily distinguishable from the control batch (Obs 388) in terms of appearance. Obs391 came closest to the appearance, taste, and texture of the original formula but had the highest AOI value. Even so, Obs391 represents a 49% reduction in AOI from the control batch. All three modified recipes had higher AOI values than the PLS model predicted (see Figure 3.20, right), but the predictions were nonetheless directionally accurate and useful for the product developers. A further reduction in AOI is desired; this is possible if some of the ingredient constraints are further relaxed but it may cause the appearance, taste, or texture to deviate more from the original product.

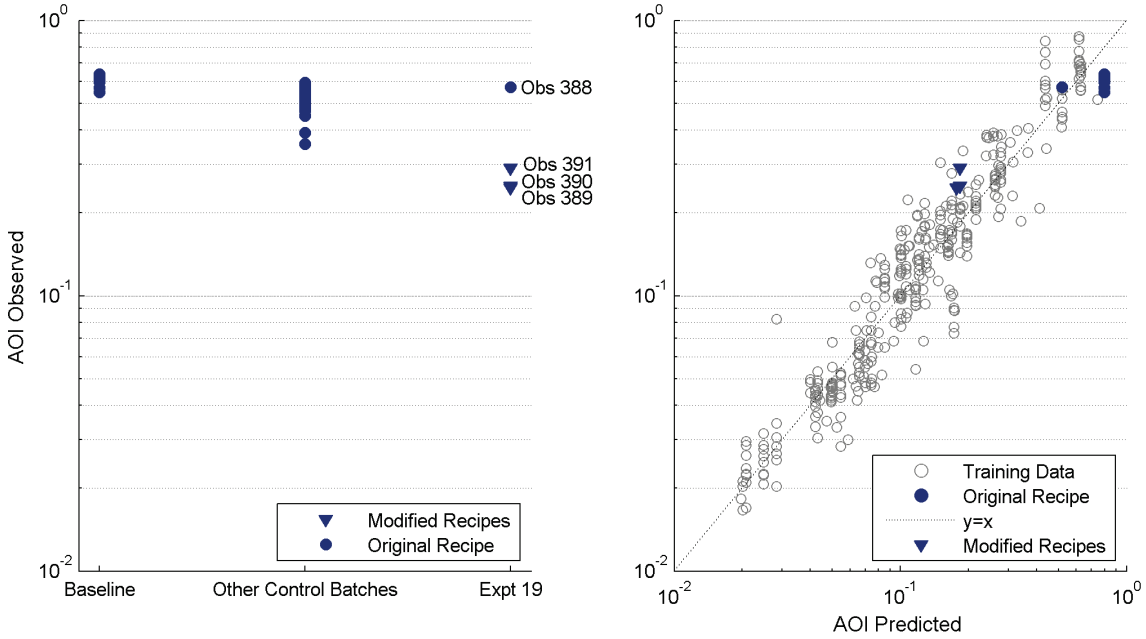

Figure 3.20 Optimization results for Formula 11

Formula 7

The first set of baseline data for Formula 7 was suspiciously high, therefore a second set of baseline data was collected during a separate production run and it was much lower. The difference was attributed to the lot-to-lot variation of ingredient VegFru2. Figure 3.21 shows that subsequent control batches were comparable to the second set of baseline data.

Expt12 was part of a designed experiment completed near the beginning of the project, long before the optimization step. The PLS model showed that Leaven2 was positively correlated with AOI while Leaven3 was negatively correlated with AOI (see Figure 3.5) and that the same correlations existed with the standard deviation of AOI (see Figure 3.6). Therefore, it

seemed sensible to substitute Leaven3 for Leaven2, and this was tested in Expt12. Obs322-Obs325 (see Figure 3.21) are variations on Formula 7 that include Leaven3 at several different levels. These trials were very successful; not only did the muffins look and taste reasonably similar to the control batch; the AOI was also reduced significantly.

When the time came to consider optimizing Formula 7, it made sense to revisit the model-informed recipes of Expt12. Obs324 and Obs325 were chosen as the preferred recipes and they were repeated with a slight modification (the elimination of Spice7) in Expt20. Of the two formulas, the modification of Obs324 (i.e. Obs398) was the preferred choice and it was repeated in Expt21 (Obs403). As seen in Figure 3.21, Obs398 and Obs403 represent an average reduction of 55% in AOI from their control batches (Obs396 and Obs402 respectively). However, the taste of the control batches was preferred by the product developers versus the modified recipe.

Further experimentation is needed to determine the AOI value for the new recipe when another lot of VegFru2 is received, i.e. one that causes AOI to be high in the control batch.

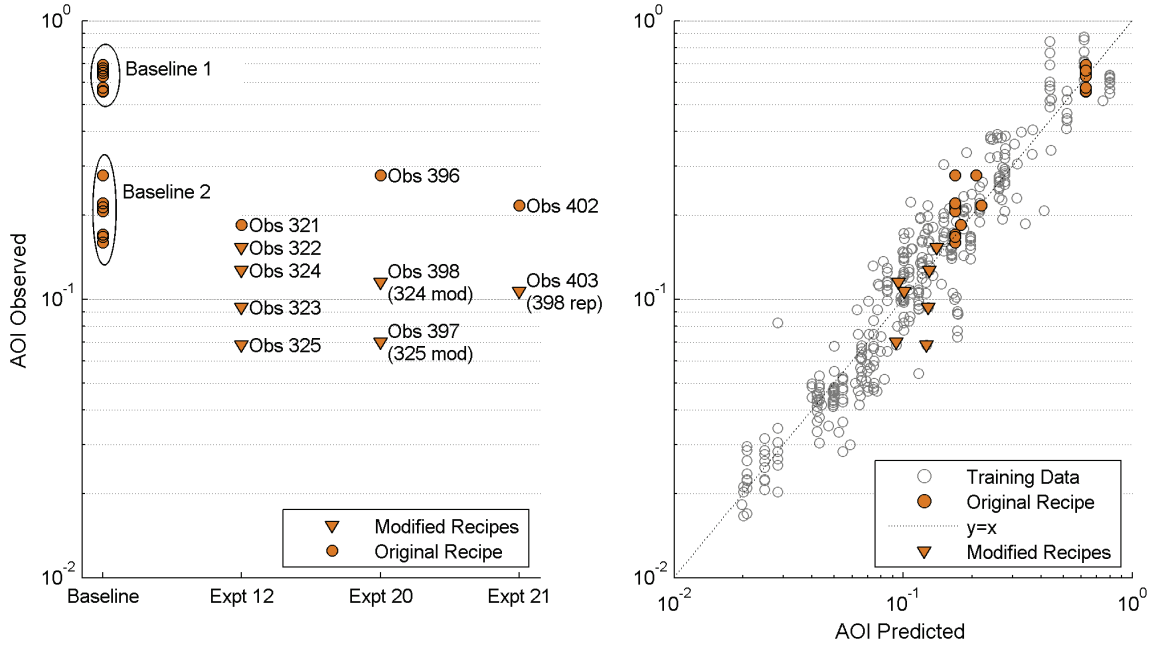

Figure 3.21 Model-informed recipe improvements for Formula 7

Formula 6

Compared to the other three formulas in this case study, Formula 6 already had fairly low values of AOI, and because of its similarity to Formula 7, the same ingredient substitution was executed. The results are shown in Figure 3.22. Obs400 and Obs401 are model-

informed recipes containing Leaven3 rather than Leaven2. Both recipes had much lower AOI values than the control batch (Obs399) made at the same time. Obs400 was preferred over Obs401 for its taste and appearance, although it was not an exact match for the original appearance and the product developers preferred the taste of the control muffin. Business considerations dictated that the process conditions for Formula 6 would have to be changed in the future, so in Expt21 the original recipe and the Obs400 recipe were produced using both the new and old process conditions. Both the original and the modified recipe showed increases in AOI values under the new process conditions. Obs400 and Obs 410 represent an average AOI reduction of 54% from their respective control batches (Obs399 and Obs408) under the old process conditions. Obs409 shows a 43% reduction compared to its control batch (Obs407) when the new process conditions are used.

AOI values for Obs400, Obs401, and Obs410 are all less than those predicted by the PLS model (see Figure 3.22, right). Recipes for Formula 7 that used the same substitution of leavening ingredients were also lower on average than the PLS model predicted (see Figure 3.21, right). This demonstrates that the model underestimates the effect of this substitution for these two similar formulas.

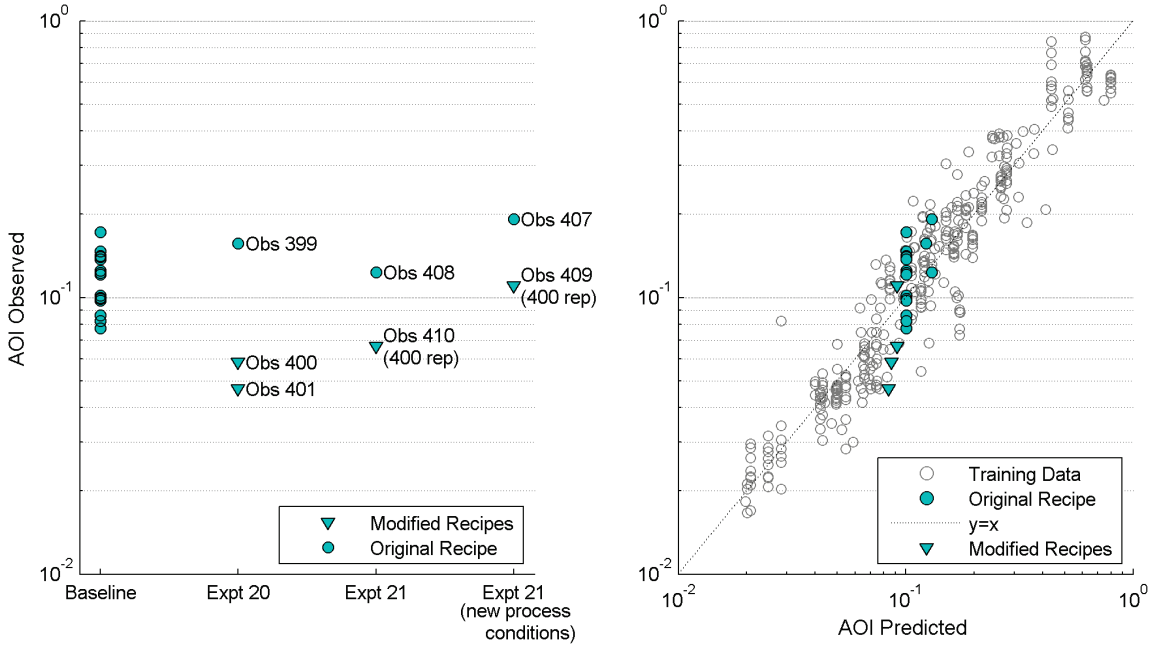

Figure 3.22 Model-informed recipe improvements for Formula 6

Formula 18

The baseline data for Formula 18 was all collected during one production run. Formula 18 shares a common ingredient (VegFru2) with Formula 7 and Formula 6 and that ingredient was found to be highly variable between ingredient lots. This explains the large difference between the range of the baseline data and the control batches completed in Expt20 and Expt21. Figure 3.23 illustrates this difference and also the AOI values for the modified recipes.

Obs393 is a model-informed recipe based on the successful inclusion of Leaven3 as a substitute for Leaven2 in Formula 7. Its SPE value is between the 95% confidence limit and the 99% confidence limit, which might have been a deterrent to executing the recipe except for the previously successful substitution in Formula 7. Obs394 and Obs395 are results of the optimization algorithm; all three modified recipes produced muffins that were comparable in appearance to the control muffins. Of particular note is that the model-informed recipe was not the preferred choice. Obs394, an optimization result, came closest to the original appearance, and in fact its taste was preferred to the original by the product developers. It was therefore repeated in Expt21 along with a slight modification (the elimination of Spice7). Obs394 and Obs406 represent an average AOI reduction of 47%, from their respective control batches (Obs392 and Obs404). AOI values for Formula 18 were higher than predicted by the PLS model (see Figure 3.23, right).

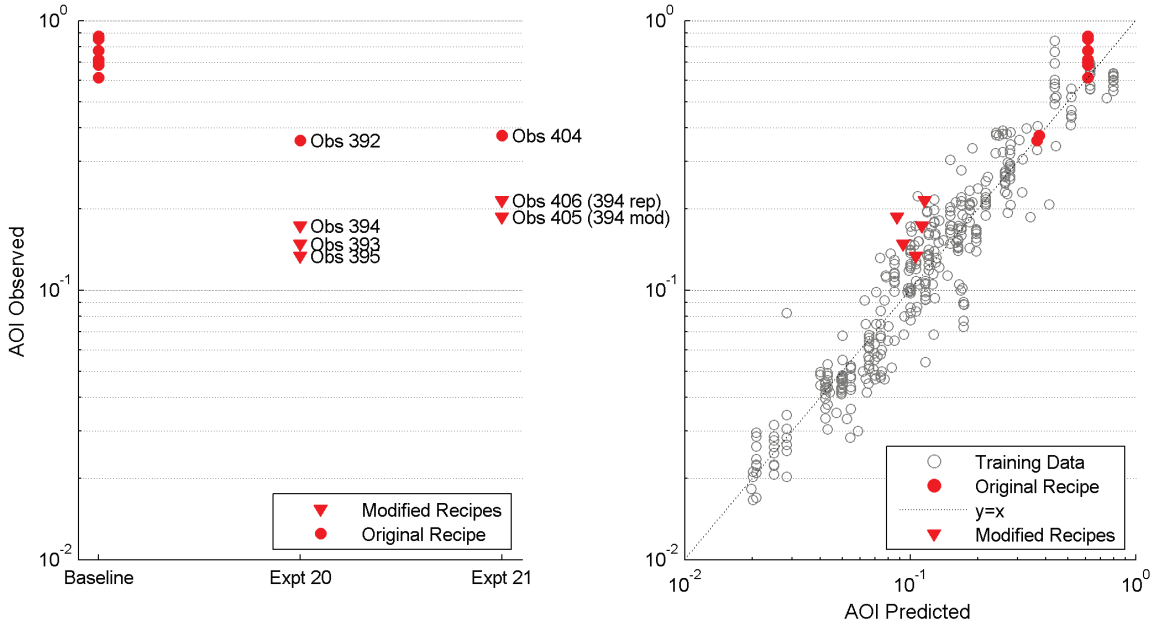

Figure 3.23 Optimization results and model-informed recipe improvements for Formula 18

Results Summary

In summary, the reductions in AOI were significant for all four modified formulas, and they are displayed in Table 3.2. The maintenance of original taste, texture, and appearance were more challenging constraints to meet; measured values for these traits were not available in the baseline data, so they could be neither modeled nor included in the optimization. Table 3.2 also shows the root mean squared error for all of the modified recipes discussed in this section. The PLS model is better at predictions for Formula 6 and Formula 7 than it is for Formula 11 and Formula 18, but this is not surprising as all of the recipes for Formula 11 and Formula 18 have significantly larger SPE and T^2 values.

	AOI Reduction (for the preferred recipe modification)	Root Mean Squared Error (for all modified recipes)	Average Squared Prediction Error (SPE-X) (for all modified recipes)	Average T^2
Formula 6	54%	0.031	17.9	2.7
Formula 7	55%	0.029	16.1	2.8
Formula 11	49%	0.074	28.4	5.3
Formula 18	47%	0.082	46.3	12.3

Table 3.2 Summary of results and PLS statistics for the four modified formulas

With regards to the PLS model, the interpretation of a loading biplot becomes very clear when the locations of the original and modified formulas are compared. Figure 3.24 shows that all of the modified formulas are located further up and to the left in $\mathbf{t}_1$-$\mathbf{t}_2$ compared to their respective original formulas. This makes sense because AOI is located in the bottom right corner, therefore formulas with high values of AOI will be located near it, and formulas with the lowest values of AOI will be located diagonally across the origin.

Selected ingredients that have been mentioned in this section are also labeled on the biplot. Leaven2 is located near AOI, indicating they are positively correlated, while Leaven3 is located in the far upper left quadrant, indicating that it is negatively correlated with AOI. Therefore it makes sense that the substitution of Leaven3 for Leaven2 would decrease AOI values. VegFru2 is an ingredient that varies widely from lot to lot, and Nutrit47 is the variable that captures this variation. Nutrit47 is highly correlated with AOI. Spice7 and Misc10 are likewise highly correlated with AOI.

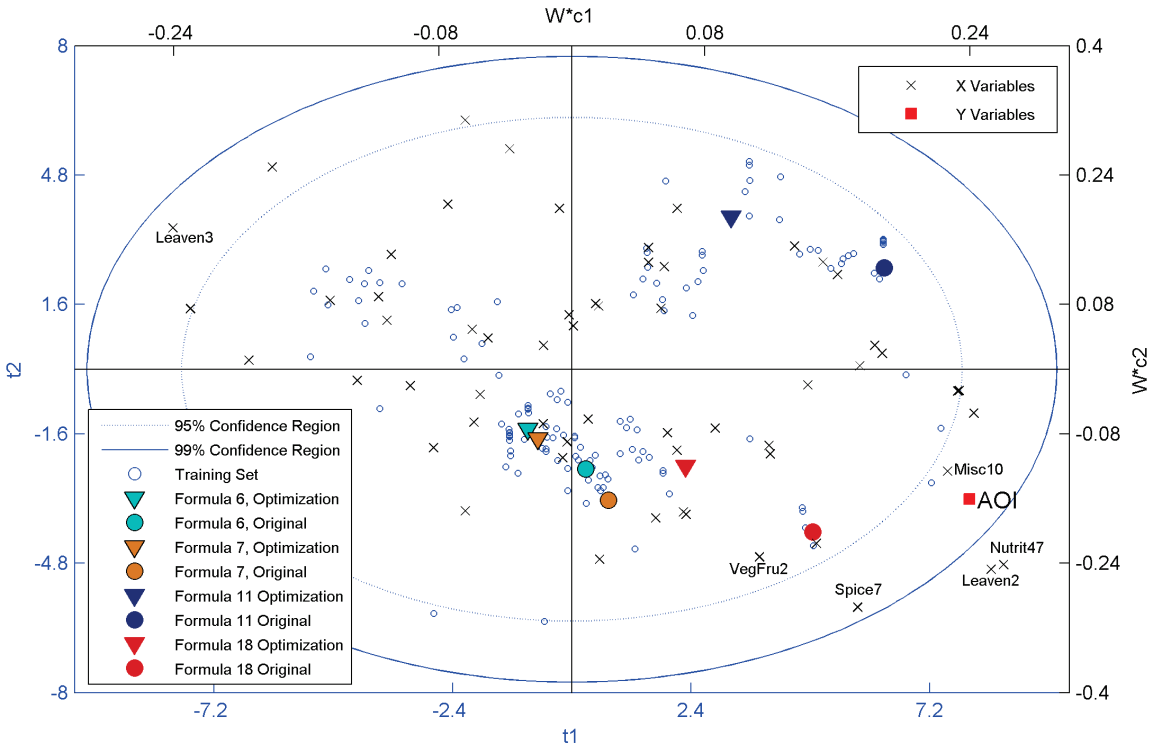

Figure 3.24 t_1-t_2 score plot comparing the locations of original and modified formulas. For each of the four products, the most successful modification is shown with its respective original formula.

3.7 Discussion and Recommendations

In this case study, latent variable model-based optimal reformulation techniques were successfully applied to four formulas for frozen muffin batters. The main goal was to reduce AOI values in prepared muffins, and that goal was achieved, with an average reduction of just over 50% for the modified formulas. A further reduction is desired for two of the four formulas, Formula 11 and Formula 18. Future testing on Formula 7 with different lots of VegFru2 will determine whether a further reduction is required for that formula. A sufficient reduction has been achieved for Formula 6, although its recipe could benefit from further experimentation to bring its taste more in line with the original product.

Maintaining the taste, texture, and appearance of the original muffin formulas was also a goal of the project. Although these traits were not present in the PLS model, some success was achieved in this area due to the use of ingredient constraints and a penalty on SPE in the objective function. An optimized recipe for Formula 18 produced muffins that were comparable in appearance, and preferred by the product developers (taste-wise) versus the original recipe. For Formula 7, a model-informed recipe produced muffins that were similar to the original recipe, although the product developers still preferred the taste of the control batch. Modifications to Formula 6 and Formula 11 yielded muffins that were distinct from the control batches, primarily in taste and appearance respectively.

Rapid reformulation has been a very useful tool in the AOI-reduction effort. Had this technique not been available, the product developers probably would have begun by performing some ingredient DOE's in the **X**-space for the high-AOI formulas. This likely would have produced some misleading results as demonstrated in section 3.3.1. Prohibitive numbers of experiments would have been needed to determine all of the ingredient effects in this manner. Another way to identify the roles of each ingredient would have been to analyze each one for a potentially AOI-related trait, but given the large number of ingredients this would have been expensive and time-consuming. In addition, individual raw ingredient testing cannot take into account the behaviour of ingredients during baking, including interaction effects.

Latent variable modeling is a more effective way to determine which ingredients have the strongest effects on AOI. It can identify ingredients as being positively or negatively correlated with AOI, faster and with less cost than exhaustive testing of each ingredient. The

results are more widely applicable than those from an **X**-space DOE, and many more factors can be tested in far fewer experiments. Rapid reformulation provided the product developers with a set of techniques to address AOI reduction in a situation that might have otherwise seemed daunting and without a clear path forward.

Several times throughout Chapter 3, data deficiencies were noted. It is worthwhile to revisit these deficiencies in order to highlight the additional potential of rapid reformulation techniques. Firstly, each set of eight replicate samples in the baseline data was collected in a single production run (for most of the formulas) and therefore they were not completely independent observations. Understandably, it may not always be practical (e.g. time-wise) to collect many independent samples. A better use of resources would be to collect fewer totally independent samples and execute a few carefully chosen experimental points in the latent variable space to augment the smaller set of baseline data.

In addition, the **X**-data was identical for all eight observations, because only the set points for each ingredient (not the measured amounts) were recorded. If ingredient measurements are very accurate then the variations in **X**, had they been recorded, would have been small and the model would not have been greatly affected. Deviations from set points, if significant, should be included in the model.

Section 3.4 discussed the potential value in having a database of prospective ingredients and their properties, namely that it enables the optimization to search over all possible ingredients to find a new recipe. In this case study such a database was lacking, so the optimization was only able to consider ingredients that had been used in a past or present formula. This was appropriate considering the case study's objectives, but the capability to incorporate ingredients that have never yet been tried in a recipe has proved very powerful (Muteki, MacGregor and Ueda 2006).

Finally, the importance of collecting data on all of the relevant properties of a final product cannot be understated. Using a PLS model with as many **Y**-variables as there are relevant final properties, an optimization algorithm can provide solutions that meet many competing constraints and criteria. It is against the desired values of each of these properties that the algorithm evaluates a potential new recipe. For the muffin batters, only one property was measured, and the goal of the project was to minimize it without affecting any of the unmeasured properties. In the absence of the unmeasured properties, the best that the

algorithm could do was to adhere to constraints in the **X**-space: the PLS model constraints and some supplementary constraints on ingredient ranges that were based on prior knowledge of the ingredients and products. These additional constraints were constructed by trial-and-error, adding time and iterations to the project. As experimentation progressed, more **Y**-variables were measured, generating new data which can be used in further iterations. As the collection of this data is adopted as a standard practice, product developers gain the ability to respond quickly to a wide range of future reformulation requirements.

Appendix to Chapter 3: Cross-Validation

Cross-validation, first described in (Wold 1978), is a technique for determining how many significant principal components are contained in a data set. In this procedure, observations are arbitrarily divided into G groups and G reduced models are built, where all of the data except the g^{th} group is used in the g^{th} model. For each reduced model, the omitted data is used as a prediction set, with each observation being left out once and only once. The sum of squared differences between the predicted and actual values in the prediction sets are calculated for each of the G reduced models, and their total is called the predictive residual sum of squares, or PRESS. The predictive ability of a model is often expressed as

$$Q^2Y = 1 - \frac{PRESS}{SS} \qquad \textbf{3.2}$$

where SS is the sum of squares of **Y**.

The cross-validation procedure is completed and a value of Q^2Y is calculated for each latent variable dimension, or component. Using ProSensus MultiVariate, a component is determined to be significant if it meets one of the following default[5] criteria: (1) Q^2Y is greater than 1% (for all of the **Y**-variables) or (2) Q^2Y for any one **Y**-variable is greater than 5%.

In implementing cross-validation, there are some choices to be made, such as the choice of the number of cross-validation groups and how the observations are assigned to those groups. (Umetrics AB 2006) suggests that between 5 and 10 cross-validation groups usually work well. By default, ProSensus Multivariate uses 7 cross-validation groups[6], and observations are arbitrarily assigned to cross-validation groups. The structure of data should also be considered when using cross-validation, as it can break down in the presence of grouped data or data from designed experiments.

Data from Designed Experiments

Consider the case where the data to be modeled is from a designed experiment. The omission of an extreme design point can change the correlation structure of the data, causing the omitted point to be an outlier in terms of the reduced model. This can cause Q^2Y to be unrealistically low.

[5] The cross-validation criteria can be adjusted in ProMV at the user's discretion.
[6] The number of cross-validation groups can be modified in ProMV at the user's discretion

Grouped Data

The presence of grouped data can also be problematic. Grouped data form clusters in the latent variable space, because of similarities between observations; either true replicates or near-replicates such as slight recipe variations. In the case of grouped data, it is unlikely, using arbitrarily-assigned cross-validation groups, that an entire cluster will be omitted from a reduced model, and the presence of similar observations (the rest of the cluster) in the reduced model will lead to an overly accurate prediction for its omitted members. This will lead to an overstatement of the model's predictive ability. (Clark and Fox 2004) demonstrated the effect of redundant data in cross-validation; adding duplicate observations caused a significant increase in Q^2Y.

The baseline data set as described in section 3.2 can suffer from both problems, i.e. the effect of designed data and the effect of grouped data. It forms clusters because there are eight replicate observations of each muffin formula, as can be seen in Figure 3.25. The $\mathbf{t}_1$-$\mathbf{t}_2$ plot (left) appears to have only one point for each muffin formula, and that is because the **X**-data is identical for all eight points. But, there is variation in the **Y**-values within a cluster, and that can be seen in the $\mathbf{u}_1$-$\mathbf{u}_2$ plot (right).

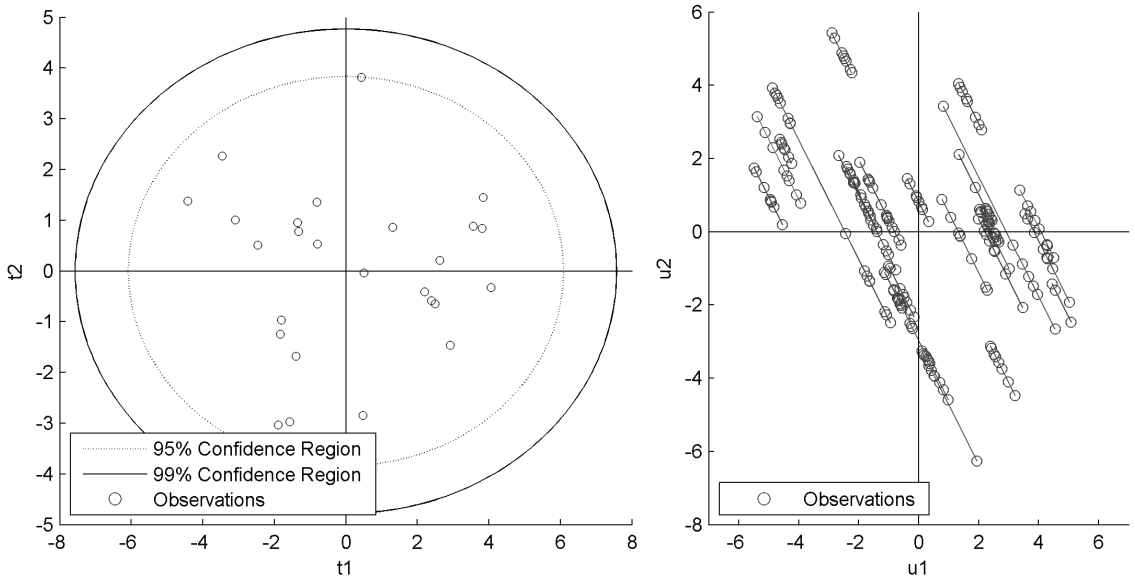

Figure 3.25 Score plots for the baseline mixture model

Lack of Independence

The baseline data also suffers from a lack of independence. For many of the formulas, all eight observations were collected during the same production run, although they were each from a different batch. Therefore, those observations have ingredient lots, environmental factors, and other sources of variation in common. The following discussion addresses cross-validation when grouped data is present in a data set, but these considerations are even more important when the group members are not completely independent. In a production environment it is often impractical to collect fully independent observations.

One solution to the grouped data problem is to implement cross-validation such that each cluster forms one of the G groups which are omitted from the reduced models. (ProSensus MultiVariate has the option to use a secondary identification variable to form the cross-validation groups.) On the other hand, using cluster-based cross-validation makes the data more similar to data from a designed experiment, and there is potential for an understated Q^2Y. Leaving out an entire cluster at a time is somewhat like having data from a factorial design and leaving out a corner point to build the model, and then using that model to predict the y-values for the corner point. The corner point is now an outlier and the chances of a good prediction are low. Figure 3.26 and Table 3.3 contrast the use of seven arbitrary cross-validation groups with cross-validation by formula group, for the baseline data. Recall that there are 26 groups in this data set. When arbitrary cross-validation groups are used, there are four significant components with a total Q^2Y of 93%, whereas using formula groups gives a total Q^2Y of 67% and only three significant components. These two values provide some idea of the possible range for the 'true' value of Q^2Y.

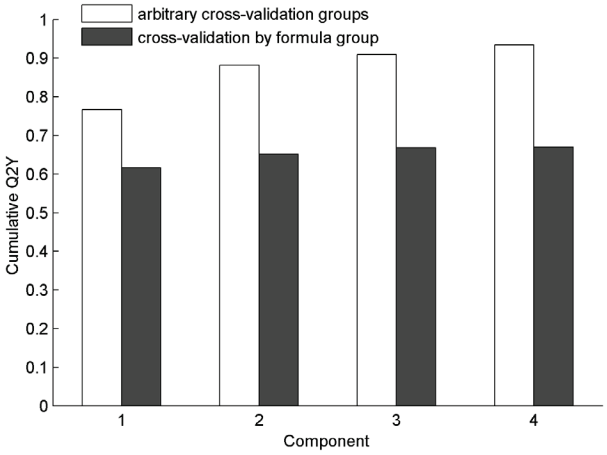

Figure 3.26 Comparison of cross-validation grouping strategies

	Q^2 (Arbitrary Cross-Validation Groups)	Q^2 (Cross-Validation by Formula Group)
PC 1	0.77	0.62
PC 2	0.88	0.65
PC 3	0.91	0.67
PC 4	0.93	

Table 3.3 Comparison of cross-validation grouping strategies

This issue of clustered data is likely to crop up in other product development projects, especially where an entire product line is being modeled. Beyond the choice of cross-validation groupings, there are further considerations. As experiments are added to a baseline data set, the groupings for cross-validation become less clear. The more a recipe is modified, the further it gets from the corresponding original formula. A practitioner must decide at what point a formula is considered to be different enough from the original that it should not be classed with its parent formula, and becomes its own group. In this case study, any formula variation was always classed with its parent formula.

Additionally, variations in some of the raw materials (such as VegFru2) are not trivial, therefore it is questionable whether recipe variants of Formula 7 in experiment 12, for example, should be grouped with other recipe variants of Formula 7 in experiment 20, which was executed with a different batch of VegFru2. In addition, a control batch is made up during each experiment, so a further quandary is whether this control batch should be grouped with all other identical (in **X**) observations or with its recipe variants in the same experiment. For the purposes of this project, cross-validation groups were defined as follows: (1) each group contains all observations of a specific muffin formula in a specific experiment, including the corresponding control batch (i.e. the cross-validation group is defined by the concatenated formula number and experiment number [Formula# Expt#]) and (2) each experiment that was added to fill 'holes' in the latent variable space was treated as its own group. This way, the value of Q^2Y falls somewhere in between the two extreme cases presented earlier.

Figure 3.27 and Table 3.4 contrast all three cross-validation alternatives discussed, for a model that includes all of the data collected in this project. The arbitrary cross-validation suggests a model of only four components (Q^2Y=86%), the most conservative method, using formula groups, suggests a model of seven components (Q^2Y=77%), and the intermediate case using [Formula# Expt#] suggests a model of five components (Q^2Y=79%). This again confirms (Clark and Fox 2004)'s findings that redundant data artificially drives up Q^2Y. The

conservative case was used in building the baseline mixture model (section 3.2), and the intermediate case was used in building all subsequent models.

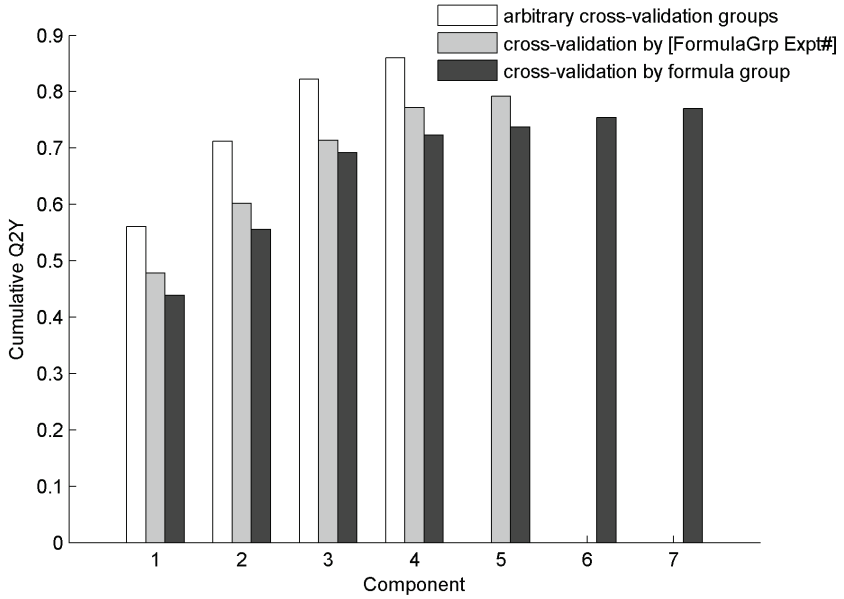

Figure 3.27 Comparison of three cross-validation alternatives

	Q^2 (Arbitrary Cross-Validation Groups)	Q^2 (Cross-Validation by [Formula# Expt#])	Q^2 (Cross-Validation by Formula Group)
PC 1	0.56	0.48	0.44
PC 2	0.71	0.60	0.56
PC 3	0.82	0.71	0.69
PC 4	0.86	0.77	0.72
PC 5		0.79	0.74
PC 6			0.75
PC 7			0.77

Table 3.4 Comparison of three cross-validation alternatives

The confidence intervals calculated for each model coefficient are greatly affected by the type of validation used, because they are also calculated during cross-validation, by a process known as jackknifing (Martens and Martens 2000). For each of the G reduced models built during cross-validation, the PLS coefficients are estimated, resulting in G estimates per coefficient. The standard error is calculated from the G estimates. If an entire cluster of data points is left out for each of the G models, then each of those models could be quite different and have very different values for each coefficient, which leads to larger confidence intervals. Figure 3.28 illustrates how much larger the confidence intervals are when arbitrary cross-validation groups are used (top) versus cross-validation based on formula group (bottom), for a model that includes all of the data collected in this project.

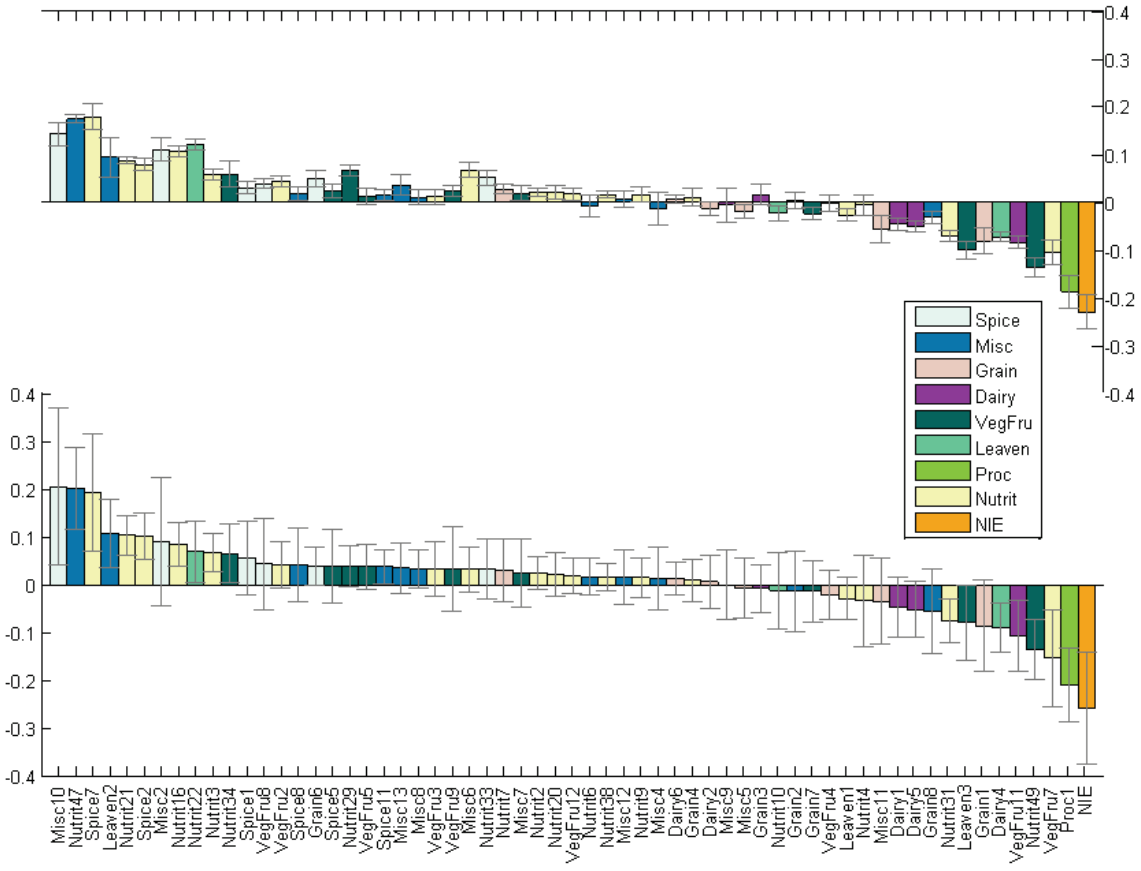

Figure 3.28 PLS coefficients and confidence intervals as calculated using arbitrary cross-validation groups (top) and flavour-based cross-validation groups (bottom). These are the two extreme cases presented in Figure 3.27.

To summarize, it is important to consider how the nature of data can impact cross-validation. Clusters may occur when a product development model includes several variations of a similar product; ie. multiple muffin flavours or polymer grades. Designed data may be present when an effort has been made to fill in any holes in the latent variable space. Practical considerations may dictate the collection of data in such a way that observations are not completely independent. Furthermore, as a recipe is modified, a decision must be made as to how similar it must be to its original formula to qualify as part of the same cross-validation group. Computing a Q^2Y value for arbitrary cross-validation and for a conservative grouping method will establish upper and lower bounds on the 'true' value of Q^2Y.

Chapter 4 Hyperspectral Image Analysis Applied to Oat Milling

Would you, could you, write some notes
On how to tell the oats from groats?
Can you tell from near or far?
Do you need the NIR?

In oat milling, an important unit operation is hulling, in which the fibrous outer hull is forcefully removed from the grain inside, called a groat. There are also many classification steps to ensure the removal of hulls and any remaining unhulled oats from the groats. The number of oats (and hulls, although they are less common) in the final product streams of groats is a key quality metric, and must be manually counted several times per shift. Counted samples are a tiny fraction of mill throughput, and only provide a snapshot of the process at sampling times. Counting is time consuming and can be error-prone and subjective due to the colour and shape similarities of oats and groats.

This chapter assesses the feasibility of an online or at-line machine vision solution for classification as it applies to oat milling. Some background information about oat milling and classification based on colour and near infrared (NIR) images is given in section 4.1. Colour images are shown to be insufficient for the desired oat/groat classification. The instrumentation for NIR imaging, its calibration and the type of data it produces are discussed in section 4.2. Unsupervised classification of the NIR images is outlined in section 4.3, followed by supervised classification in section 4.4. All samples of oats, groats and hulls described in this chapter were provided by PepsiCo Foods Canada's Quaker Oats plant in Peterborough, Ontario. The samples used in sections 4.3 and 4.4 contain mixed, unknown oat cultivars. This is typical in oat milling, i.e. the cultivars are not usually shipped and stored separately. However, several trials of pure cultivars are executed each year at the Peterborough plant, to determine the milling yield of new cultivars. The results impact oat purchasing decisions for the following year. Samples of the pure cultivars tested in 2010/2011 were retained as model validation samples, so that the effect of cultivar on classification could be investigated. Model validation is covered in section 4.5.

4.1 Background

In terms of world production, wheat is one of the largest cereal crops. This is reflected in the volume of literature available on the subjects of wheat production and processing. Oats by comparison, are a relatively small grain crop with an average annual world production less than 5% of that of wheat, based on the years 1994-2003 (Ozaki, McClure and Christy 2007). Unlike wheat, oats are harvested with their hulls intact. The majority of oats are used as livestock feed, and the hulls provide fibre in this application. For human consumption though, the hulls must be removed. Hull removal is a unit operation in oat milling.

Oat milling can be described as four main processes; cleaning, hulling, kilning, and finishing. The cleaning process takes out foreign materials such as stones and other grains. Hulling removes the fibrous hull from the seed inside, which is called a groat. Kilning destroys the enzyme that would cause sprouting. The finishing system consists of many sequential classification steps, to grade the groats according to size and remove any remaining hull pieces. Groats also undergo cutting and/or rolling, depending on their end use.

The operators of an oat mill must perform many manual checks to ensure efficient operation. Because the process takes advantage of gravity for many of the classification steps, the equipment is spread vertically over many different floors and the operators by no means have visibility of the entire process. Typically, equipment provides little feedback to the control room. There are many classification steps involving screens which need to be checked and periodically cleaned with compressed air. The angles of grading screens must be adjusted to ensure optimal separations. By-product streams must be checked to ensure that good product is not being lost, and moisture tests must be performed. One of the most important tasks is counting the number of oats in the groats, as this is a key quality metric.

Sampling and counting is carried out several times per shift for each of the two final product streams of groats. Grade A go directly to rolling, so this check is the last chance to prevent the fibrous hull from entering a retail package. It may be of interest to the reader that discovering an oat hull in one's breakfast cereal is akin to having a popcorn hull stuck in one's teeth; a bit unpleasant perhaps, but not harmful. Grade B are cut lengthwise and then rolled, and a greater quantity of intact oats can be tolerated because the hulls will be cut and aspirated off.

Counting is time consuming and can be error-prone. More frequent data would be helpful for process monitoring, but is infeasible due to the effort required. Counting is not limited to the two finished product streams. It is also a key part of setting up the hulling process, which is absolutely critical to milling yield.

The yield of the milling process can be stated as the weight of oats required to produce a tonne of cut groats. Unlike other grains which are threshed during harvest, oats arrive at the mill with their hulls still attached; therefore, the single largest yield loss is the removal of the hulls. Hulls are removed in an impact huller. Setting up the impact hullers is critical in balancing the opposing goals of high yield and excellent quality. Aggressive hulling means some groats will be broken during hulling. Small pieces and fines are lost to by-product streams. Conservative hulling will result in some hulls remaining attached to the groats, which is a significant quality concern.

This study seeks to determine whether oats can be distinguished from groats by machine vision technology. Besides being useful as a quality check on finished groats, an automated method for distinguishing oats from groats would be very helpful for setting up impact hullers and checking by-product streams for good product. Furthermore, an online system for this task would enable better process monitoring and could serve as feedback for process control.

Classification using Colour Images

Some preliminary work was completed to determine the capability of colour imaging for classification of oats and groats. The Acurum instrument[7] was used for this test. Acurum is intended for analysis of wheat, to aid the user in making better decisions with regards to buying, selling and blending grain by assessing visual grading factors quantitatively. Acurum is a bench top system which meters grain onto a small conveyor and images them a few seeds at a time. Image analysis produces a value for each of many factors on a seed by seed basis, which are then passed to an artificial neural network for classification. (Metzler and Egan 2004) describe the Acurum system in further detail.

In this study, the Acurum system was used to gather image data on a seed by seed basis, but not for classification, since its algorithms are designed for wheat. The result is a large data

[7] Supplier: DuPont Canada

set containing more than 100 feature variables extracted from the images to characterize shape and colour. Several samples of oats and groats (mixed cultivars), each containing thousands of seeds, were imaged in Acurum. This data was analyzed using PLS-DA, described in section 4.4. Colour and shape variables were used both separately and together, but classification was unsuccessful. The data did not provide a clear separation between oats and groats in the latent variable space.

Another automated digital grain inspection system, the Foss Tecator Graincheck, also uses artificial neural networks to analyze digital colour images. (Hall, Tarr and Karopoulos 2003) used this equipment (model 2312), to attempt classification of whole groats, broken groats, and hulls. Oats were originally included as a fourth class, but it was determined that they could not be distinguished from the hulls. Furthermore, the most common misclassification in the test samples was groats identified as hulls, with a misclassification rate of 0.6%. This rate is too high to be used for final product streams.

Classification using Near Infrared Spectra

The use of NIR spectroscopy for analysis of grains and seeds is well established and well documented in the literature. It is a fast, non-destructive testing method which offers many benefits over wet chemistry; little or no sample preparation, no chemicals required, low cost per test and easy installation to name a few. The Canadian Grain Commission began using NIR in 1975 as a method of composition analysis for wheat. Since that time, NIR has become useful for many other grain applications as well, not just in composition analysis, but also in analyzing functionality and in discriminant analysis, or classification. Its use is not common, however, among the lower-volume grain crops. (Ozaki, McClure and Christy 2007).

In literature, there are very few applications of NIR to oats. (Redaelli and Berardo 2007) showed that NIR can accurately predict the fibre composition of oat hulls, and although this is an example of composition analysis, it is of interest here. The impact of oat cultivar and growing location on the various fibre components were tested, and the impact of cultivar was found to play a significant role, with growing location playing a minor role. This relates to one of the goals of this thesis, namely to determine whether oat cultivar affects the NIR spectra enough to impact classification of oats and groats.

Basis of Near Infrared Spectra

Near infrared radiation is located just beyond the red end of the visible spectrum, from 800nm to 2500nm (Ozaki, McClure and Christy 2007). Energy absorption in the NIR region is due to the overtones and combinations of fundamental molecular vibrations. The fundamental vibrations are seen in the mid-infrared region (2500nm-5000nm), where organic functional groups have specific, known locations. This enables identification of molecules by their mid-IR spectra, however, the mid-IR transmitting materials required for instrumentation are relatively expensive (Wilks 2006). Less expensive materials can be used in NIR instruments, but the bands in NIR spectra tend to be broad and overlapping, making interpretation difficult. This is often cited as a weakness of NIR spectroscopy, but these highly correlated bands are easily handled by latent variable methods.

4.2 Instrumentation

The desktop line-scan NIR imaging spectrometer used in this study[8] is shown in Figure 4.1. It contains a spectrograph[9] which is integrated between the objective lens and a matrix detector, in this case, the back of an NIR camera.

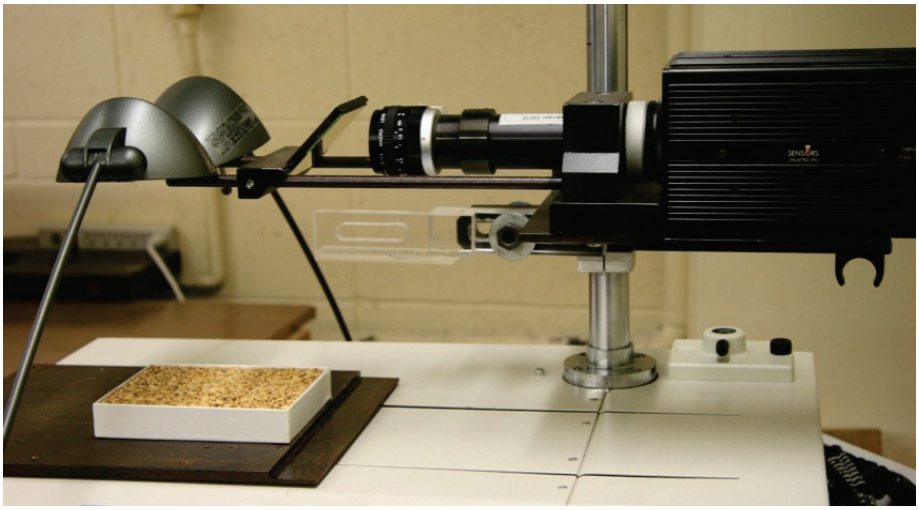

Figure 4.1 Desktop line-scan NIR imaging spectrometer

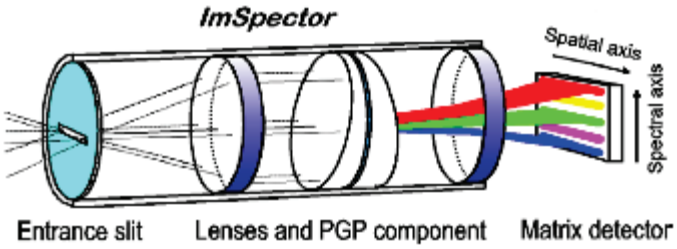

Figure 4.2 ImSpector imaging spectrograph (Specim Spectral Imaging Ltd. 2003)

Figure 4.2 illustrates how the spectrograph separates the NIR wavelengths, using a series of lenses and a prism-grating-prism (PGP) component. There are several available technologies for wavelength isolation; (McClure and Tsuchikawa 2007) provide a good summary. The principle of a PGP component in particular is detailed in (Aikio 2001), and (Hyvarinen, Herrala and Dall'Ava 1998) highlight its benefits in industrial applications. For the purposes of this thesis the key point is that the wavelengths are separated into bands before they reach the NIR camera back, resulting in the simultaneous capture of all of the wavelengths, for every pixel in the scan line. That is, the image captured by the NIR camera has one physical dimension (x) and one spectral dimension (λ). The second physical

[8] Supplier: Technologie d'Avanguardia
[9] Supplier: SpecIm Spectral Imaging Ltd.

dimension (y) is scanned as the sample tray moves across the scan line. This configuration emulates an online monitoring system, as shown in Figure 4.3.

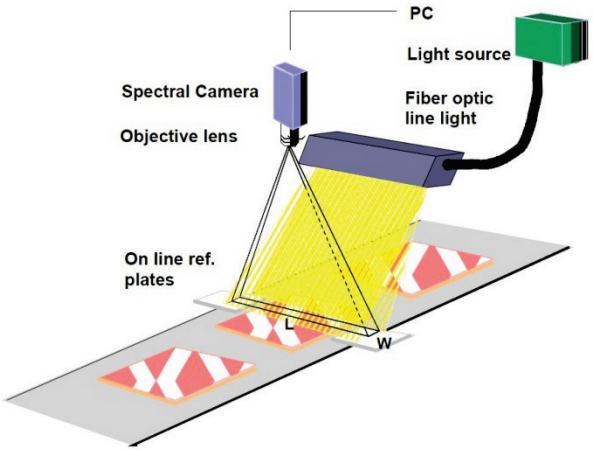

Figure 4.3 Online configuration of a line-scan imaging spectrometer (Specim Spectral Imaging Ltd. 2001)

The resulting data are considered to be hyperspectral images, which are characterized by many (i.e. more than 100) wavelength bands, and the fact that each pixel can be expressed as a spectrum (Grahn and Geladi 2007). An 'xy slice' of the data cube at any given NIR channel is a grayscale image composed of one intensity value per pixel. The hyperspectral data is illustrated in Figure 4.4.

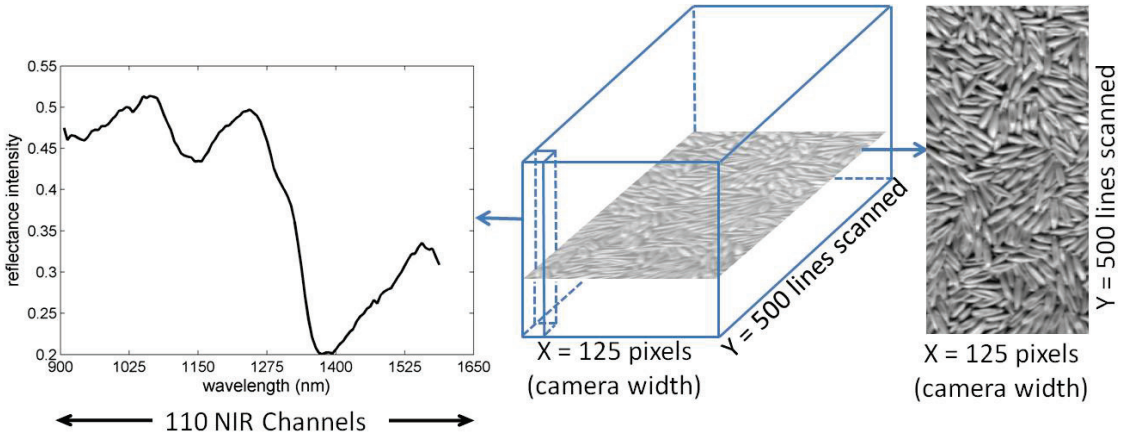

Figure 4.4 Hyperspectral image data topology

The NIR camera (matrix detector, as shown in Figure 4.2) is an array of Indium-Gallium-Arsenide (InGaAs) charge-coupled devices (CCD's), which simultaneously collects data at 128 pixels by 128 wavelengths bands or channels. The 128 channels cover the spectral range 900nm -1700nm, so the spectral resolution is 6.25nm. The camera system (objective lens, spectrograph, and NIR camera back) was positioned as close to the samples as possible, with a resulting resolution of approximately 0.45mm/pixel in the x-direction. The

resolution in the y-direction is determined by the scanner bed speed, and was set to approximately 0.23 mm/pixel. Due to the coarseness of the speed adjustment, this was as close as possible to the resolution in the x-direction. The number of lines scanned is set by the user; 500 lines in this case, in order to capture the desired area. Because of degradation of the sensor array, only 125 pixels and 110 wavelength bands were used, resulting in the (125 × 500 × 110) data array as shown in Figure 4.4. In physical terms, the size of each sample was 116.3mm by 58mm.

For sample illumination, two 60W halogen desk lamps were used. A linear halogen light source as shown in Figure 4.3 was tried, but having two light sources was preferable, as it reduced some of the shadows between grains. Each time the spectrometer was used, the lamps were allowed to warm up for several minutes before proceeding with calibration.

Calibration Procedure

Calibration is necessary, to account for noise that occurs in the data due to inherent instabilities or the age of equipment, differences between the individual InGaAs sensors, and uneven illumination of samples. The calibration used in this thesis is a two step process.

The first calibration step accounts for dark current, a small current present in the CCDs in the absence of light. It also accounts for any drift in the light sources between sessions, and for uneven lighting across the scan line (Hyvarinen, Herrala and Dall'Ava 1998).

In this first step, the user needs to take one scan of 'dark' and one of 'white' at the beginning of an imaging session. The dark and white images have the same dimensions as the InGaAs CCD array. The dark image is taken with the lens cap on, such that no light is present. For the white image, an optically diffuse material with 98% reflectance[10] is used. For each NIR image recorded during the session, the reflectance for each pixel is calculated as

$$r_{xy\lambda} = \frac{s_{xy\lambda} - d_{x\lambda}}{w_{x\lambda} - d_{x\lambda}} \qquad \textbf{4.1}$$

where $s_{xy\lambda}$ is the raw spectral intensity count for one pixel, and $d_{x\lambda}$ and $w_{x\lambda}$ are the dark and white values recorded at the same location on the InGaAs CCD array. $r_{xy\lambda}$ is the reflectance intensity of the pixel at location (x,y,λ).

[10] Supplier: Gigahertz-Optik

Equation 4.1 is called one-point calibration, because only one reference material is used. (Geladi, Burger and Lestander 2004) tested one-point calibration versus linear and quadratic models using four reference materials (2%, 50%, 75% and 99% reflectance standards) and found that the linear and quadratic models were better. The linear and quadratic models compensate for differences between sensors in the InGaAs array, nonlinearities in sensor response, and uneven lighting. The downside to using more reflectance standards is that they are very expensive.

(Z. Liu 2006), (Liu, Yu and MacGregor 2007) demonstrated that shim stock, a less expensive material, was useful as a substitute for multiple reflectance standards when calibrating the same NIR spectrometer used in this thesis. Shim stock is manufactured to specific thicknesses, in different colours, and has a fairly uniform appearance. Table 4.1 shows the thicknesses of the shim stock used, and their corresponding colours.

Shim Stock Thickness	Colour
0.030	Orange
0.025	White
0.020	Yellow
0.015	Pink

Table 4.1 Shim stock thicknesses and colours

Liu's procedure for this second calibration step is as follows: first, an image is taken at several locations in each coloured shim, and its reflectance intensity values are calculated using Equation 4.1. Then an average line image and an average spectrum are calculated for each image. Finally, a linear regression model is fit between the average line images and the average spectra.

In this study, 4 images were taken of each coloured shim, and 400 lines were imaged each time. Each of the 16 images was therefore approximately 93mm × 58mm. Figure 4.5 illustrates the calculation of the average line image and average spectra from the original data cube.

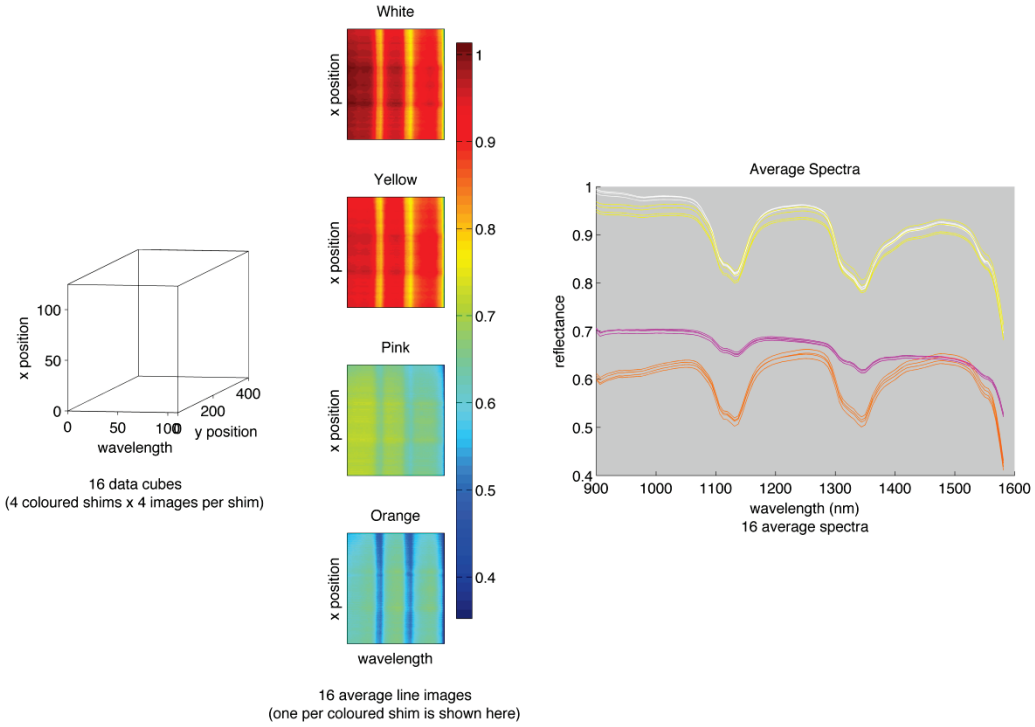

Figure 4.5 An illustration of the calculation of average line images and average spectra. Calculations begin with one data cube per image (left). Averaging along the y-dimension yields an average line image for each data cube (centre). Finally, averaging along the x-dimension yields an average spectra for each data cube (right).

The least-squares regression can be expressed as

$$s_\lambda = \alpha_{x\lambda} + \beta_{x\lambda} \cdot l_{x\lambda} \qquad \textbf{4.2}$$

where $l_{x\lambda}$ is the average line image (over all values of y) and s_λ is the average spectra for a given wavelength. The result is two matrices, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, each having the dimensions $(x \times \lambda)$. These matrices are shown as images in Figure 4.6. They exhibit the same anomalies (near x-position 80) as the calibration matrices presented in (Liu, Yu and MacGregor 2007).

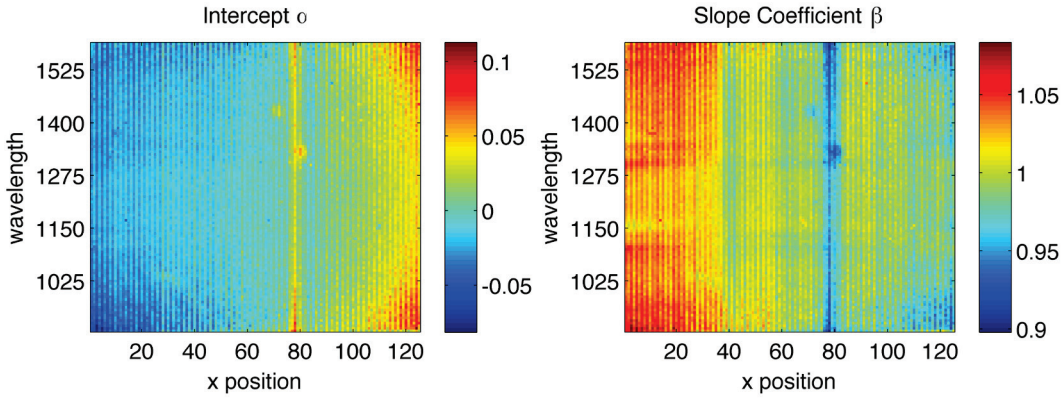

Figure 4.6 Intercept and slope matrices resulting from Equation 4.2

These matrices can be used in Equation 4.3 to correct the reflectance intensities of NIR images as calculated previously in Equation 4.1.

$$r_{xy\lambda,corrected} = \alpha_{x\lambda} + \beta_{x\lambda} \cdot r_{xy\lambda} \quad \textbf{4.3}$$

Note that if the dark current had not been subtracted from the coloured shim images, its effect would be included in the **α** matrix.

Figure 4.7 demonstrates the ability of this second calibration step to reduce the pixel-to-pixel variation introduced by the InGaAs CCDs. The grayscale image of the orange shim (before correction) at approximately 1200nm shows streak lines in the scanning direction, indicating a difference between CCDs in the x-direction. This difference is illustrated by the spectra at two locations in the shim. After correction, the grayscale image looks much more uniform, and the spectra of the two pixels are nearly identical.

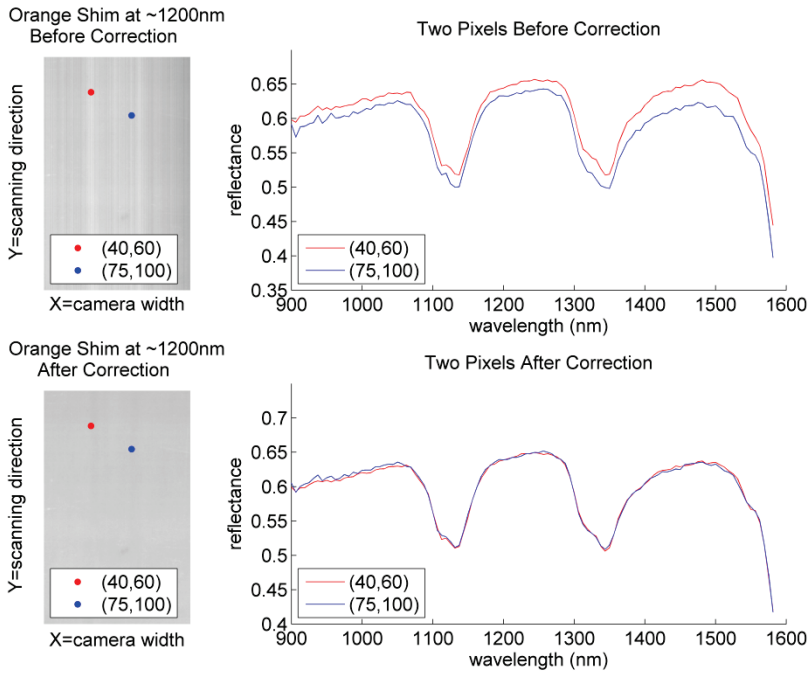

Figure 4.7 Orange shim at 1200nm, and two pixel spectra, shown before and after correction using α and β matrices

Data Unfolding and Preprocessing

As shown previously in Figure 4.4, the hyperspectral image data forms a cube. This cube must be unfolded for model building, to form a two-dimensional array. Each row in the array will contain 110 intensity values, one for each wavelength band. Thus, the matrix can

be thought of as a list of spectra, where each list entry pertains to a specific pixel position in the 'xy slice'.

In model building terms, each pixel is an observation and the intensity at each wavelength band is a variable. Hyperspectral imaging generates huge data sets. The images presented in this chapter contain 125 pixels wide × 500 scanned lines = 62500 pixels (observations) and 110 wavelengths (variables).

4.3 Unsupervised Classification of NIR Images

Some exploratory work was completed prior to building a classification model. Four samples were collected at convenient sampling points in the mill, during a typical production run (i.e. mixed, unknown oat cultivars). Samples included Grade A groats, which are a product stream, hulls, which are a by-product stream, and two mid-process streams of oats, called large and stub. The oats are classified by size into large and stub streams prior to hulling because the hullers need to be set up differently for each stream to improve hulling efficiency and reduce yield loss. Both sizes of oats were collected because (Ozaki, McClure and Christy 2007) reported differences in NIR spectra due to seed size. The spectra of the four samples were compared, by inspection and using principal components analysis (PCA). Figure 4.8 shows the shapes of the spectra, for a sample of pixels selected at regular intervals across the x- and y-directions.

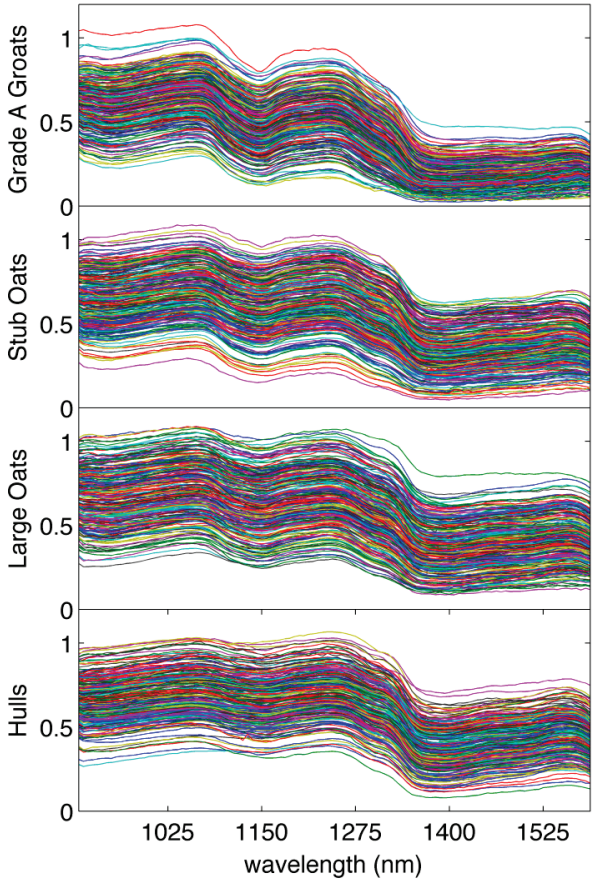

Figure 4.8 NIR spectra of exploratory samples (mixed cultivars)

PCA

PCA is a latent variable method, similar to PLS which is described briefly in 2.1. Where PLS is a regression method relating **X** and **Y** matrices, PCA is a projection method for just one block of data, usually termed **X**. The matrix form of the projection equation is

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{P}^{\mathrm{T}} + \mathbf{E} \qquad \textbf{4.4}$$

Recall from section 2.1 that the **T**-matrix contains 'scores', or each observation's coordinates in latent variable space. Each latent variable (column of **T**) is a linear combination of the **X**-variables; the coefficients of which are stored in **P**. Therefore the values in **P** explain the importance of each **X**-variable to each latent variable, as in

$$\mathbf{T} = \mathbf{X} \cdot \mathbf{P}^{\mathrm{T}} \qquad \textbf{4.5}$$

PCA extracts latent variable components ($\mathbf{t}_1$,$\mathbf{t}_2$ etc.) that explain the greatest amount of variance in the **X**-matrix. The first component extracts the linear combination of wavelengths that exhibit the greatest variation. The second component extracts a linear combination that is orthogonal to the first component and exhibits the next greatest variation. In determining the latent variable directions, PCA has no influence from a **Y**-matrix. When applied to a classification problem, this is called unsupervised classification, because the model is not built with advance knowledge of the class groups. Even if the class groups are known, unsupervised classification can offer insight into if and how the observations naturally fall into clusters.

Model Building

This analysis was completed using MACCMIA[11]. The four samples discussed above were combined into one composite image, shown in Figure 4.9 as a colour image. The NIR data from all four samples was formed into a composite image the same manner and then unfolded into a two-dimensional matrix. As an unsupervised classification, this PCA model built from the composite NIR image will indicate whether the stub oats and large oats are in fact, separate classes in the NIR spectra. It will also show the distance between clusters which suggests the relative ease of classification.

[11] MACCMIA is an image analysis software package freely available from the McMaster Advanced Control Consortium (MACC) at McMaster University.
http://macc.mcmaster.ca/research/software/maccmia

Figure 4.10 shows the first four principle components as grayscale images. When calculating a PCA model for multivariate image analysis, the data is not usually mean-centered and scaled, therefore the $\mathbf{t}_1$ scores measure each pixel's distance from the mean. $\mathbf{t}_2$ captures the next largest direction of variance, which in this case separates the groats from both sizes of oats and the hulls. The meaning of $\mathbf{t}_3$ is unclear and $\mathbf{t}_4$ captures a slight difference between the hulls and the other three samples.

Figure 4.9 Composite colour image, containing four mixed-cultivar samples

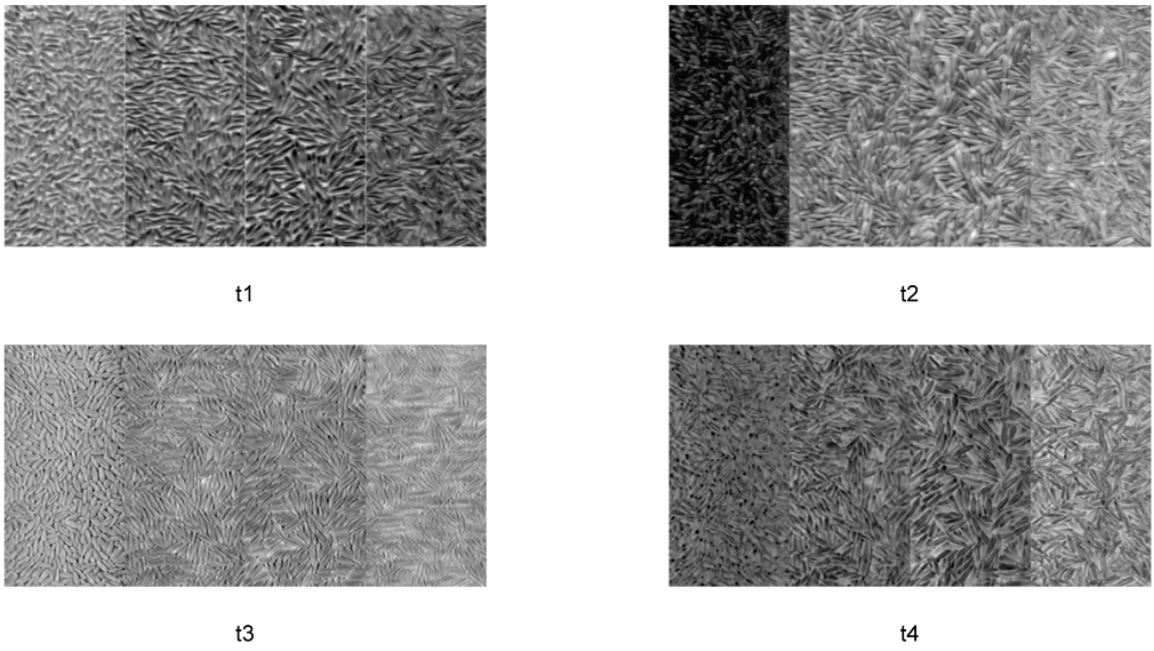

Figure 4.10 The first four principle components shown as grayscale images

Figure 4.11 shows the loadings for the same four principle components. Loadings express the importance of each wavelength band in describing the each latent variable. For example, the second component, which separates the groats from the other three classes, is strongly influenced by the wavelengths above 1300nm and below 1100nm, but not so much by the middle wavelengths.

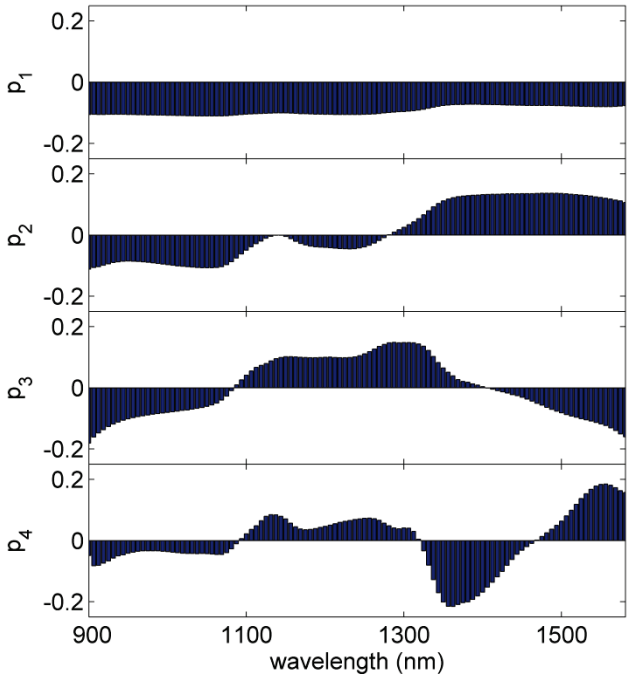

Figure 4.11 Loadings for the first four principle components

The PCA indicates that there are only two main classes in the data. The best two dimensions that demonstrate the class separation are $\mathbf{t}_2$ and $\mathbf{t}_4$, and the $\mathbf{t}_2$-$\mathbf{t}_4$ score plot is shown in Figure 4.12 (top right). Recall that a score plot displays the distribution of observations in a plane of the latent variable space. Figure 4.12 (top left) displays the $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_4$ scores as colours (red, green, and blue respectively) to provide a false colour composite image in the image space. When a white mask is applied to the score space plot, the pixels located under it are displayed as white in their corresponding locations in the image space. Through the use of masks it can be seen that the smaller, left cluster contains the groat pixels, while hulls are grouped together with oats in the larger, right cluster. Within the right cluster, hulls fall to the far right, whereas oats take up the space between the groats and hulls. This indicates that the NIR energy doesn't penetrate the hulls very deeply, and therefore the spectrum of an intact oat is influenced more by its hull than its groat inside. Further work, such as multi-dimensional masking (Liu, et al. 2005), could be pursued using the PCA model, but because the class memberships are known, supervised classification will be explored.

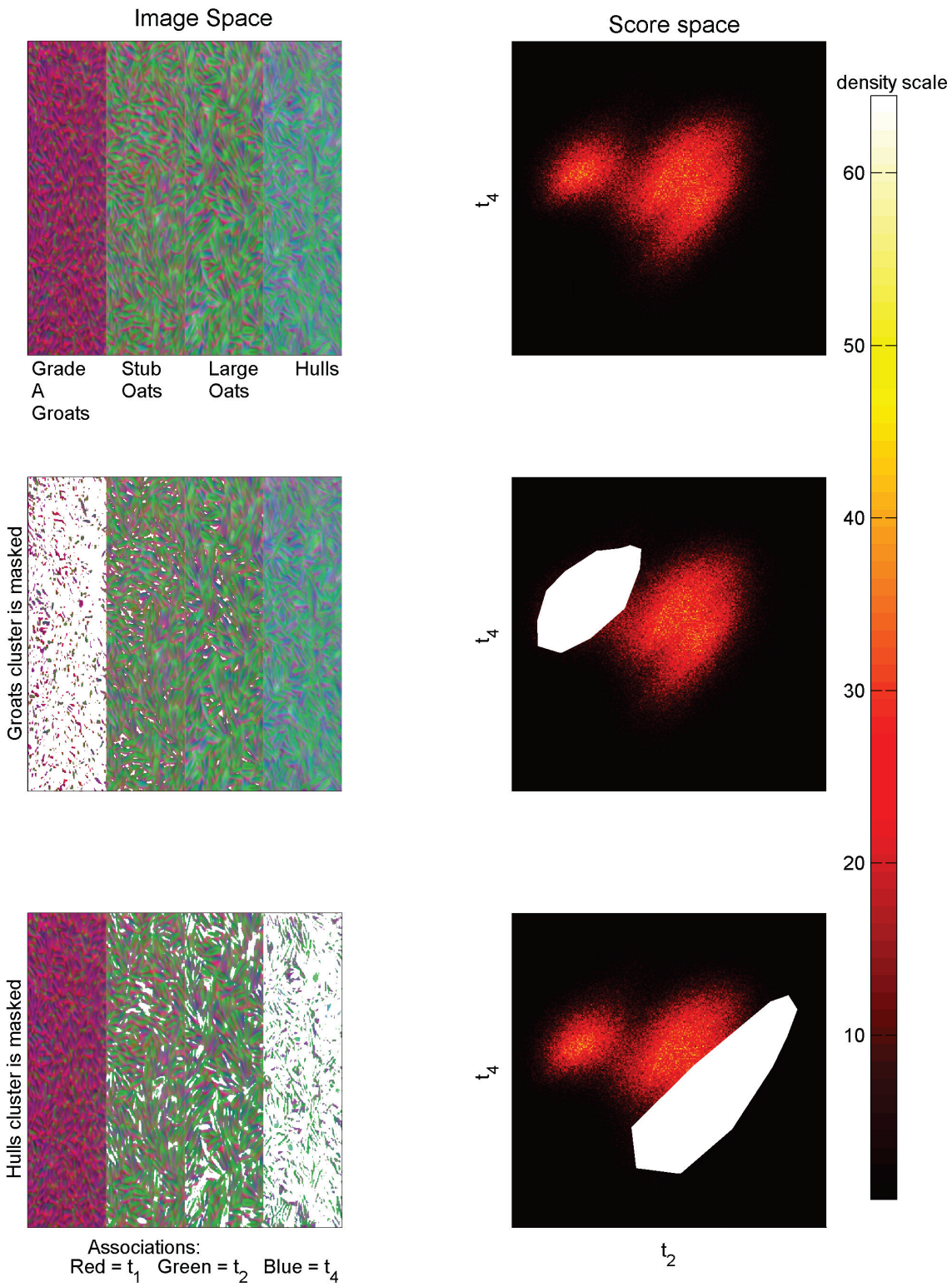

Figure 4.12 Masks shown in the latent variable space (right) and in the image space (left). It is clear that there are two main classes in the data; groats and hulls/oats. The image space is shown in false colours; the first principle component is shown as red, the second as green, and the fourth as blue. The score space plots are coloured by pixel density.

4.4 Supervised Classification of NIR Images

The motivation for classification is keeping oat hulls out of finished product, whether they are loose or part of an intact oat. Because the exploratory work showed that hulls are in some sense a subset of the oats class, it makes sense to proceed with just two classes. This section describes supervised classification using two of the four samples described in section 4.3, the Grade A groats and the large oats. They will be referred to as the training samples. The stub oats and hulls were reserved to be used as validation samples.

PLS-DA

PLS-DA is a variation on PLS, which performs discriminant analysis, or the prediction of class membership for each observation. PLS-DA is a supervised classification method, meaning that the class membership of each pixel is known before the model is built. Whereas PCA illustrates whether or not the data naturally fall into clusters, PLS-DA orients the latent variable directions to accommodate the best separation between the clusters (classes). These known class memberships form the **Y**-matrix of the model.

There are as many **Y**-variables as there are classes, and they are all binary variables whose values are one if the observation belongs to the class and zero otherwise. Other than this specific form for **Y**, building a PLS-DA model is exactly the same as a PLS model. In this case study, each pixel is an observation, so the model will predict the class membership pixel-wise. Although the **Y**-matrix is binary, the model predictions are continuous, with values ranging from below zero to above one. A threshold value is applied to the prediction values to convert them back into binary values.

Model Building

Recall that the training samples were collected during regular production (i.e. mixed cultivars). They were hand sorted to ensure their purity as best as possible visually (i.e. no oats in the groats and vice versa), and imaged according to the procedure laid out in section 4.2. The primary calibration was applied, followed by a low-pass filter to eliminate single pixel outliers. Then, the secondary calibration step (using the α and β matrices) was completed, followed by data unfolding. PLS_Toolbox[12] in Matlab[13] was used to build a PLS-DA model and also to calculate the threshold that best separates the two classes. Figure 4.13

[12] Supplier: Eigenvector Research Incorporated
[13] Supplier: The Mathworks Incorporated

shows the PLS-DA model coefficients in relation to the spectra of the training samples. The three vertical lines are the wavelengths used by the Spectral Scanner software[14] to generate a false-colour preview of the data.

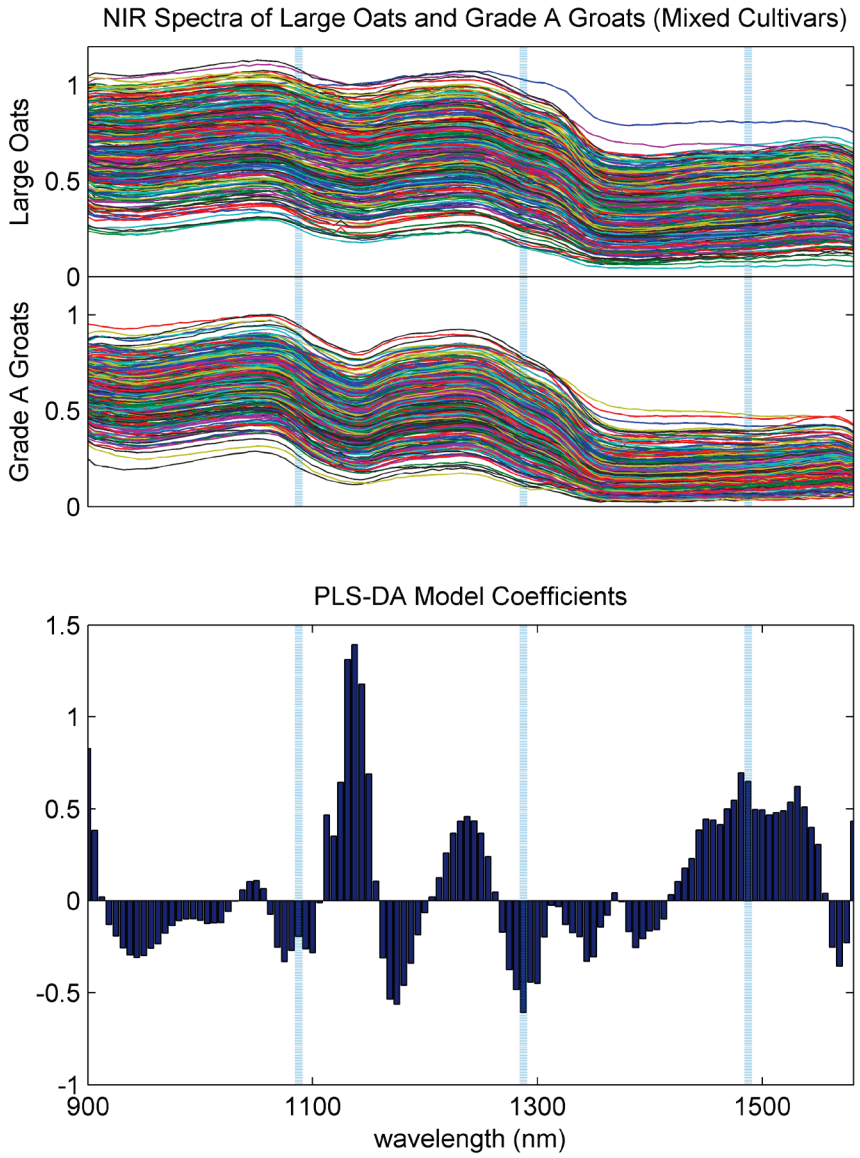

Figure 4.13 Top: NIR spectra of the training samples for a sample of pixels selected at regular intervals across the x and y directions. Bottom: The PLS-DA model coefficients. The three vertical lines represent the wavelengths used to generate the false-colour 'Spectral Scanner Preview' image in Figure 4.15

The model generates a prediction value for each pixel. Histograms of the prediction values are shown in Figure 4.14. This demonstrates that the oats and groats are nicely separated by the PLS-DA model, with a very slight overlap. The green curves shown are normal

[14] Supplier: Technologie d'Avanguardia

distributions fitted to each class. Their intersection is the threshold that is applied to the prediction values to determine class membership.

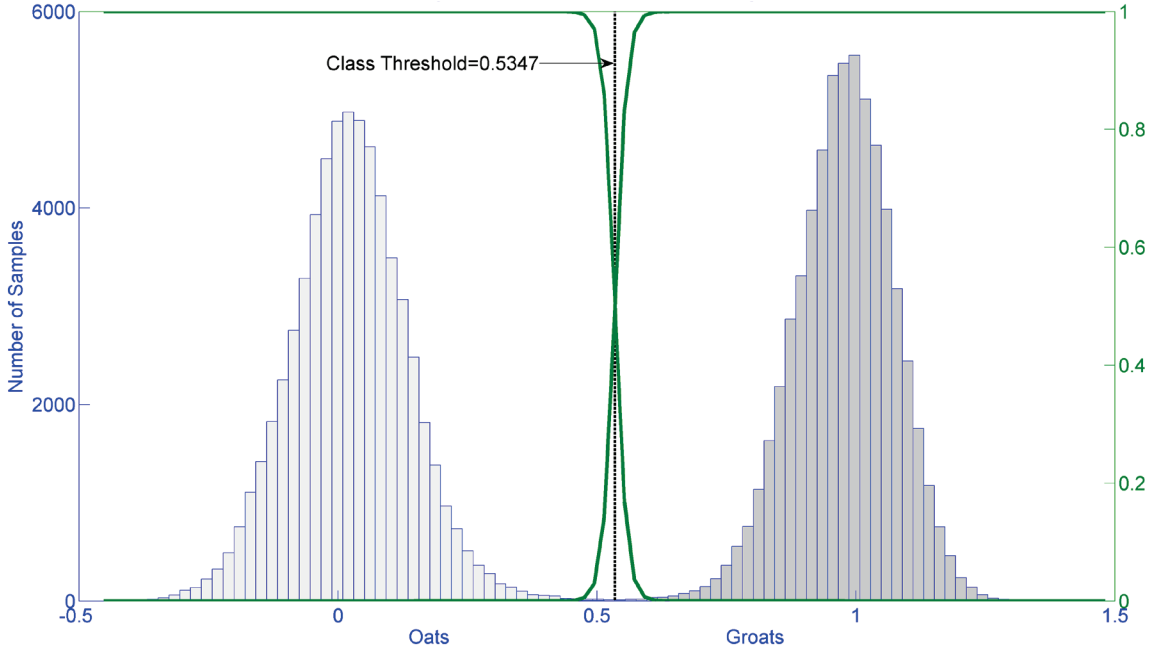

Figure 4.14 Histrogram depicting the prediction values for each pixel in the training images

The training samples are shown in four different image modes in Figure 4.15. The colour images demonstrate how similar the two classes are in colour, shape, and size.

The Spectral Scanner preview images are false colour images based on three wavelengths (~1095nm, ~1295nm, and ~1495nm) (D'Agostini 2011), shown in Figure 4.13. These images are visible to the user during scanning, and they were often helpful in identifying oats and groats that were missed during manual sorting. If a user was able to choose which wavelengths to use in the preview images, the one at ~1095nm should be moved to ~1150nm (the largest peak in the coefficients plot in Figure 4.13) in order to best display any sorting mistakes such that they might be removed and the image retaken.

The prediction images are simply that; images made up of the prediction values for each pixel. After the calculated threshold is applied, binary images are the result. A perfect oat/groat classification would produce an entirely white image for the oats and an entirely black image for the groats. The classification shown in Figure 4.15 appears to be slightly less than perfect. However, note that the clusters of dark pixels in the binary prediction image for oats correspond to oats where the hulls are split or the groat is protruding from the end

of the oat. These features are visible to the eye when viewing the sample closely but are difficult to see in the corresponding colour image. These oats could be removed and the image re-taken, but a few pixels from the opposite class do not impact the PLS-DA model very much. There are 26 pixels in the oat image that have been classified as groat pixels, or 0.042%. There are eight pixels in the groat image that have been classified as oat pixels, or 0.013%.

This is a very successful classification, especially given that the 'misclassified' oats have known causes. Implementation of these methods could be accompanied by some post-processing of the binary images to determine the sizes of misclassified clusters and filter out pixel clusters that are too small to be of concern.

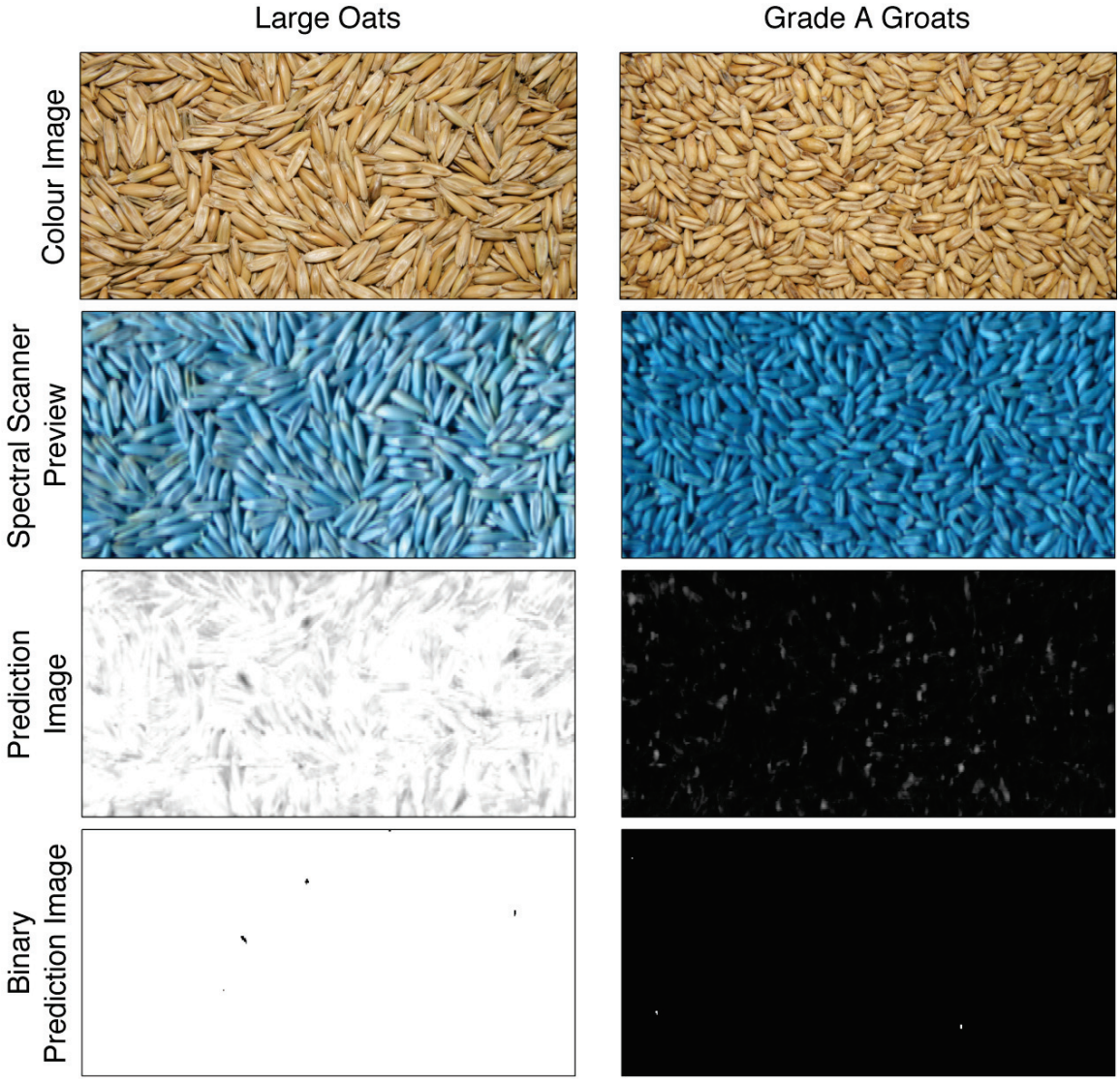

Figure 4.15 Training sample images, shown in four image modes

4.5 Model Validation, Mixed Cultivars

Of the four samples used in the exploratory work described in section 4.3, two were used as training samples for the PLS-DA model in section 4.4. In this section, the remaining two samples, stub oats and hulls, are used as model validation samples, and Figure 4.16 shows their PLS-DA results alongside those of training samples.

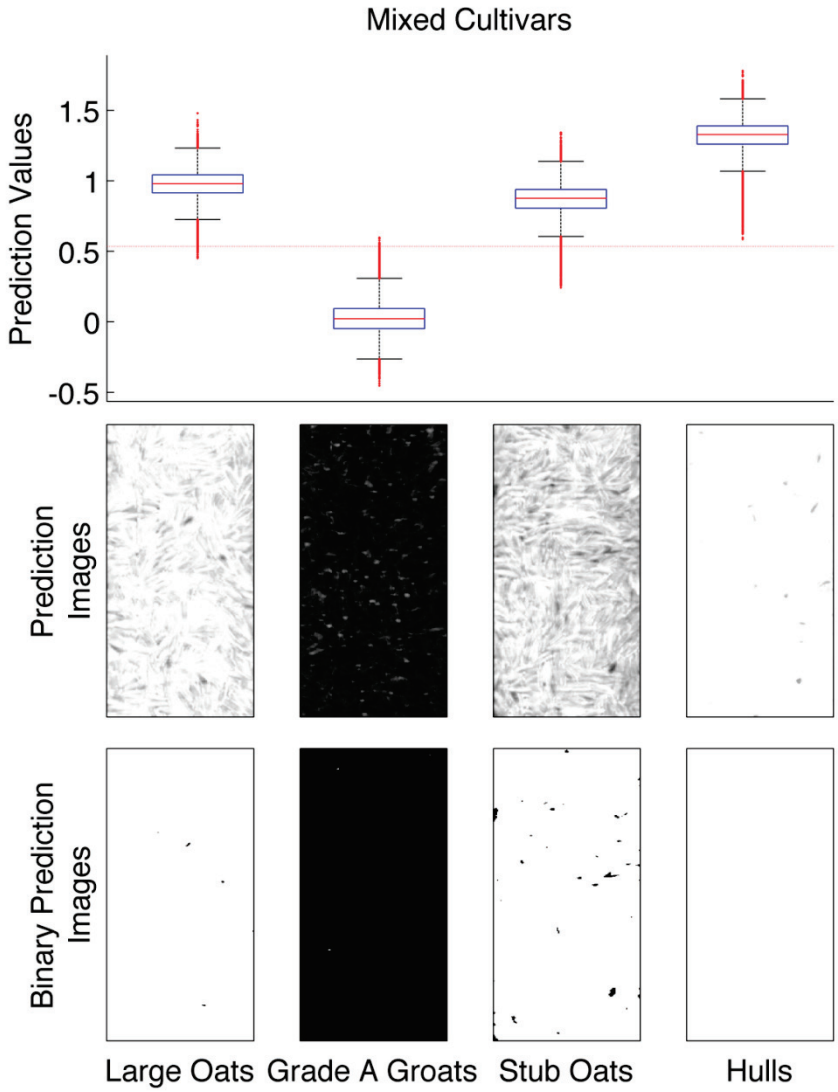

Figure 4.16 PLS-DA results for the four mixed -cultivar samples

These results validate the fact that the hulls are further from the groats, in the latent variable space, than they are from the oats. It also shows that the stub oats are slightly closer to the groats than the large oats (based on the fact that the prediction image is darker

overall for stub oats than for large oats). In the stub oats sample, there are several clusters of 'misclassified' pixels (i.e. black areas, assumed by the model to be groats), but they are not all actually misclassified. Figure 4.17 displays an enlargement of the binary prediction image for stub oats along with its colour image. Cluster A contains a groat, partially buried under the top layer of oats, and an oat that has split open, exposing the groat to the NIR camera. Clusters B, and C are also split oats. Cluster D is a groat that was missed during manual sorting. Some of these features can be seen by looking closely at the colour image; others were evident only when viewing the sample directly. It was not possible to assign a cause to every cluster; reasons for the smaller clusters were not evident. They may be groats or split oats partially hidden in the second or third layer of oats.

Figure 4.17 Stub oats binary prediction image (top) and colour image (bottom)

4.6 Model Validation, Pure Cultivars

Each year, pure cultivar trials take place in the mill. Specific oat cultivars are tracked, shipped, and stored separately from field to mill, so that they can be processed separately and evaluated for their milling yield. These trials provided a nice opportunity to collect validation samples for the classification model, and determine whether cultivar, moisture level, or growing location would affect the classification. Cultivar and growing location were known ahead of time, but the moisture level is a disturbance variable. Sample sets of three cultivars, called 'Nice', 'Dancer', and 'Sherwood', were retained as validation samples for NIR analysis. Three sets of Sherwood samples were retained; one grown in northern Ontario and two grown in southern Ontario; for a total of five sample sets as shown in Table 4.2.

Sample Set	Cultivar	Growing Location	Figure
1	Sherwood	Southern Ontario	Figure 4.19
2	Sherwood	Southern Ontario	Not shown
3	Sherwood	Northern Ontario	Figure 4.20
4	Nice	Unknown	Figure 4.21
5	Dancer	Unknown	Figure 4.21

Table 4.2 Cultivar, growing location, and figure numbers for validation samples

Each sample set contained eight samples, taken from specific sampling locations in the oat mill. The sampling locations were chosen such that seed size and moisture variations would be represented for both oats and groats. This is an example of indirect design in the latent variable space, meaning that the design factors are secondary variables that could influence the **X**-variables (spectral values). Indirect design is common in spectroscopy and multivariate calibration applications (Wold, Josefson, et al. 2004). Figure 4.18 illustrates the sampling locations along with the moisture level and density for each sample.

The amount of preparation work for these model validation samples was tremendous; they were manually sorted to ensure purity, which is a very time-consuming task. Then, as they were imaged, the 'image reconstruction preview' (visible in the Spectral Scanner software) was inspected and it was often evident where an oat or groat had been missed. In that case, the offending grains were removed from the sample with tweezers, and the image was re-taken. Colour images were taken at the same time, to be used when analyzing the prediction results.

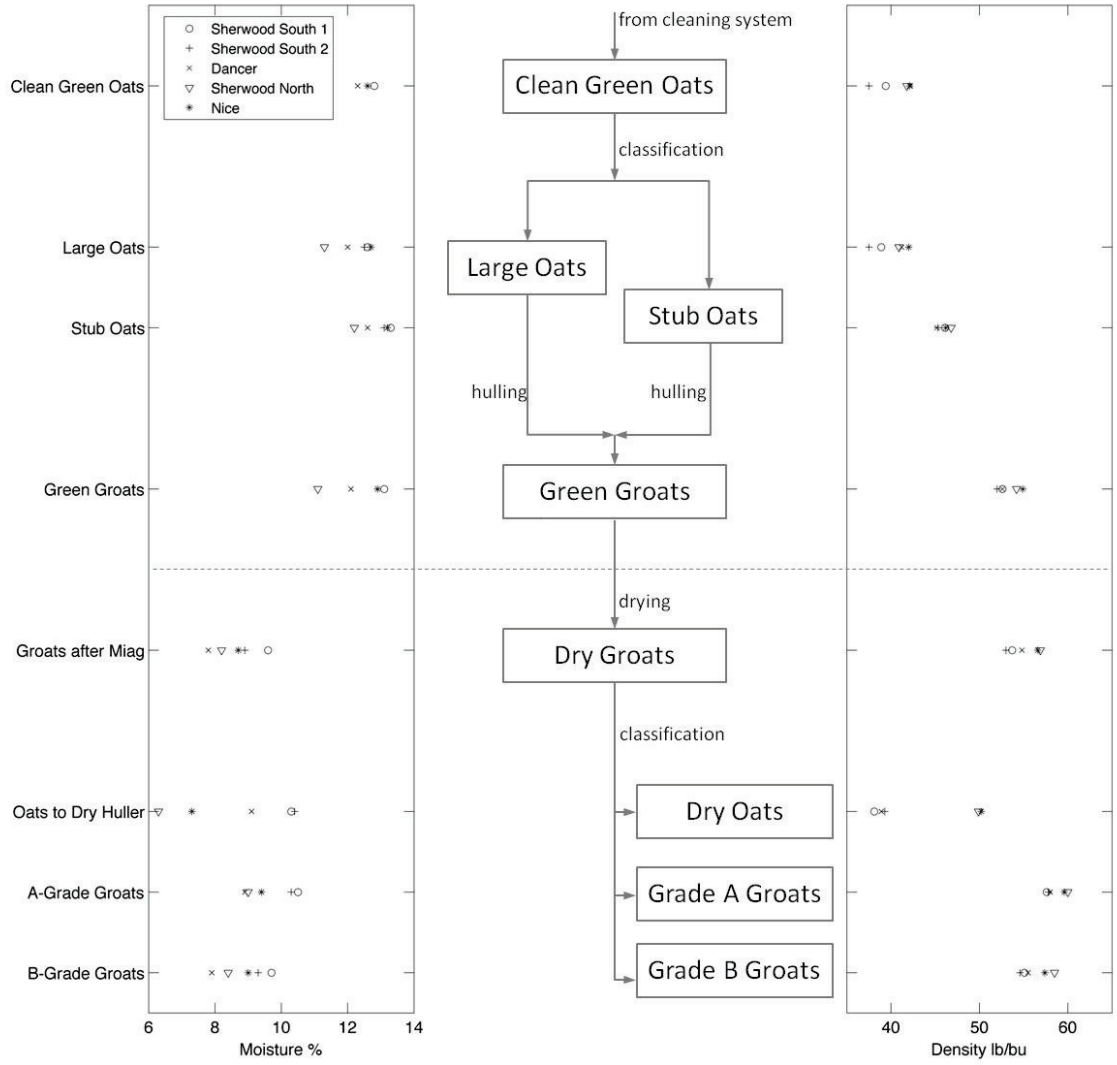

Figure 4.18 Sampling locations for validation samples, together with their moisture and density data. The sampling locations were chosen such that seed size and moisture variations would be represented for both oats and groats

For the first set of samples collected, Sherwood South 1, all eight samples were sorted and imaged. The model predictions are shown in Figure 4.19.

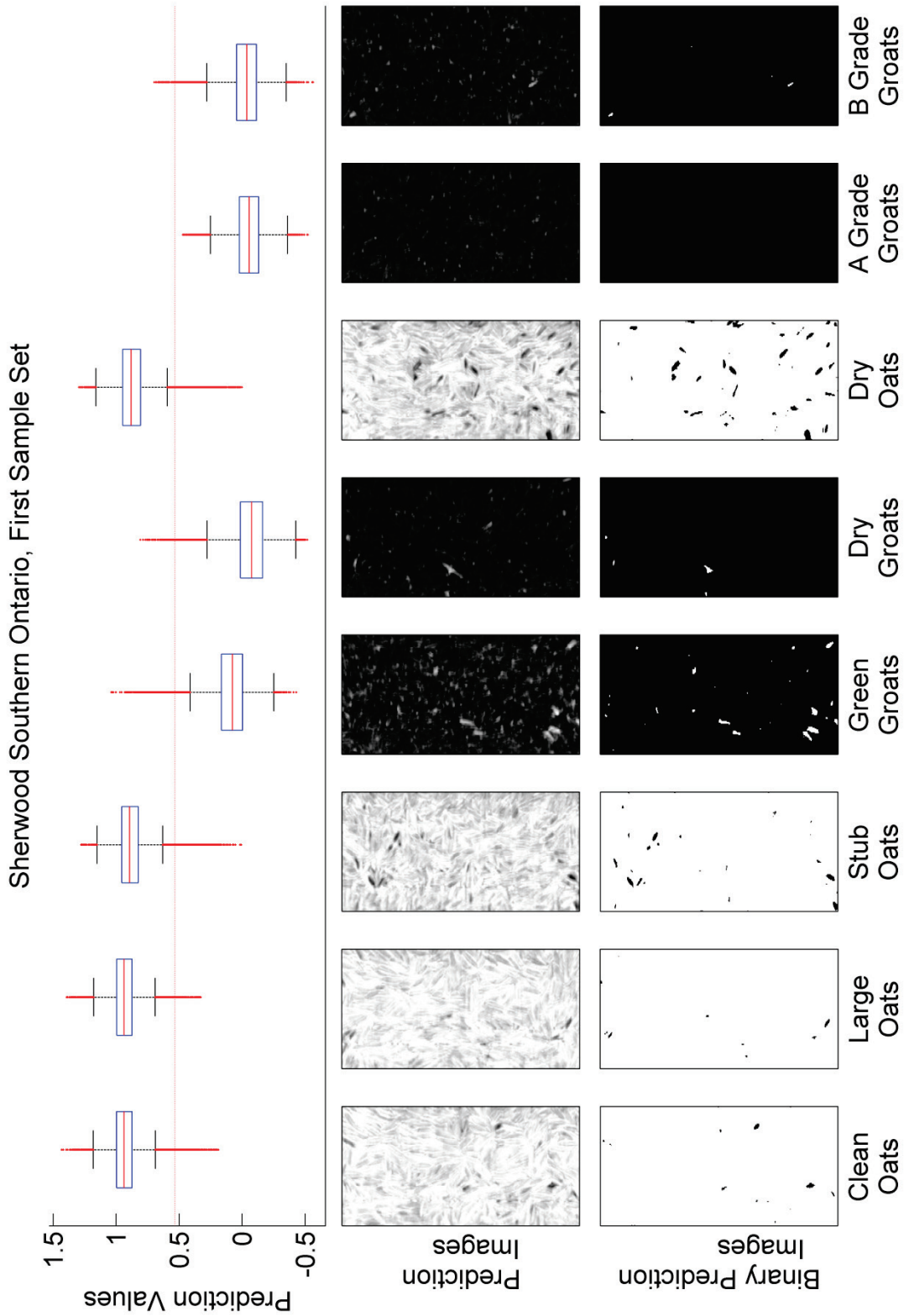

Figure 4.19 PLS-DA predictions for Sherwood Southern Ontario (First Sample Set)

Overall, the classification looks quite good, indicating that the model built using mixed-cultivar samples works well for the Sherwood cultivar. The second set of Sherwood Southern Ontario samples showed very similar results, therefore their prediction values are omitted here.

Most of the binary prediction images contain at least a few misclassified pixels. Each binary prediction image was compared with its corresponding colour image to determine the reason for the misclassifications. Where there was a clear cause for a misclassification, it was recorded. Nearly all of the larger clusters of misclassified pixels were justifiable misclassifications. Some were identified as either oats or groats that were missed during manual sorting. Among the clusters identified as groats by the PLS-DA model, most were actually oats, split or otherwise damaged to expose the groat inside.

The dry oats sample displays the most misclassifications. The process stream where the dry oats are sampled is meant to contain dry oats that are on their way to the 'dry oats huller' for a second chance at hulling. In practice, this stream contains up to 60% groats. Manual sorting of samples from this stream was therefore extremely time consuming – worse by an order of magnitude than samples from other locations. On close inspection, most of the misclassified clusters are in fact split oats.

The effect of seed size is minimal, confirming the results of the exploratory work (section 4.3). This can be seen by comparing large and stub oats. Stub oats show a few more misclassifications than large oats; on inspection the majority of those are also split oats. There is no significant difference between the PLS-DA prediction values of A grade and B grade groats even though B grade contains more small groat pieces. The misclassified clusters in the B grade image were identified as a broken, discoloured groat, and a very thin oat.

As for the effect of moisture, if there was one, it would be shown best by the comparison of green groats and dry groats. The validity of comparing oats before and after drying is somewhat negated because of the number of split oats in the dry oats sample. Comparing the prediction images of green groats and dry groats, the green groats sample has more misclassifications, but all of the larger white clusters in the binary prediction image were identified as oats or hull fragments. The largest white cluster in the dry groats binary image was identified as an oat, partially hidden under the top layer of groats. Apart from these

justifiable misclassifications, the two binary images are very similar. Looking back to the grayscale prediction images and the boxplot though, the green groats and dry groats do look different, with the green groats exhibiting many tiny whitish flecks. Inspecting this sample very closely reveals that these whitish flecks are the fuzzy ends of each groat. The same is true in the dry groats sample, but they have less fuzz. Therefore it cannot be concluded that the drop in moisture during drying is the reason for the difference in prediction values.

Eliminating seed size as a factor allows the elimination of four of the eight sampling locations. For the remaining sample sets, the following samples were retained: clean oats, green groats, dry oats, and grade A groats. These samples represent both oats and groats before and after drying. The following figure displays prediction results for the Sherwood North samples, which were generally lower in moisture and higher in density than Sherwood South (see Figure 4.18 for this comparison).

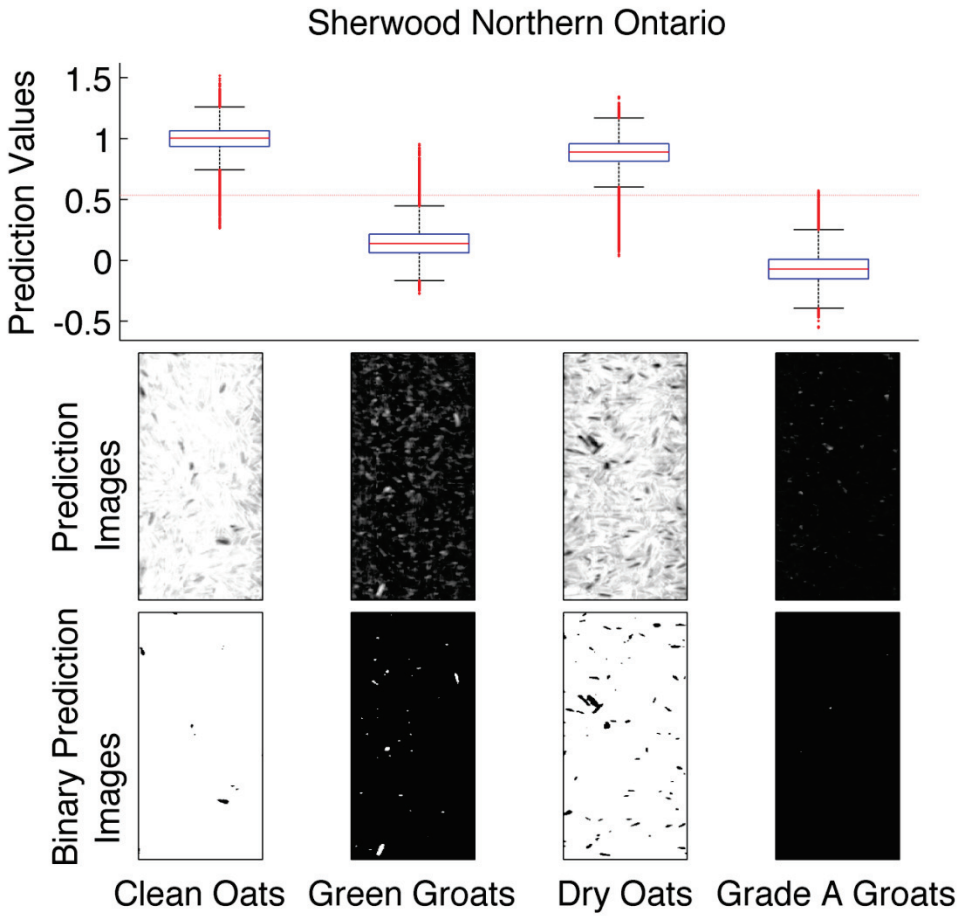

Figure 4.20 PLS-DA predictions for Sherwood Northern Ontario

Again, the dry oats sample shows the most misclassified clusters, and again, the large clusters were found to be justifiable misclassifications, with the majority of them due to oats with split hulls and exposed groats. The larger clusters of white pixels in the green groats image were again identified as mistakes in manual sorting, and the larger clusters of black in the clean oats image were either groats that were missed, or oats with split hulls.

Because of the large number of oats with split hulls in the dry oats, imaging further samples of dry groats was judged to be irrelevant. Prediction results for the final two cultivars, Nice and Dancer, are shown in Figure 4.21.

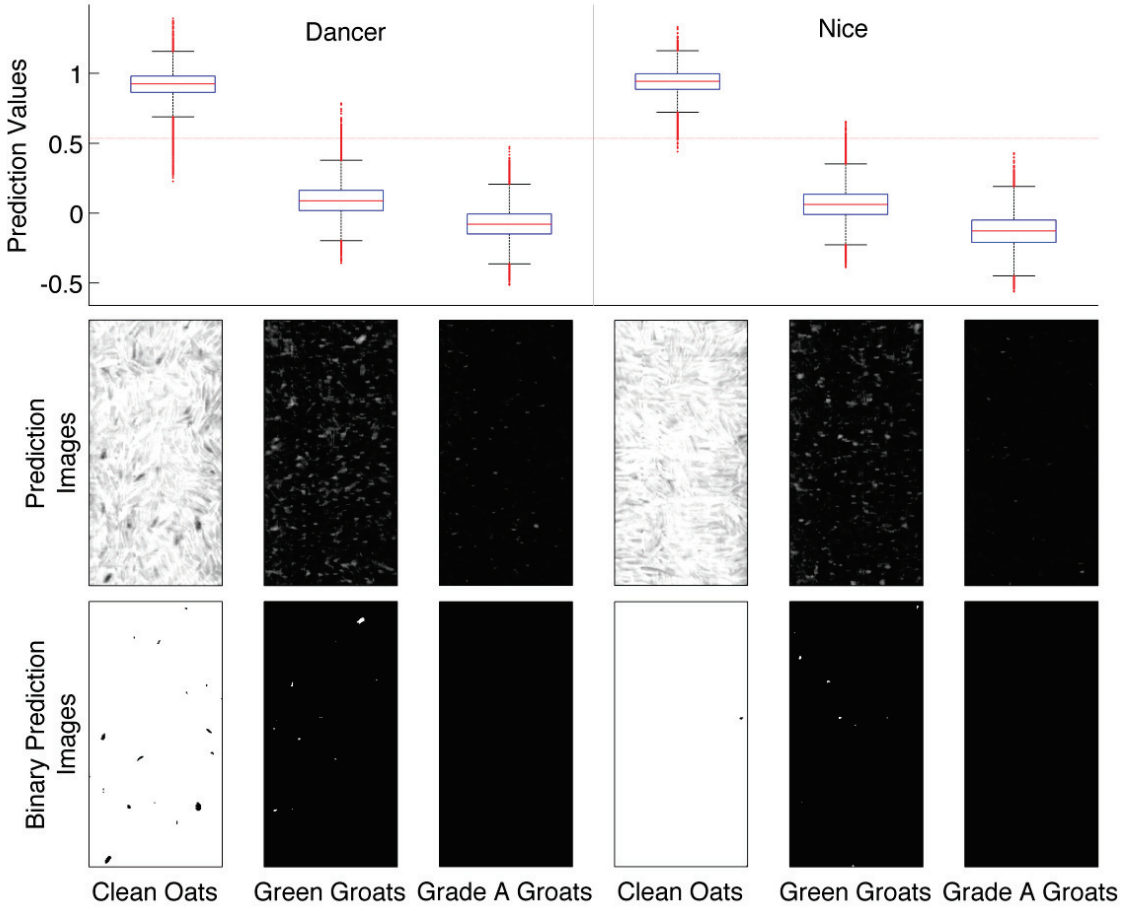

Figure 4.21 PLS-DA predictions for Dancer and Nice cultivars

Similar to previous samples, the larger clusters of misclassifications were found to be oats with split hulls or errors in manual sorting. The Nice samples were the easiest to sort manually; each of them was very pure.

4.7 Discussion and Recommendations

Overall, the PLS-DA based on mixed-cultivar samples provided good classification of oats and groats of all the cultivars. Variations in cultivar, growing location, and moisture, to the extent that they were included in the validation samples, do not pose a threat to the possibility of one robust classification model that would handle those variations. This is good news for the feasibility of a machine vision solution for classification.

On the other hand, there were many misclassifications. It was possible to identify many of the larger clusters of misclassified pixels as justifiable misclassifications, by referring to the colour images. The majority were due to split oats with exposed groats or human error during manual sorting. Recall that any mistakes that were obvious in the false-colour preview shown by the scanner software were removed and the image re-taken. Even so, the number of human errors discovered by the PLS-DA model was considerable.

Based on the analysis presented in this section, it appears feasible to use NIR imaging and a PLS-DA model for checking final product streams of groats. Oats and hull fragments are easily identified by their NIR 'signatures'. It also appears feasible to find groats and broken groat pieces in the hulls by-product stream. Either of these analyses could be carried out offline or online. Some post-processing of prediction images would be required, and for the final product streams, rules would need to be established as to the minimum pixel cluster size (i.e. minimum hull fragment size) that would be considered a quality concern. The issue of split oats with partially exposed groats would be taken into account with this minimum cluster size, because a split oat by its nature will nearly always present at least a sliver of its hull to the camera. For the by-product stream, rules would need to be established as to the minimum pixel cluster size that would be considered a valuable groat fragment. The types of post-processing required are standard image analysis procedures available in commercial software. (The Mathworks, Inc. 1994-2011, Example 2) shows some of these procedures applied to images of grains of rice. (The Mathworks, Inc. 1994-2011, Examples and Webinars) demonstrates the capabilities of Matlab's Image Processing Toolbox.

Mid-process classification is muddied by the existence of oats with split hulls that present partially exposed groats to the NIR camera. As such, it would be difficult to use an NIR instrument to obtain an accurate estimate of the percentage oats and groats in a stream that contains both. For a stream such as green groats, which is intended to contain mostly

groats, an online instrument could alert operators to significant shifts in the number of oats and hulls in the stream. An operator could then visually inspect a sample to determine the correct course of action.

Chapter 5 Conclusions

For PLS and PCA
And also PLS-DA,
Oats and muffins were the test
And each of them has shown success.

Two case studies, which each address a genuine challenge in the food industry, have been presented. In both cases, a specific set of circumstances were addressed using latent variable methods.

Chapter 2 introduced the techniques encompassed by rapid product development using latent variable methods. The goal of these techniques is to quickly develop a new product having specified properties by using existing data and a few carefully selected experiments.

In Chapter 3, rapid product development was applied to the reformulation of frozen muffin batters. Two types of PLS models were presented; a more traditional mixture model and a modification on Muteki's mixture-property model (Muteki, MacGregor and Ueda 2006). While both types of models were shown to have good predictive ability, Q^2Y was higher by 10% for the modified mixture-property model. However, it contained some empty regions in the latent variable space, so designed experiments were executed to enhance the model. Subsequently, optimization was used to generate new recipes that reduced the values of a specific quality attribute, AOI, while maintaining as best as possible the taste, texture, and appearance of the original products. New recipes were created for four of the 26 original formulas, and these were made up in the laboratory and analyzed. AOI reduction was very successful for all four formulas while the maintenance of the other muffin characteristics was accomplished to a moderate degree. The positive outcomes achieved in this project demonstrate the great potential of using rapid reformulation for food products; in fact, the project is ongoing and the suggestions put forth in section 3.7 will be integrated into the next phase.

Chapter 4 discussed some of the challenges of quality control in oat milling. Manual assessment of final product streams (counting the number of oats in the groats) is a time consuming and therefore relatively infrequent task. Consequently, statistical process monitoring is not practical. The case study explored the feasibility of using NIR imaging combined with hyperspectral image analysis to classify oats and groats. A PCA model was

employed to explore the image data and subsequently, a PLS-DA model was shown to produce a nice separation between the oat and groat classes.

An indirect experimental design was used to collect validation samples. The results showed that the variations in oat cultivar, moisture, seed size, and growing location present in the validation set could all be handled by a single classification model. The ability of the PLS-DA model to distinguish oats from groats was superior to that of the researcher; many 'misclassifications' were due to human error. Hyperspectral analysis of NIR images was therefore determined to be a feasible classification methodology which could be used to develop industrial machine vision equipment for oat milling. There is some future work required in post-processing the PLS-DA model predictions because the analysis has been conducted pixel-wise, but the required procedures are straightforward and are readily available in commercial software.

In conclusion, the latent variable methods of PCA, PLS, and PLS-DA were successful in achieving the goals of two very different applications in the food industry. The power of these methods is their ability to distill large sets of data down to a fewer number of dimensions that capture the directions of greatest variance. The resulting models are relatively simple compared to the original data, but are extremely powerful. In the case of product development, they allow a direct path towards a successful new product, which saves time and resources as compared to traditional trial-and-error methods. For classification applications in oat milling, the models offer a better classification than is achievable by the human eye.

References

Aikio, Mauri. *Hyperspectral Prism-Grating-Prism Imaging Spectrograph.* Espoo: Valtion teknillinen tutkimuskeskus (VTT), 2001.

Box, George E.P., J. Stuart Hunter, and William G. Hunter. *Statistics for Experimenters.* Hoboken: John Wiley & Sons, Inc., 2005.

Clark, Robert D., and Peter C. Fox. "Statistical Variation in Progressive Scrambling." *Journal of Computer-Aided Molecular Design* 18 (2004): 563-576.

Cornell, John A. *Experiments with Mixtures.* 3rd Edition. New York: John Wiley & Sons, Inc., 2002.

D'Agostini, Maurizio, interview by Emily Nichols. *Image Reconstruction Query* (May 30, 2011).

Fance, Wilfred James. *The Student's Technology of Breadmaking and Flour Confectionary.* London: Lowe & Brydone Printers Ltd., 1966.

Garcia-Muñoz, Salvador, Theodora Kourti, John F., Apruzzese, Francesca MacGregor, and Marc Champagne. "Optimization of Batch Operating Policies. Part I. Handling Multiple Solutions." *Industrial Engineering Chemistry Research* 45 (2006): 7856-7866.

Geladi, Paul, Jim Burger, and Torbjorn Lestander. "Hyperspectral Imaging: Calibration Problems and Solutions." *Chemometrics and Intelligent Laboratory Systems* 72 (2004): 209-217.

Grahn, Hans F., and Paul Geladi, . *Techniques and Applications of Hyperspectral Image Analysis.* Chichester: John Wiley & Sons Ltd., 2007.

Grassmann, Peter. *Physical Principles of Chemical Engineering.* New York: Pergamon Press, 1971.

Hall, M. B., A. W. Tarr, and M. Karopoulos. "Using Digital Imaging to Estimate Groat Per Cent and Milling Yield in Oats." *Journal of Cereal Science* 37 (2003): 343-348.

Hyvarinen, Timo, Esko Herrala, and Alberto Dall'Ava. "Direct Sight Imaging Spectrograph: A Unique Add-on Component Brings Spectral Imaging to Industrial Applications." *SPIE - Digital Solid State Cameras: Designs and Applications.* San Jose, 1998.

Jaeckle, Christiane, and John F. MacGregor. "Product Design through Multivariate Statistical Analysis of Process Data." *AICHE Journal* 44, no. 5 (1998): 1105-1118.

Joachim, David, Andrew Schloss, and Philip A. Handel. *The Science of Good Food.* Toronto: Robert Rose, Inc., 2008.

Kettaneh-Wold, Nouna. "Analysis of mixture data with partial least squares." *Chemometrics and Intelligent Laboratory Systems* (Elsevier Science Publishers B.V.) 14 (1992): 57-69.

Kourti, Theodora, and John F. MacGregor. "Multivariate SPC methods for process and product monitoring." *Journal of Quality Technology* 28, no. 4 (1996): 409-428.

Liu, J. Jay, Manish H. Bharati, Kevin G. Dunn, and John F. MacGregor. "Automatic masking in multivariate image analysis using support vector machines." *Chemometrics and Intelligent Laboratory Systems* 79, no. 1-2 (2005): 42-54.

Liu, Zheng. *NIR Imaging and its Application to Wheat Grading.* Hamilton: McMaster University, Masters Thesis, 2006.

Liu, Zheng, Honglu Yu, and John F. MacGregor. "Standardization of line-scan NIR imaging systems." *Journal of Chemometrics* 21 (2007): 88-95.

Martens, H., and M Martens. "Modified Jack-knife estimation of parameter uncertainty in bilinear modelling by partial least squares regression (PLSR)." *Food Quality and Preference* 11 (2000): 5-16.

McClure, W.Fred., and Satoru Tsuchikawa. "Instruments." In *Near Infrared Spectroscopy in Food Science and Technology*, by Yukihiro Ozaki, W. Fred McClure and Alfred A. Christy, 75-107. Hoboken: John Wiley & Sons, Inc., 2007.

Metzler, Ward, and James F. Egan. "Expanding the Frontiers of Test." *Evaluation Engineering.* August 2004. http://www.evaluationengineering.com/index.php/solutions/instrumentation/expanding-the-frontiers-of-test.html (accessed 05 10, 2011).

Muteki, Koji, and John F. MacGregor. "Sequential Design of Mixture Experiments for the Development of New Products." *Journal of Chemometrics* 21 (2007): 496-505.

Muteki, Koji, John F. MacGregor, and Toshihiro Ueda. "Estimation of Missing Data Using Latent Variable Methods with Auxiliary Information." *Chemometrics and Intelligent Laboratory Systems*, 2005: 41-50.

Muteki, Koji, John F. MacGregor, and Toshihiro Ueda. "Rapid Development of New Polymer Blends: The Optimal Selection of Materials and Blend Ratios." *Industrial & Engineering Chemistry Research* 45 (2006): 4653-4660.

Ozaki, Yukihiro, W. Fred McClure, and Alfred A. Christy, . *Near-Infrared Spectroscopy in Food Science and Technology.* Hoboken: John Wiley & Sons, Inc., 2007.

Redaelli, Rita, and Nicola Berardo. "Prediction of fibre components in oat hulls by near infrared reflectance spectroscopy." *Journal of the Science of Food and Agriculture* 87 (2007): 580-585.

Specim Spectral Imaging Ltd. *ImSpector Imaging Spectrograph User Manual Version 2.21.* User Manual, Oulu,Finland: Specim Spectral Imaging Ltd., 2003.

Specim Spectral Imaging Ltd. *Spectral Camera User Manual Ver.1.1.* User Manual, Oulu,Finland: Specim Spectral Imaging Ltd., 2001.

The Mathworks, Inc. *Example 2 - Analyzing Images - Matlab and Simulink Example.* 1994-2011. http://www.mathworks.com/help/toolbox/images/f0-8778.html (accessed 06 16, 2011).

—. *Examples and Webinars - Image Processing Toolbox for Matlab and Simulink.* 1994-2011. http://www.mathworks.com/products/image/demos.html (accessed 05 12, 2011).

Umetrics AB. *Multi- and Megavariate Data Analysis Part I: Basic Principles and Applications.* Umea: Umetrics AB, 2006.

van Raaij, Joop, Marieke Hendriksen, and Hans Verhagen. "Potential for Improvement of Population Diet Through Reformulation of Commonly Eaten Foods." *Public Health Nutrition* 12, no. 3 (2008): 325-330.

Wilks, Paul. "NIR Versus Mid-IR: How To Choose." *Spectroscopy Online.* April 1, 2006. http://spectroscopyonline.findanalytichem.com/spectroscopy/Near-IR+Spectroscopy/NIR-Versus-Mid-IR-How-to-Choose/ArticleStandard/Article/detail/318524 (accessed May 20, 2011).

Wold, Svante. "Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models." *Technometrics* 20, no. 4 (1978): 397-405.

Wold, Svante, Mats Josefson, Johan Gottfries, and Anna Linusson. "The utility of multivariate design in PLS modeling." *Journal of Chemometrics* 18 (2004): 156-165.

Wold, Svante, Michael Sjostrom, and Lennart Eriksson. "PLS-regression: a basic tool of chemometrics." *Chemometrics and Intelligent Laboratory Systems* 58 (2001): 109-130.