RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS

Joe Luis Villa Medina

Dipòsit Legal: T.1521-2013

RELIABILITY OF CLASSIFICATION AND PREDICTION IN *k*-NEAREST NEIGHBOURS

Joe Luis Villa Medina

DOCTORAL THESIS

UNIVERSITAT ROVIRA I VIRGILI

UNIVERSITAT ROVIRA I VIRGILI
RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS
Joe Luis Villa Medina
Dipòsit Legal: T.1521-2013

Joe Luis Villa Medina

RELIABILITY OF CLASSIFICATION AND PREDICTION IN *k*-NEAREST NEIGHBOURS

DOCTORAL THESIS

Supervisors
Dr. Ricard Boqué Martí
Dr. Joan Ferré Baldrich

Department of Analytical Chemistry and Organic Chemistry

UNIVERSITAT ROVIRA I VIRGILI
Tarragona
2013

UNIVERSITAT ROVIRA I VIRGILI
RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS
Joe Luis Villa Medina
Dipòsit Legal: T.1521-2013

UNIVERSITAT ROVIRA I VIRGILI
Department of Analytical Chemistry
and Organic Chemistry

Dr. RICARD BOQUÉ MARTÍ and **Dr. JOAN FERRÉ BALDRICH**, associate professors of the Department of Analytical Chemistry and Organic Chemistry at Rovira I Virgili University.

CERTIFY:

The Doctoral Thesis entitled: **RELIABILITY OF CLASSIFICATION AND PREDICTION IN *k*-NEAREST NEIGHBOURS**, presented by **JOE LUIS VILLA MEDINA** to receive the degree of Doctor of the Rovira I Virgili University, has been carried out under our supervision, in the Department of Analytical Chemistry and Organic Chemistry at Rovira I Virgili University, and all the results presented in this thesis were obtained in experiments conducted by the above mentioned student.

Tarragona, September 2013

Dr. Ricard Boqué Martí **Dr. Joan Ferré Baldrich**

UNIVERSITAT ROVIRA I VIRGILI
RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS
Joe Luis Villa Medina
Dipòsit Legal: T.1521-2013

Dipòsit Legal: T.1521-2013

I am very grateful, in general, for all those who in one way or another have helped me with personal and professional development and in particular all those who encouraged me and helped me throughout my doctoral studies.

This thesis was supervised by Dr. Ricard Boqué and Dr. Joan Ferre. The work was undertaken in the Chemometrics and Qualimetrics research group, headed by Prof. F. Xavier Rius, of the Analytical Chemistry and Organic Chemistry Department at the Rovira i Virgili University in Tarragona, Spain. And funded by the 'Agència de Gestió d'Ajusts Universitaris i Recerca' of the Catalan Government. To them my most sincere thanks for all the support and all the help they gave me during my thesis.

UNIVERSITAT ROVIRA I VIRGILI
RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS
Joe Luis Villa Medina
Dipòsit Legal: T.1521-2013

UNIVERSITAT ROVIRA I VIRGILI
RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS
Joe Luis Villa Medina
Dipòsit Legal: T.1521-2013

Dedicado a mis padres, porque sin su apoyo, dedicación y esfuerzo todo esto no hubiese sido posible.

UNIVERSITAT ROVIRA I VIRGILI
RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS
Joe Luis Villa Medina
Dipòsit Legal: T.1521-2013

UNIVERSITAT ROVIRA I VIRGILI
RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS
Joe Luis Villa Medina
Dipòsit Legal: T.1521-2013

En memoria de mi abuelita, Alcira, y de mi tío, Chago, a quienes perdí en la distancia, y cuyo duelo, a la vez que me hacía vivir los momentos más difíciles de mi vida, me animaba a no fracasar en el intento……..

UNIVERSITAT ROVIRA I VIRGILI
RELIABILITY OF CLASSIFICATION AND PREDICTION IN K-NEAREST NEIGHBOURS
Joe Luis Villa Medina
Dipòsit Legal: T.1521-2013

Contents

CONTENTS 13

1. INTRODUCTION AND OBJECTIVES 19

1.1 INTRODUCTION 19
1.2 OBJECTIVES 21
1.3 NOTATION 23
1.4 ABBREVIATIONS 24
1.5 STRUCTURE OF THE THESIS 24
1.6 REFERENCES 27

2. STATE OF THE ART OF *K*NN AND BOOTSTRAP 33

2.1 PATTERN RECOGNITION 33
2.1.1 DATA COLLECTION 33
2.1.2 PREPROCESSING 34
2.1.3 CALCULATION OF THE CLASSIFIER 35
2.1.4 VALIDATION 36
2.1.5 OPTIMIZATION 38
2.1.6 CLASSIFICATION (PREDICTION) 38
2.2 *K*NN IN CLASSIFICATION 39
2.2.1 CHEMICAL AND RELATED APPLICATIONS 41
2.2.2 ADVANTAGES AND DISADVANTAGES 41
2.2.3 VARIATIONS OF *K*NN 42
2.2.4 PROBABILISTIC APPROACH 45
2.3 OTHER CLASSIFICATION METHODS 47

2.3.1 BAYES' DECISION RULE 47
2.3.2 LINEAR DISCRIMINANT ANALYSIS 48
2.4 MULTIVARIATE CALIBRATION 49
2.4.1 DATA COLLECTION 50
2.4.2 PREPROCESSING 50
2.4.3 CALCULATION OF THE MODEL 51
2.4.4 OPTIMIZATION OF THE MODEL. 51
2.4.5 VALIDATING THE MODEL. 52
2.4.6 PREDICTION 52
2.4.7 PREDICTION UNCERTAINTY 53
2.4.8 OUTLIER DETECTION 54
2.5 *K*NN FOR PREDICTION 55
2.5.1 CHEMICAL AND RELATED APPLICATIONS 56
2.5.2 ADVANTAGES AND DISADVANTAGES OF *K*NN IN PREDICTION 56
2.6 BOOTSTRAP 57
2.6.1 NOTATION AND BOOTSTRAP GENERALITIES 57
2.6.2 BOOTSTRAP APPLICATIONS 61
2.7 RELIABILITY 72
2.7.1 RELIABILITY OF CLASSIFICATION 73
2.7.2 RELIABILITY OF PREDICTION 73
2.8 REFERENCES 75

3. RELIABILITY OF *K*-NEAREST NEIGHBOURS IN CLASSIFICATION 92

3.1 INTRODUCTION 92
3.2 PAPER. CALCULATION OF THE PROBABILITY OF CORRECT CLASSIFICATION IN PROBABILISTIC BAGGED *K*-NEAREST NEIGHBOURS. 94

3.2.1 INTRODUCTION 96
3.2.2 METHODS 99
3.2.3 EXPERIMENTAL SECTION 104
3.2.4 RESULTS AND DISCUSSION 107
3.2.5 CONCLUSIONS 125
3.2.6 REFERENCES 126

4. INFLUENCE OF THE MEASUREMENT ERROR ON THE RELIABILITY OF CLASSIFICATION WITH *K*NN 132

4.1 INTRODUCTION 132

4.2 PAPER. BAGGED *K*-NEAREST NEIGHBOURS WITH UNCERTAINTY IN THE VARIABLES. 133

4.2.1 INTRODUCTION 135
4.2.2 METHODS 137
4.2.3 EXPERIMENTAL SECTION 142
4.2.4 RESULTS AND DISCUSSION 145
4.2.5 CONCLUSIONS 156
4.2.6 REFERENCES 159

5. MULTIVARIATE CALIBRATION WITH *K*-NEAREST NEIGHBOURS 164

5.1 INTRODUCTION 164

5.2 PAPER. MULTIVARIATE CALIBRATION WITH *K*-NEAREST NEIGHBOURS. 167

5.2.1 INTRODUCTION 169
5.2.2 METHODS 170
5.2.3 EXPERIMENTAL SECTION 173

5.2.4 RESULTS AND DISCUSSION 175
5.2.5 PHARMACEUTICAL DATASET 182
5.2.6 CONCLUSIONS 187
5.2.7 REFERENCES 189

6. UNCERTAINTY OF PREDICTIONS WITH *K*-NEAREST NEIGHBOURS 194

6.1 INTRODUCTION 194
6.2 PAPER. UNCERTAINTY OF PREDICTIONS WITH *K*-NEAREST NEIGHBOURS 196
6.2.1 INTRODUCTION 198
6.2.2 METHODS 200
6.2.3 RESULTS AND DISCUSSION 207
6.2.4 CONCLUSIONS 210
6.2.5 REFERENCES 211

7. CONCLUSIONS 218

7.1 INTRODUCTION 218
7.2 ABOUT THE RELIABILITY OF CLASSIFICATION USING *K*NN 219
7.3 ABOUT THE RELIABILITY OF PREDICTION USING *K*NN 220
7.4 FUTURE WORK 221

CHAPTER 1

INTRODUCTION AND OBJECTIVES

1. Introduction and objectives

1.1 Introduction

Data analysis using multivariate statistical methods is becoming a routine step in many analytical processes. Multivariate classification [1-9] and multivariate calibration [10-25] allow the analytical scientist to predict a class label or a property value of an unknown object using multiple instrumental responses (e.g., a near-infrared spectrum) or values of physical and chemical properties of that object.

The result of a prediction process (a class label, as in classification, or a property value, as in multivariate calibration), should include its uncertainty or its degree of reliability [26-31]. When a class label is assigned to an unknown object, the probability that the assigned label represents the true class should be indicated. This probability will depend, among others, on the ability of the classification model for discriminating or modelling the given class, and also on the similarity between the measurements made on that object and the measurements made on training objects of that class. Noise, uninformative variables, systematic data variations (e.g., baseline variation) will also influence the uncertainty of the assigned class label. Similarly, a value of property predicted using a multivariate calibration model should be accompanied with a value indicating the range within which the true value of the property value lies with a given probability.

Despite the great variety of applications that use multivariate classification and multivariate calibration [8, 32], the uncertainty of the classification/calibration results is not always reported, and its calculation is often controversial [33] and never simple, because of the complex mathematics involved in the prediction process. Furthermore, uncertainty is very sensitive to certain conditions in the data (i.e. normality, homoscedasticity…), especially when these are not fulfilled, an aspect which is not always checked. Because of these difficulties, it is common to report average uncertainty values obtained from a validation set. These values, calculated as a measure of average performance of the model, are usually attached to any new unknown object that must be predicted, independently on the intrinsic characteristics of that object. In other words, the prediction of two different unknown objects, located at different positions in the multivariate space, are both given the same "average" uncertainty. In order to improve the quality of this assessment, one trend in Analytical Chemistry is to report individual uncertainty values for each object being analyzed.

The words ""uncertainty" and "reliability" are often used in the literature. See P. De Bievre "Uncertainty or Reliability" [34] J. D. R. Thomas "Reliability versus Uncertainty for Analytical Measurement" [35]. In all cases the uncertainty is defined as a parameter, associated with the result of a measurement, which characterizes the dispersion of the values that could reasonably be attributed to the measurand [36]. Reliability is considered as a quantitative indication of the quality of a result [35, 37]. According the Comprehensive Chemometrics: Chemical and Biochemical Data Analysis, section 1.02 [38], *the result of a measurement has to be accompanied by an evaluation of*

(un)certainty or its degree of reliability. This is done by means of a confidence interval. For this reason, this thesis focuses in the development of alternatives to obtain particular reliabilities for each unknown object.

During the development of this thesis several classification and prediction methods and different reliability estimation approaches were studied. From the different methods available, we chose the *k* nearest neighbours (*k*NN) method, a non parametric method and one of the most intuitive for classification and prediction purposes, but for which no reliability estimation approaches were found.

1.2 Objectives

The aim of this thesis is to develop new chemometric methods of classification and calibration, based on *k*NN, which can provide the uncertainty or the degree of reliability of the classification and prediction, respectively. To achieve this general objective the following specific objectives are proposed:

1. To study and discuss in detail the *k*NN method for classification and prediction and to evaluate its advantages and disadvantages.
2. To discuss the different approaches for reliability estimation in classification and prediction and to evaluate which of them can be applied to *k*NN.

3. To develop a classification method based on *k*NN that can improve the classification results of the classical *k*NN method and, at the same time, provide an estimation of the reliability of classification.
4. To develop a calibration method based on *k*NN that provides the reliability of prediction.

The use of *k*NN for classification was studied first. Two variants of classification with *k*NN, which provide the reliability of the classification for a specific object, were investigated. In the first method, the reliability of classification was computed using only the information of the objects, that is, the values of the multiple instrumental variables measured. In the second method, the uncertainty of the instrumental variables was also taken into account in the training step of the classifier and in the classification of the unknown objects. The methods mentioned above combined *k*NN with the resampling *bootstrap* method. Bootstrap has been shown to improve classification results [39] and has already been used to estimate the uncertainty of other prediction methods [35, 40]

In the second part of this thesis, a variation of *k*NN for predicting continuous variables was developed. The procedure use direct orthogonalization (DO) to improve the prediction ability of *k*NN. DO is used to remove irrelevant variability in the independent variables and to improve the identification of the *k* neighbours that will be used for prediction. Finally, for this method, bootstrap confidence intervals were computed to obtain the object specific uncertainty of prediction.

1.3 Notation

In this thesis the following notation has been used:

I	number objects: $i = 1,2,\ldots,I$
J	number of variables: $j=1,2,\ldots,J$
$\mathbf{X,Y}$	matrix, bold capital letter: $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_I\}$
$\mathbf{x,y}$	vector, bold lowercase letter: $\mathbf{x}=\{x_1,x_2,\ldots,x_j\}$
x, y	scalar, italic lowercase letter
c	class, $c=\{1,2,\ldots,C\}$
I_c	number of objects belonging to class c
$\mathbf{X}_c$	matrix of objects of class c
$\mathbf{y}_c$	vector of assigned classes for the objects of class c
$\mathbf{x}_t$	unknown object or test object
k	number of nearest neighbours used by kNN
r	number of nearest neighbours used in Hamamoto's bootstrap
I_{cal}	number of objects in the calibration set
I_{val}	number of objects in the validation set
(^)	hat symbol used to indicate predicted value
σ, σ^2	standard deviation and variance
h	leverage

1.4 Abbreviations

In this thesis the following abbreviations have been used:

BCa	Bias Corrected and accelerated
DO	Direct Orthogonalization
*k*NN	*k* nearest neighbours
LDA	Linear Discriminant Analysis
LOOCV	Leave-one-out cross-validation
PCA	Principal Component Analysis
PLS	Partial Least Squares
SNV	Standard Normal Variate

1.5 Structure of the thesis

This thesis has been structured in 6 chapters:

Chapter 1, *Introduction and objectives,* contains the introduction, objectives, notation and structure of the thesis.

Chapter 2, *State of the art of k*NN *in chemometrics,* has six parts. In the first part, pattern recognition is briefly introduced, and the relationship with classification methods and the steps used to carry out a classification are explained. The second part explains the *k*NN method in classification along with its advantages and limitations, and the relationship with other classification methods. The third part describes multivariate calibration and the steps needed until a prediction is

obtained, including the calculation of confidence intervals and the identification of outliers. The fourth part describes the use of *k*NN method for prediction, its advantages and limitations and its chemical applications. In the fifth part, the bootstrap method is described, along with its variations and its application to obtain confidence intervals. In the last part, the concept of reliability, both in classification and prediction, is discussed.

Chapter 3, *Reliability of classification with k*NN, describes a new method to compute the reliability of classification for *k*NN using the bootstrap method. This chapter corresponds to the published paper: *Calculation of the probability of correct classification in probabilistic bagged k-nearest neighbours*, published in Chemometrics and Intelligent Laboratory Systems, Vol 94 No 1 (2008) 51-59.

Chapter 4, *Influence of the uncertainty in the classification reliability of k*NN, describes a new classification method based on *k*NN that takes into account the uncertainty in the **X**-values to classify an unknown object. This chapter corresponds to the published paper: *Bagged k-nearest neighbours with uncertainty in the variables*, published in Analytica Chimica Acta. 646 (2009) 62-68

Chapter 5, *Multivariate calibration with k-Nearest Neighbours,* introduces Direct Orthogonalization *k*-Nearest Neighbours (DO*k*NN), which uses *k*NN for predicting continuous properties.

Chapter 6, *Uncertainty of predictions with k-Nearest Neighbours,* propose a new bootstrap-based method to compute the uncertainty of

predictions of the Direct Orthogonalization *k*NN (DO*k*NN) method presented in chapter 5.

Chapter 7: *Conclusions*. The characteristics of the proposed methods, their advantages and limitations, their applicability and, also, the future work related to this research, are summarized and discussed.

1.6 References

1. Marini F (2010) Classification methods in chemometrics. Current Analytical Chemistry 6:72

2. Ni L, Zhang L, Xie J, Luo J (2009) Pattern recognition of chinese flue-cured tobaccos by an improved and simplified *k*-nearest neighbors classification algorithm on near infrared spectra. Analytica Chimica Acta 633:43

3. Villegas JI, Kubička D, Reinikainen S, Addová G, Kubinec R, Salmi T, Murzin DY (2007) Classification and pattern recognition of acyclic octenes based on mass spectra. Talanta 72:1573

4. Lukasiak BM, Faria R, Zomer S, Brereton RG, Duncan JC (2006) Pattern recognition for the analysis of polymeric materials. Analyst 131:73

5. Alonso-Salces RM, Herrero C, Barranco A, Berrueta LA, Gallo B, Vicente F (2005) Classification of apple fruits according to their maturity state by the pattern recognition analysis of their polyphenolic compositions. Food Chemistry 93:113

6. Pierce KM, Hope JL, Johnson KJ, Wright BW, Synovec RE (2005) Classification of gasoline data obtained by gas chromatography using a piecewise alignment algorithm combined with feature selection and principal component analysis. Journal of Chromatography A 1096:101

7. Kiang MY (2003) A comparative assessment of classification methods. Decision Support Systems 35:441

8. Duda RO, Hart PE, Stork DG (2001) Pattern classification. John Wiley & Sons, New York

9. Kowalski BR, Bender CF (1972) Pattern recognition. Powerful approach to interpreting chemical data. Journal of the American Chemical Society 94:5632

10. Tarumi (2009) Multivariate Calibration with basis functions derived from optical filters. Analytical Chemistry 81:2199

11. Xiang (2009) Robust Calibration design in the pharmaceutical quantitative measurements with near-infrared (NIR) spectroscopy:

avoiding the chemometric Pitfalls. Journal of Pharmaceutical Sciences 98:1155

12. Bona MT, Andres JM (2008) Application of chemometric tools for coal classification and multivariate calibration by transmission and drift mid-infrared spectroscopy. Analytica Chimica Acta 624:68

13. Gomez V, Callao MP (2008) Analytical applications of second-order calibration methods. Analytica Chimica Acta 627:169

14. Ribeiro MPA, Pádua TF, Leite OD, Giordano RLC, Giordano RC (2008) Multivariate calibration methods applied to the monitoring of the enzymatic synthesis of amipicilin. Chemometrics and Intelligent Laboratory Systems 90:169

15. Sulub Y, LoBrutto R, Vivilecchia R, Wabuyele B (2008) Near-infrared multivariate calibration updating using placebo: A content uniformity determination of pharmaceutical tablets. Vibrational Spectroscopy 46:128

16. Forina M, Lanteri S, Casale M (2007) Multivariate calibration. Journal of Chromatography A 1158:61

17. Escandar GM, Damiani PC, Goicoechea HC, Olivieri AC (2006) A review of multivariate calibration methods applied to biomedical analysis. Microchemical Journal 82:29

18. Gabrielsson J, Trygg J (2006) Recent developments in multivariate calibration. Critical Reviews in Analytical Chemistry 36:243

19. Kalivas JH, Gemperline P (2006) Sampling theory, distribution functions and the multivariate normal distribution. In: Practical Guide to Chemometrics, Second Edition, editor P.J. Gemperline.CRC Press Taylor & Francis Group, Boca Raton, Florida.

20. Hanrahan G, Udeh F, Patil DG (2005) In: Paul Worsfold, Alan Townshend, Colin Poole (eds) Encyclopedia of Analytical Science, Elsevier, Oxford

21. Benoudjit N, Cools E, Meurens M, Verleysen M (2004) Chemometric calibration of infrared spectrometers: selection and validation of variables by non-linear models. Chemometrics and Intelligent Laboratory Systems 70:47

22. Bro R (2003) Multivariate calibration: What is in chemometrics for the analytical chemist? Analytica Chimica Acta 500:185

23. Brereton RG (2000) Introduction to multivariate calibration in analytical chemistry. Analyst 125:2125

24. Forina M, Casolino MC, de la Pezuela Martinez C (1998) Multivariate calibration: applications to pharmaceutical analysis. Journal of Pharmaceutical and Biomedical Analysis 18:21

25. Martens H, Naes T (1989) Multivariate calibration. Wiley, Chichester.

26. Ellison SLR, Gregory S (1998) Perspective Quantifying uncertainty in qualitative analysis. Analyst 123:1155

27. EURACHEM - CITAC (2000) Quantifying uncertainty in analytical measurement. CITAC Guide Number 4:126

28. Zhang L, Garcia-Munoz S (2009) A comparison of different methods to estimate prediction uncertainty using Partial Least Squares (PLS): A practitioner's perspective. Chemometrics and Intelligent Laboratory Systems 97:152

29. Meyer VR (2007) Measurement uncertainty. Journal of Chromatography A 1158:15

30. Olivieri AC, Faber N.M, Ferré J, Boqué R, Kalivas JH, Mark H (2006) Uncertainty estimation and figures of merit for multivariate calibration (IUPAC Technical Report). Pure and Applied Chemistry 78:633

31. Ortiz MC, Sarabia LA, Sánchez MS, Herrero A (2009) In: Editors-in-Chief: Stephen D. Brown, Romà Tauler, Beata Walczak (eds) Comprehensive Chemometrics, Elsevier, Oxford

32. Wold S, Sjöström M, Eriksson L (2001) PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems 58:109

33. Faber NM, Song XH, Hopke PK (2003) Sample-specific standard error of prediction for partial least squares regression. Trends in Analytical Chemistry 22:330

34. De Bièvre P (1996) "Uncertainty" or "reliability"?. Accreditation and Quality Assurance: Journal for Quality, Comparability and Reliability in Chemical Measurement 1:195

35. Thomas JD (1996) Reliability versus uncertainty for analytical measurement. Analyst 121:1519

36. ISO/IEC 17025:2005: General requirements for the competence of testing and calibration laboratories

37. Desimoni E, Brunetti B (2011) Uncertainty of measurement and conformity assessment: a review. Analytical and Bioanalytical Chemistry 400:1729

38. Ortiz MC, Sarabia LA, Sánchez MS, Herrero A (2009) In: Editors-in-Chief: Stephen D. Brown, Romà Tauler, Beata Walczak (eds) Comprehensive Chemometrics, Elsevier, Oxford

39. Breiman L (1996) Bagging predictors. Machine Learning 24:123

40. Carpenter J, Bithell J (2000) Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. Statistics in Medicine 19:1141

CHAPTER 2

STATE OF THE ART OF *k*NN AND BOOTSTRAP

2. State of the art of *k*NN and Bootstrap

2.1 Pattern recognition

Pattern recognition can be defined as *"the act of taking in raw data and taking an action based on the "category" of the pattern"* [1]. An example of supervised pattern recognition would be to classify one wine sample (object) of unknown origin into one of several predefined origins (classes) based on its physicochemical properties [2]. A large variety of supervised pattern recognition has been applied in Chemistry [3].

In general, supervised pattern recognition involves: data collection, preprocessing, calculation of the classifier (or classification rule), optimization, validation and prediction [1, 4-6].

2.1.1 Data Collection

Supervised pattern recognition starts with collecting a representative set of objects of known class along with the measurement of characteristic variables (chemical, physical, sensory,...) on those objects. An object is described by J variables $\mathbf{x}=[x_1, x_j,..., x_J]$, with $j=1,...,J$. For a given class c ($c=1,2,...,C$) of I_c objects, $\mathbf{X}_c$ represents the matrix of measurements and $\mathbf{y}_c$ represents the column vector of class labels. The x-data are grouped in a matrix $\mathbf{X}$ $(I \times J)=[\mathbf{X}_1, \mathbf{X}_2,..., \mathbf{X}_C]$ and the y-data in a column vector of the classes $\mathbf{y}$ $(I \times 1)=[\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_C]$, with $I = \sum_{c=1}^{C} I_c$. This data scheme is represented in Figure 2.1.

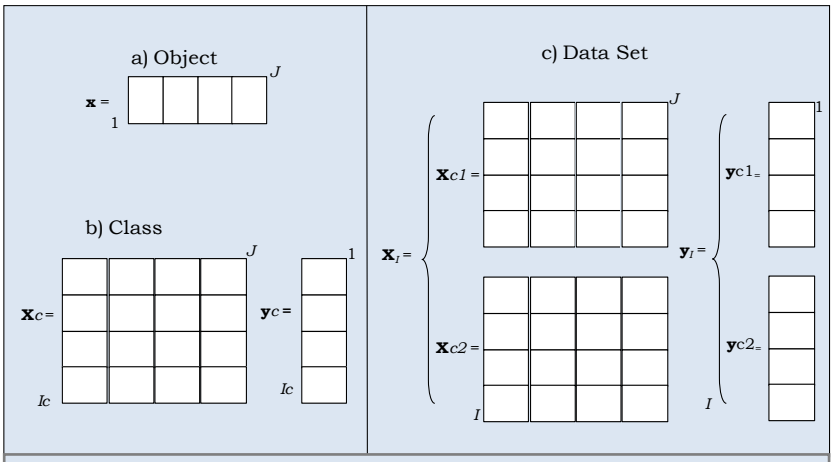

Figure 2.1 Representation of: a) object; b) objects of one class and c) data set with two classes

2.1.2 Preprocessing

Preprocessing is a set of mathematical transformations performed on the data with the aim of changing the representation of the information contained in them [7]. It is used to scale the data appropriately and to remove data variability that is not related with the property (class) to be predicted. Examples of preprocessing methods are: mean centering, scaling, standard normal variate (SNV), detrending and derivatives, among others [8].

Preprocessing can include variable selection and variable reduction. The aim of selecting variables (e.g. specific wavelengths in a spectrum) is to keep those that are the most characteristic of the class being modelled, and remove those that contain unrelated variation or are very noisy [6]. Variable reduction is used to determine a subspace of lower dimensionality where the data lies. An example of a variable reduction method is Principal Component Analysis, PCA [9].

2.1.3 Calculation of the classifier

A classifier is a mathematical model calculated from a set of training objects that relates a (discrete or continuous) variable space to a discrete set of classes. Some classifiers focus on discrimination whiles other focus on modelling the classes. Discriminant methods find a set of optimal boundaries between classes so that the object is classified according to its position with respect to the boundary. Some methods calculate explicitly the boundaries between classes (e.g. Linear Discriminant Analysis, LDA) while in others these boundaries are implicit (e.g. *k*-nearest neighbours, *k*NN). Class modelling methods (e.g. SIMCA) build a model for each class, and an unknown object is assigned either in a class, in several classes or in none of the classes, depending if the unknown object fits or not the class models. This thesis will focus on the first type of methods, concretely *k*NN.

Classification methods can also be parametric or non-parametric. Parametric methods use the parameters of the statistical distribution of the objects in the development of the model or rule of classification. LDA, which is based on the multivariate normal distribution, is among the parametric methods. The main shortcoming of parametric methods is that their application is subjected to the fulfilment of statistical requirements. Non-parametric methods (e.g. *k*NN), on the other hand, do not use such parameters and do not need to comply with these requirements. Their disadvantage compared with parametric methods is that obtaining the probabilities of correct classification is not straightforward [4, 6].

2.1.4 Validation

Once the classifier has been calculated it is necessary to measure its ability to correctly classify future unknown objects. Measures of future classification performance [10] can be computed with the help of a validation set, by cross-validation or by resampling methods.

If the number of objects is large, the dataset **X**, can be divided into a training set, with I_{cal} objects, and a validation set with I_{val} objects. The training set is used to build the classifier and the validation set is used to validate it. The classification performance is then computed by comparing the predicted classes and the known class of the objects in the validation set. Different approaches can be used for dividing **X** into training and validation sets. Two popular methods are randomly or with algorithms as the Kennard and Stone's algorithm [11], which selects objects that are uniformly distributed over the variable space [12]. *m-fold* cross-validation uses *I-m* objects in the training set and *m* in the validation set. [13]. This procedure is repeated until all training objects have been left-out and predicted once and the *I* classification results are used to compute the classification performance measure. When *m*=1, the procedure is called *leave-one-out* cross-validation (LOOCV).

Resampling methods, such as jackknife and bootstrap, have also been used to compute the classification performance [13]. In jackknife a new dataset, known as jackknife replication, is obtained by removing one object of the original dataset. The jackknife replication, with *I*-1 objects, is used to calculate the parameters of classification performance. This procedure is repeated until all objects have been

removed once and I measures of classification performance are obtained. Finally, the classification performance, for the evaluated dataset, is calculated as the mean of the I parameters obtained for each jackknife replication [14]. Bootstrap works by generating B new datasets from the original dataset by resampling with or without replacement [15], and then each new dataset is used to compute the bootstrap parameters of classification performance. The final parameters of classification performance, for the evaluated dataset are obtained by averaging the B bootstrap parameters obtained. Bootstrap is described in details in section 2.6.

The results of classification are presented in a $C \times C$ confusion matrix where C is the number of classes [10]. Table 2.1 shows an example of a confusion matrix for three classes. The diagonal contains the number of objects correctly classified in each class; the off-diagonal cells contain the number of objects that have been assigned to a class different than the true class.

Table 2.1 Confusion matrix for a dataset with three classes and I evaluated objects.

Real Class	**Assigned Class**			
	1	**2**	**3**	
1	$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	$I_{c,1}$
2	$I_{2,1}$	$I_{2,2}$	$I_{2,3}$	$I_{c,2}$
3	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$	$I_{c,3}$
	$I_{\hat{c},1}$	$I_{\hat{c},2}$	$I_{\hat{c},3}$	

I_c and $I_{\hat{c}}$ are the number of objects in the true and in the assigned class respectively. Every entry $I_{c,\hat{c}}$ is the number of objects that belong to the class c and have been assigned to class $\hat{c}$. The diagonal of the table contains the number of objects that have been correctly classified

The most commonly used measure of performance of a classifier is the Classification Error Rate (CER), which is the percentage of the wrongly assigned objects. It is given by [10]:

$$CER = \frac{I_{val} - \sum_{c=1}^{C} I_{c,\hat{c}}}{I_v} \times 100 \qquad Eq.2.1$$

where I_{val} is the number of classified objects (it can be replaced by I if cross-validation is used) and $I_{c,\hat{c}}$ is the number of objects of class c that have been correctly classified.

2.1.5 Optimization

A preliminary calculated classifier offers insight about the data structure and the contributions of the different objects and variables to the classification process. The exploration of these preliminary results can point to the presence of outliers (either objects, variables or both) that could be removed, or suggest new preprocessing schemes that could be applied in order to improve the classification performance. This involves recalculating and validating again the classifier.

2.1.6 Classification (Prediction)

Once the classifier has been optimized and validated it can be used to classify unknown objects. The variables for new objects are measured, preprocessed following the same scheme as used for the training data, and submitted to the classification rule. A class label is then assigned to every unknown object. In addition to the class label, the classification result should also contain a measure of the reliability of

that assignment, in the same way as quantitative results are requested to be reported with a measure of its uncertainty. The reliability of classification will be discussed in section 2.7.

2.2 *k*NN in classification

The classical *k*-nearest neighbours (*k*NN) method is a nonparametric method that assigns an unknown object, $\mathbf{x}_t$, into the class where the majority of its *k* nearest neighbours belong. Figure 2.2 illustrates the classification process in *k*NN. The circle encloses the three objects considered for *k*=3; the unknown object $\mathbf{x}_t$ is classified in the class 1 because two of the three neighbours belong to class 1.

The *k*NN algorithm works as follows:

1) Compute the distance between $\mathbf{x}_t$ and all objects in **X**, where **X** is the $I \times J$ matrix of training objects, with I_c objects belonging to c ($c = 1,\ldots,C$) possible classes and J variables have been measured.

2) Sort the objects according to the distances in ascending order and count how many class labels of each class are among the first k sorted objects.

3) Classify $\mathbf{x}_t$ in the class to which the majority of its k nearest neighbours belong.

The Euclidean distance is the most usual metric [16-19]. It is calculated as:

$$d_E(\mathbf{x}_t, \mathbf{x}_i) = \sqrt{\sum_{j=1}^{J} (\mathbf{x}_{tj} - \mathbf{x}_{ij})^2} \qquad Eq.\,2.2$$

where $d_E(\mathbf{x}_t, \mathbf{x}_i)$ is the Euclidean distance between the unknown object,$\mathbf{x}_t$, and the training object, $\mathbf{x}_i$. Other metrics, such as the Manhattan distance, the cosine coefficient or the Lagrange distance have been used [1, 19].

There are several approaches for finding the appropriate k [20, 21]. The most common is to test several values of k by cross-validation [1] and keep the k giving the lowest classification error rate.

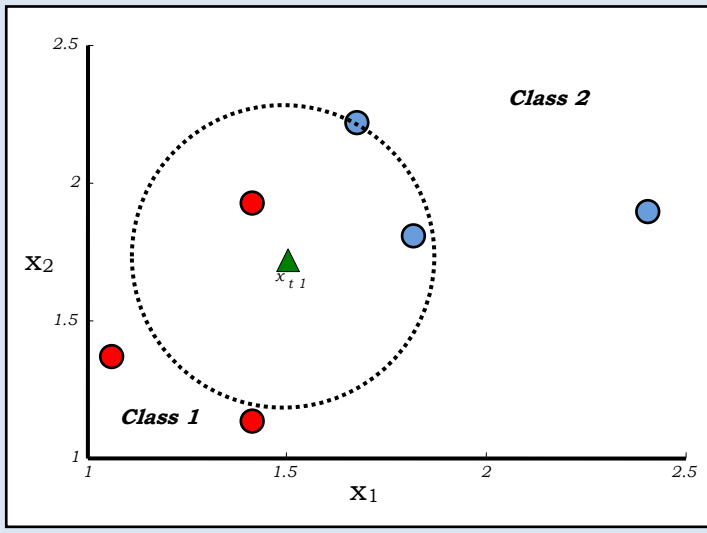

Figure 2.2 kNN classification method with k=3.

2.2.1 Chemical and related applications

*k*NN has been applied in Chemistry since the early 1970s. For instance, 1-NN was used to classify molecular structures using its nuclear magnetic resonance (NMR) spectra [22]. *k*NN was also used to classify the elements of the periodic table with respect to their representative oxide [7] and to classify hydrocarbons within three different classes using their mass spectra [23]. *k*NN has also been used to classify obsidian source samples into four classes according to their origin [24]; honeys according to their type [25]; sensor array data for two types of chemical warfare agents [26]; samples within a group of extraction kinetics of fat in bakery products using mid infrared spectra [27]; apples according to their maturity [28]; polymer materials into four polymer classes [29]; whiskeys into three classes of commercial whiskeys [30]; soil samples according to their geographic origin [31]; samples of green tea within four green tea grade levels [32] and for discrimination of the geographical origin of Codonopsis pilosula [33]. In short, the applications of *k*NN in chemistry are varied and offer, in general, good results of classification.

2.2.2 Advantages and disadvantages

The main advantage of *k*NN for classification is that it is non parametric [34], i.e. the distribution of the data does not need to be known. Also, it is conceptually and computationally quite simple because it is based on distances, it is multi class, it does not assume a linear separability of the data [7, 34-35], it is very stable (i.e. small changes in the training data do not lead to significantly different classification results [36]), it can learn from a small set of objects and it

can incrementally add new information and give competitive performance [37]. The limitations of *k*NN are that it does not perform well if the classes are unbalanced, i.e. if number of the objects in the training classes is very different from one class to another, because it increases the probability of finding nearest neighbours belonging to the class with the largest number of objects. Also, it is sensitive to the *k* value [38], which must be optimized. Although several probabilistic approaches are known for *k*NN, they are not used to provide the reliability of the classification for a particular object [39]. One reason is that probabilistic approaches only work well when the number of training objects is very large [40], which is not always common in chemical applications. Another important limitation of the *k*NN method is *the course of dimensionality*, which suggests the *peaking phenomenon,* i.e. for a constant number of objects, the peak of classification accuracy decreases when the number of variables increases [41-43]. This can be avoided by using a large number of objects or by reducing the dimensionality of the data [44-46].

2.2.3 Variations of *k*NN

Several variations of the *k*NN method have been proposed with the aim of improving its classification performance.

2.2.3.1 *Changes in the metric used to find the neighbours*

The metric affects the results of *k*NN. When different metrics were tested (Euclidean, Manhattan, Cosine coefficient, Camberra, Lance-Williams and Lagrange), i.e. the Lance-Williams, Manhattan and Camberra gave comparable classification error rates and, in some

cases, *k*NN gave better results than LDA [19]. However, Lance-Williams' and Camberra's metrics are not applicable to data with negative values, which are often found in chemical data (e.g. when autoscaling or derivatives have been applied).

2.2.3.2 *Variable reduction*

The aim of reducing the dimensionality of a data matrix is to remove the uninformative variables that can affect negatively the classification results [47]. In this sense, several methods have been applied before classifying with *k*NN [46]. For example local PCA (for each individual class) or global PCA (for entire training dataset) have been used before the classification with *k*NN [48, 49]. The Multi-label dimensionality reduction method (MDDM) has also been used. MDDM attempts to project the original data into a lower-dimensional feature space maximizing the dependence between the original feature description and the associated class labels [50].

2.2.3.3 *Reduction of the number of objects*

The aim of reducing the number of objects is to condense the number of the objects used in the training set to reduce the storage and computing requirements needed by *k*NN and to improve the results of classification. Hart [51] proposed the *condensed nearest neighbours rule* (CNN). In CNN, a consistent subset is obtained from the collected dataset. A consistent subset is a training set which classifies correctly the objects in the test set [51]. Variations of this method have been proposed [52, 53]. Kuncheva [54] used genetic algorithms for selecting the objects in the dataset. Other strategies to reduce the number of

objects and some applications of them have been described by Desarathy *et al* [55], Sanchez *et al* [56] and Raicharoen *et al* [57].

2.2.3.4 *Combination with other classifiers*

The combination of two or more classifiers is done with the aim of obtaining more accurate classifiers at the expense of increasing their complexity [58]. *k*NN has been combined in three different ways. First, *k*NN has been combined with variations of itself. For example, Wilson [59] used *k*NN to reduce the number of objects in the dataset and then used 1NN to classify unknown objects. Second, *k*NN has been combined with other methods such as LDA [60]; support vector machines [61]; multi-label learning [62]; fuzzy methods [63, 64]; classification trees [65]; Lineal Discriminant Analysis [45] and Differential Evolution to optimization problems [66]. Finally, *bagging* (***B****ootstrap* ***AGG****regat****ING***) has been used for generating multiple versions of *k*NN [36]. In this case, the classifiers are built on bootstrap replicates of the training set. A bootstrap replicate (also called bootstrap sample) is a new dataset generated by sampling with replacement from the original training set [67]. Then, for each bootstrap sample, a given unknown object is classified using *k*NN. This procedure is repeated *B* times and finally the unknown object is classified in the class in which it was more frequently classified [36](see section 2.6). Breiman [36] argued that bagged *k*NN does not improve classification results because *k*NN is very stable. This conclusion was obtained using the majority vote as the classification rule for bagging, i.e. an object is assigned to the class in which it was more frequently classified. However, Hamamoto *et al.* [68] developed a new method to obtain the bootstrap training set, in which the new objects are created,

not selected, from the original dataset. The new bootstrap training set is then used to classify the unknown objects using *k*NN. Although this method has low classification error rates, it does not provide the value of reliability of classification.

2.2.4 Probabilistic approach

Several probabilistic approaches had been proposed for classification using *k*NN. Of them, the best known uses the posterior probability. The probability that a given test object belongs to class c is given by [1]:

$$P(class/\mathbf{x_v}) = \frac{k_c}{k} \qquad Eq.\,2.3$$

where k_c is the number of nearest neighbours of the test object in the training set that belong to class c, and k is the number of neighbours considered for classification. In this case, the unknown object is classified in the class where the posterior probability is the largest:

$$\begin{matrix} unknown\ object\ \ \mathbf{x}_t \in class\ c \\ \text{if } P(class\ c/\mathbf{x}_t) \end{matrix} = \max_{c \in C} P(class/\mathbf{x_t}) \qquad Eq.\,2.4$$

Note that this measure of posterior probability has the inconvenience of being the same for any unknown object that has the same k_c. The unknonwn objects, however, can be in slightly different locations and be more or less close to their neighbours. Figure 2.3 shows the scatter plot of two variables with three training objects of class 1 (red points) and three training objects of class 2 (blue points). This figure also shows the Euclidean space for k = 3 for three unknown objects $\mathbf{x}_{t1}$, $\mathbf{x}_{t2}$ and $\mathbf{x}_{t3}$, indicated by the continuous, dashed and dotted line

respectively. In these Euclidean spaces there are two objects that belong to class 2 and one object that belongs to class 1, and though the unknown objects are in different positions in the variable space, they are classified in the same class with the same value of posterior probability (2/3 = 0.66). Intuitively, the closer the unknown object is to their neighbours, the more reliable the classification should be. This, however, is not accounted for by the value $k_c|k$. Moreover, this probability measure only takes a few discrete values, e.g., for k=3, it only takes values of 0, 1/3, 2/3 and 1. One would expect that the reliability should change continuously for different positions of the unknown object in the variable space.

Steel and Patterson [69] developed an analytical formula for the calculation of the ideal bootstrap estimate prediction error for kNN. However, this formula is complex and the authors only recommend its use for values of k less than five, otherwise it requires excessive computational effort.

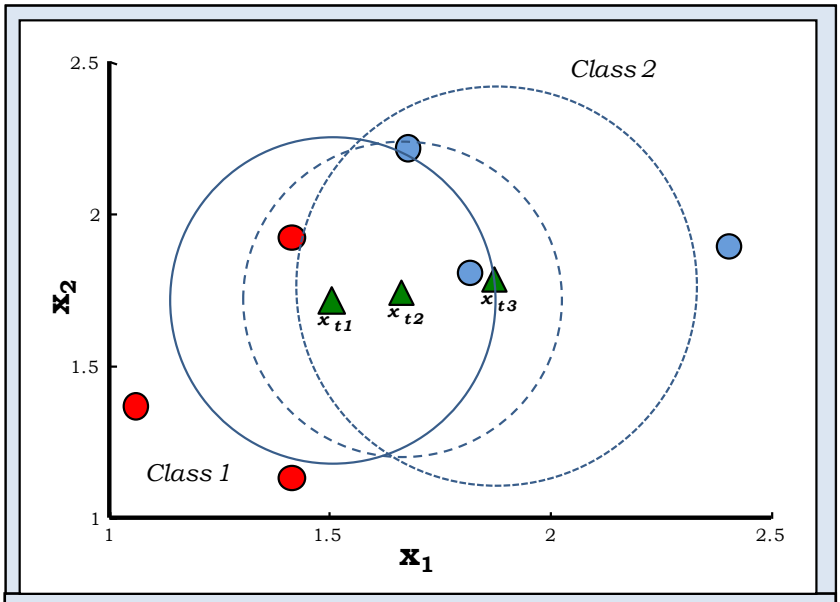

Figure 2.3 Classification of three objetcs by *k*NN. All objects (green triangles) are classified in the same class (class 2) with the same posterior probability value.

2.3 Other classification methods

Excellent books and reviews on supervised classification methods can be found elsewhere [1, 4, 6, 40]. In this section we only describe other classification methods used in this thesis.

2.3.1 Bayes' decision rule

The Bayes' decision rule is widely used in pattern recognition [1, 13]. It assigns an unknown object to the class with the highest posterior probability. The posterior probability that the real class is c given that the vector of features, $\mathbf{x}_t$ has been measured for object to be classified, is computed as:

$$P(class\ c|\mathbf{x_t}) = \frac{p(\mathbf{x_t}|class\ c)\ P(class\ c)}{p(\mathbf{x_t})} \qquad Eq.2.5$$

where $p(\mathbf{x_t}|class\ c)$ is a value of the class conditional probability density function, $P(class\ c)$ is the prior class probability and $p(\mathbf{x_t})$ is the evidence factor, which is used to scale the probability value between 0 and 1 [1], which is computed as:

$$p(\mathbf{x_t}) = \sum_{c=1}^{C} p(\mathbf{x_t}|class\ c)P(class\ c) \qquad Eq.2.6$$

It is to note that the Bayes' decision requires the probability density functions of each class $p(\mathbf{x_t}|class\ c)$ to be known, which is not always easy.

2.3.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) finds boundaries between classes [4]. LDA assigns an unknown object $\mathbf{x_t}$ to the class with the smallest discriminant score (d_s). This discriminant score is given by [70]:

$$d_s = (\mathbf{x_t} - \bar{\mathbf{x}}_c)^T \sum_{pooled}^{-1} (\mathbf{x_t} - \bar{\mathbf{x}}_c) - 2\ln P(class\ c) \qquad Eq.2.7$$

where, $\bar{\mathbf{x}}_c$ is the class centroid; $(\dots)^T$ indicates transposition; $P(class\ c)$ is the prior probability of the class c and $\sum_{pooled}^{-1}$ is the inverse of the pooled covariance matrix. The pooled covariance matrix is used because in LDA the class covariance matrices are assumed equal [70].

An extensive explanation of LDA and some application examples in chemistry and related fields can be found in references [70-72].

The Bayes' rule and LDA are used in this thesis as reference methods. The Bayes' decision rule is considered to be the optimal method of classification [1, 5] because it minimizes the conditional risk of classification, i.e. minimizes the error rate of classification. LDA, on the other hand, has been widely studied and has many applications [6].

Both the Bayes' rule and LDA require the fulfilment of several requisites for classification. i.e. in Bayes the multivariate normality density functions must be known [1], while LDA needs the number of the objects be higher than the number of variables to avoid singularity [70].

2.4 Multivariate calibration

Multivariate calibration [73-78] is one of the cornerstones of chemometrics. In general terms, multivariate calibration is used for predicting properties of interest (y_1, y_2,…,y_q), for example concentration, from a number of predictor variables (x_1, x_2,…,x_J), for example spectra [79, 80]. Both properties and predictor variables are related by the calibration model [81]. Linear models, i.e., a linear relationship between the dependent variables and the model coefficients are the most common in multivariate calibration. For a set of training data, the linear model can be represented by:

$$\mathbf{y} = \mathbf{X\beta} + \mathbf{e_Y} \qquad Eq.\,2.8$$

where $\boldsymbol{y}$ is the $I\times1$ vector of the predictand or the property to be estimated, $\mathbf{X}$ is the $I\times J$ matrix of predictor variables, $\boldsymbol{\beta}$ is the $J\times1$ vector of "true" model coefficients, which must be estimated, and $\mathbf{e_Y}$ is the error.

Multivariate calibration involves the following steps [4, 10, 82], which we describe below in more detail: data collection, data preprocessing, calculation, optimization, validation of the model and prediction.

2.4.1 Data collection

A measurements matrix (i.e. spectra), $\mathbf{X}$, of a considerable number of training objects should ideally be obtained from the same population and should include all possible sources of physical and chemical variability to be found in the future objects to be predicted. The property of interest, y, must also be measured in these objects by a suitable reference method. If the number of objects is rather large, $\mathbf{X}$ can be divided into a calibration or training set (with I_{cal} objects) and test set (with I_{val} objects).

2.4.2 Preprocessing

Preprocessing is used to adequately scale the data and suppress the contribution in the measured $\mathbf{X}$ that is not related to the property of interest with the aim of simplifying the model and increasing the accuracy and precision of the results. Existing preprocessing methods include mean-centering, autoscaling, multiplicative scatter correction (MSC), first and second derivative, standard normal variate (SNV), offset correction and others [8, 12, 19] .

2.4.3 Calculation of the model

A variety of multivariate calibration algorithms have been reported [77, 78, 83]. PLS uses the information contained in both **X** and **y**, during the calibration and compresses the data in such a way that the maximal variances in both **X** and **y** is explained [10]. Details about PLS can be found elsewhere [77, 78, 84, 85] . In this thesis, PLS is used as a reference method because it is the most representative among the multivariate calibration methods [86] and probably the most popular [87, 88].

2.4.4 Optimization of the model.

Once the model is obtained it will be optimized. A critical step in PLS, which required to be optimized, is the selection of the number of factors used in the model. One of the most frequent strategies used to select the number of factors is cross-validation. In this case the calibration set is used to obtain the fitting error, it is evaluated with the cross validation root mean square error of prediction (CV-RMSEP) obtained using the first PC with *Eq.* 2.9 (see section 2.4.5 for details). Then PCs 2, 3, 4, etc. are used to obtain the CV-RMSEP. The values of CV-RMSEP obtained are plotted against the numbers of PCs used for the calculations. The number of PCs is selected as the PC which the valued of CV-RMSEP is minimized. Other strategies to optimize the model include: variable selection, outliers elimination, preprocessing, etc [89, 90].

2.4.5 Validating the model.

The aim of the validation is to determine the accuracy of a multivariate calibration model. With this purpose, the calibration model is applied to a validation set, and the root mean square error of prediction, RMSEP, is computed as

$$\text{RMSEP} = \sqrt{\left(\frac{1}{I_{val}} \sum_{i_{val}=1}^{I_{val}} \left(\hat{y}_{i_{val}} - y_{i_{val}}\right)^2\right)} \qquad Eq.\,2.9$$

where, I_{val} is the number of the objects in the validation set, and $y_{i_{val}}$ and $\hat{y}_{i_{val}}$ are the reference and the predicted value of the evaluated property in the object *i*, respectively.

Cross-validation (LOOCV and *m*-fold cross validation, see section 2.1.4) can also be used to estimate the predictive ability of the model. In this case, *I*–1 and *I*–*m* objects are used to build the model and the left-out objects are predicted. This procedure is repeated until all objects have been predicted. Finally the cross-validation root mean square error of prediction, CV-RMSEP, is computed using *Eq.* 2.9, by replacing $\hat{y}_{i_{val}}$ by $\hat{y}_{(i)}$ which is the value of the property in the object $i (i = 1,\ldots,I)$ obtained by cross-validation.

2.4.6 Prediction

Once the calibration model has been validated it is used to predict the property of interest in unknown objects, $\mathbf{x}_t$:

$$\hat{\mathbf{y}} = \mathbf{x}_t^{\mathbf{T}}\mathbf{b} \qquad Eq\ 2.10$$

where **b** is the vector of regression coefficients and $\hat{\mathbf{y}}$ is the predicted property.

2.4.7 Prediction uncertainty

Each prediction should ideally be reported together with an estimate of its uncertainty [91, 92]. The uncertainty is defined as a parameter, associated with the result of a measurement, which characterizes the dispersion of the values that could reasonably be attributed to the measurand [93]. Several approaches have been proposed to compute the object-specific uncertainty in multivariate calibration. The most often used are: the U-deviation expression [94], which was improved by De Vries and Ter Braak [95], the errors-in-variables (EIV) approach [96] and resampling methods [97].

The improved U-deviation expression was implemented in the Unscrambler® software package [94, 95, 98]. Another expression for object-specific uncertainty is based on the Error-In-Variables (EIV) model and includes the measurement of errors in both the predictor and predicted variables. The estimated standard deviation of prediction is obtained by

$$\hat{\sigma}_{\hat{y}_{i,\mathrm{pred}}} = \left[\mathrm{RMSEC}^2 \times \left(1 + h_{i,\mathrm{pred}}\right) - \sigma_{y_{cal}}^2\right]^{1/2} \qquad Eq.\ 2.11$$

where $\sigma^2_{y_{cal}}$ is an estimate of the precision of the reference method. Details on the use of this formula are described elsewhere [94, 96, 99].

Another approach to compute the uncertainty in multivariate calibration is to use resampling methods [85, 100]. One of these methods is bootstrap. In bootstrap many new data sets are created by sampling with replacement from the original data [101]. This method has also been used, for example, to compute the uncertainty of multivariate regression coefficients [102] and the uncertainty in prediction of samples of bilinear [103] and three-way methods [104, 105]. See section 2.6 for details about bootstrap and confidence intervals obtained with bootstrap.

2.4.8 Outlier detection

Outlier detection is an important aspect in the development of a calibration model. An outlier is any measured value or predicted value that is significantly different from the rest of the data [106]. Outliers both in calibration and in validation must be detected and removed.

Outliers during the calibration stage can damage the model fitting [107]. In the prediction stage, the detection of outliers will increase the confidence in the predictions [73]. Several methods have been developed to detect outliers both at the training and at the prediction stage [73, 108-113]. The success of multivariate calibration models depends of its correct application and interpretation.

2.5 *k*NN for prediction

*k*NN can be used to predict continuous properties like other inverse calibration methods [114]. A property of an unknown object is predicted in *k*NN by finding the *k* nearest neighbours for this object in the training data matrix **X** and calculating the weighted mean [115],

$$\hat{y}_v = \sum_{i=1}^{k} w_i y_i \qquad Eq.2.12$$

with

$$w_i = \frac{e^{-d_i}}{\sum_{i=1}^{k} e^{-d_i}}$$

where y_i is the property value of the *i*th nearest neighbour (i=1,2,...,k), and k is the number of nearest neighbours considered in the prediction. In this case, the y values of the k nearest neighbours are weighted by the distance of the unknown object, d_i, to its nearest neighbours in **X**, so that a neighbour with a smaller distance is given a higher weight, w_i.

The prediction can alternatively be obtained as a mean of the y_i values of these neighbours (3) [116, 117]:

$$\hat{y}_v = \frac{\sum_{i=1}^{k} y_i}{k} \qquad Eq.2.13$$

Other prediction methods using *k*NN have been described by Nigsch *et al* [118]. In them, besides the weighted and arithmetical mean described by *Eq.* 2.12 and *Eq.* 2.13 respectively, they propose a geometrical average (*Eq.* 2.14) and an average weighted by the inverse distance (*Eq.* 2.15), to compute the predicted value:

$$\hat{y}_v = \prod_{i=1}^{k} y_i^{1/k} \qquad Eq\ 2.14$$

$$\hat{y}_v = \sum_{i=1}^{k} wd_i y_i \qquad Eq\ 2.15$$

where

$$w_{d_i} = \frac{1}{d_i} \frac{1}{\sum_{i=1}^{k} \frac{1}{d_i}}$$

2.5.1 Chemical and related applications

*k*NN for prediction has been used in structure-activity/property relationships (QSAR) studies to predict the volume of distribution at steady state and clearance of antimicrobial agents in humans using a quantitative structure-pharmacokinetic parameters relationship model [115] and the toxicity activity and anticonvulsant activity in different compounds [119], among others [116, 120, 121]. It has also been used to predict melting points of organic molecules and drugs [118], fat content in samples of chopped meat [122] and clime reconstruction from fossil pollen remains [117].

2.5.2 Advantages and disadvantages of *k*NN in prediction

Prediction with *k*NN has similar advantages than classification with *k*NN. i.e. it is conceptually and computationally quite simple and new

objects can be added to the training data without the need to recalculate a model (as it happens in PLS). Moreover, *k*NN can be considered as an approach to inverse calibration, because it is not necessary to know all the species present in a sample to predict the property of interest. It can be used as a non parametric method, since *k*NN does not require the probability distribution function of the data to be known. Another interesting advantage is its robustness to the presence of outliers in **X**. Most inverse calibration methods calculate a latent variable space that might be largely influenced by the presence of outliers in **X.** However, *k*NN also presents disadvantages as: 1) limited prediction ability compared with PLS, 2) it is quite sensitive to data preprocessing [19], and 3) no method to compute the prediction uncertainty has been developed yet.

2.6 Bootstrap

2.6.1 Notation and bootstrap generalities

This section first reviews the statistical concepts that are needed to understand bootstrap.

A population, $\mathcal{U}$, is a set of objects with similar characteristics about which we need some information. Information about the population is usually obtained by examining a small subset of its objects, called a statistical sample. A statistical sample is a collection of I objects taken randomly and with equal probability from a population [123]. Notice that the term sample is used in a statistical way, and not in a chemical way, where a sample is an object which should be analyzed to determine a property [100]. These samples can be selected with or without replacement, depending whether the selected objects are

returned or not to the population once they have been selected. This means that, with replacement, an object can appear more than once in a sample, whereas without replacement it cannot.

Statistical inference is used to describe the relationship between a sample and the population from which it was drawn [79, 124]. For an independent sample, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$, with I objects extracted from a population using a given probability distribution function, $\mathcal{F}$, an empirical distribution function exists, $\hat{\mathcal{F}}$, which is considered a discrete distribution with a probability, $1/I$, assigned to each object in the sample. The hat symbol "∧", like in $\hat{\mathcal{F}}$, is used to indicate that the value is an estimation obtained from the sample. A sample taken from a probability distribution function, $\mathcal{F}$, which assigns a probability value to each selected object, can be represented as [123]:

$$\mathcal{F} \rightarrow (x_1, x_1, \ldots, x_I) \qquad Eq.2.16$$

The probability distribution function can be used to build a statistic $\hat{\theta}$, for a parameter, θ [125]. A parameter, θ, is a numerical quantity that describes a population. It can be written as $\theta = t(\mathcal{F})$**,** i.e. the value of θ is obtained by applying the function $t(\cdot)$ to the distribution function $\mathcal{F}$, for example the true mean of $\mathcal{F}$, $\mu(\mathcal{F})$. A statistic, $\hat{\theta}$, is obtained from the sample $\mathbf{X}$ the sample mean, $\hat{\theta} = t(\hat{\mathcal{F}}) = \overline{\mathbf{X}}$ [79, 123, 125]. When, for a sample, a large number of objects are available and/or when the probability distribution is known or it can be assumed, it is possible to use analytical formulas to compute the statistic and make inference about the population. This is a parametric approach. In a nonparametric approach, the probability of a distribution function is unknown or not assumed. In this case, it is necessary to use

nonparametric methods for making such inferences; one of these methods is bootstrap. The expression "bootstrap" derives from the phrase to pull oneself up by one's bootstrap (Adventures of Baron Munchausen, by Rudolph Erich Raspe [123]). In bootstrap, the data are resampled with replacement many times (*B*), in order to generate empirical estimates of the statistics and use them to make inferences about the population [126]. Bootstrap is used to obtain an statistic, $\hat{\theta}$, from $\mathbf{X}$ using *B* bootstrap samples $\mathbf{X}^*$ drawn from a distribution close to the unknown distribution $\mathcal{F}$. In our example, the bootstrap mean should be obtained as:

$$\hat{\theta}^* = t(\hat{\mathcal{F}}^*) = \overline{\mathbf{X}}^*$$

Notice that the asterisk is used to indicate the bootstrap sample. It is important to know how the statistic $\hat{\theta}$ is distributed around the parameter, θ. For this, bootstrap is used to obtain the distribution of $\hat{\theta}^*$ around $\hat{\theta}$. This distribution should be due, principally, to the random variation or to the systematic error, which can be measured by the standard error and the bias, respectively [127].

Several bootstrap setups have been developed [123, 125, 127-129]. Of these, the basic bootstrap method, proposed originally by Efron, consists on the following steps [67]:

1. For a given sample $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$, i.e. the original sample**,** build the sample probability distribution $\hat{\mathcal{F}}$, by giving an equal probability of $1/I$ to each object $\mathbf{x}_1,\ldots,\mathbf{x}_I$.

2. From $\hat{\mathcal{F}}$, draw a random sample of size I with replacement. This is the bootstrap sample $\mathbf{X}_b^* = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_I^*\}$.
3. Compute the statistic of interest, $\hat{\theta}$, from this resample, yielding a bootstrap replication, $\hat{\theta}_b^*$.
4. Repeat steps 2 and 3 a large number of times, B.
5. Use the B different $\hat{\theta}_b^*$ to obtain an approximation to $\hat{\theta}$ and its accuracy, which can be used to do inferences about θ.

Figure 2.4 shows how bootstrap works and table 2.2 illustrates Efron's bootstrap for a sample $\mathbf{x}$. Five bootstrap samples, $\mathbf{x}_b^*$, were generated and their respective bootstrap replications (i.e. $\hat{\theta}_b^* = \text{mean}$) were computed. Finally, the mean of the bootstrap replications was obtained, in this case $\hat{\theta}_B^*$ =9.47, which is similar to the mean estimated with the original sample, $\hat{\theta}$ =9.46.

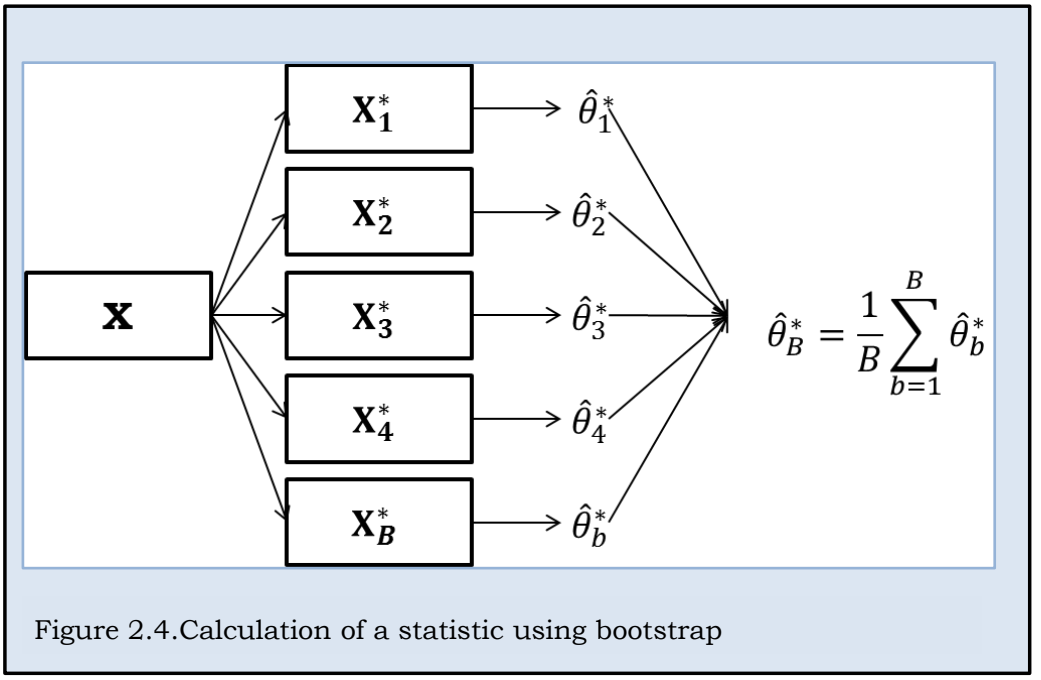

Figure 2.4.Calculation of a statistic using bootstrap

By bootstrapping many new bootstraps data set are obtained, by modifying of the variable space where the unknown sample will be classified. This strategy can improve the capability of prediction of the classifier.

Table 2.2. Efron's Bootstrap for a sample, **x,** obtained randomly from a normal distribution with $\mathcal{N}(\mu = 0 + 10, \sigma^2 = 1)$.

Objects	**x**	$\mathbf{x}_1^*$	$\mathbf{x}_2^*$	$\mathbf{x}_3^*$	$\mathbf{x}_4^*$	$\mathbf{x}_5^*$
x_1	9.37	10.94	9.37	9.37	9.36	7.67
x_2	7.67	7.88	11.06	7.88	9.37	10.94
x_3	8.77	9.30	9.36	9.89	10.94	11.06
x_4	11.06	7.88	9.37	9.30	11.06	10.38
x_5	9.89	7.67	7.67	9.89	9.36	7.67
x_6	10.38	9.89	8.77	9.89	10.38	10.94
x_7	10.94	9.30	7.67	9.36	7.88	11.06
x_8	7.88	9.30	10.94	10.38	9.89	9.36
x_9	9.36	9.89	8.77	8.77	11.06	9.36
x_{10}	9.30	9.36	7.67	10.94	7.67	10.38
Average	$\hat{\theta}$ =9.46	$\hat{\theta}_1^*$ =9.14	$\hat{\theta}_2^*$ =9.07	$\hat{\theta}_3^*$ =9.57	$\hat{\theta}_4^*$ =9.70	$\hat{\theta}_5^*$ =9.88
$\hat{\theta}_B^*$ =9.47						

2.6.2 Bootstrap applications

Wehrens *et al.* [127] distinguish between three types of applications in bootstrap: point estimates (i.e. bias or standard error), interval estimates (i.e. confidence intervals) and hypothesis testing. Here we describe the methods to obtain the point estimate and confidence intervals used in this work. Hypothesis testing has been described elsewhere [128, 130].

2.6.2.1 *Point estimates*

In bootstrap, the statistic of interest is often calculated as a mean of the B bootstrap replications, $\hat{\theta}_B^*$, obtained following the procedure described above. When the statistic of interest is a single number it is called a *point estimate* [127]. Besides obtaining the statistic, bootstrap is also used to obtain a measure of its accuracy. For this, the *standard error* and the *bias* are the most often employed measures [123, 131].

The standard error is used to know the variation around a mean value. Using bootstrap, the standard error, $\widehat{se}_B$, is estimated by:

$$\widehat{se}_B = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}_b^* - \hat{\theta}_B^*)} \qquad Eq.2.17$$

where $\hat{\theta}_B^*$ is the statistic estimated by bootstrap (i.e. the mean of B bootstrap replications, $\hat{\theta}_b^*$) computed as:

$$\hat{\theta}_B^* = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b^* \qquad Eq.2.18$$

The bias, which is the difference between the statistics of interest, $\hat{\theta}$, obtained with the original sample and the value of the parameter, θ [123], is estimated by:

$$\widehat{bias}_B = \hat{\theta}_B^* - \hat{\theta} \qquad Eq.2.19$$

If the bias is larger than the standard error, the statistic can be corrected as follows [127]:

$$\hat{\theta}_{bc} = \hat{\theta} - \widehat{bias}_B = 2\hat{\theta} - \hat{\theta}^*_B \qquad Eq.2.20$$

where $\hat{\theta}_{bc}$ is the statistic corrected with the bias. To carry out this correction it is recommended that the absolute value of the bias be bigger than 0.25 times the standard error [123].

2.6.2.2 *Jackknife and nested bootstrap*

Jackknife and nested bootstrap can also be used to calculate the point estimates. Jackknife is another resampling method and it has been used for estimating the bias, the standard error and confidence intervals [123, 132]. It was proposed by Tukey in 1958 [133] as an improvement of the method described by Quenouille [134, 135]. By jackknifing, a statistic, $\hat{\theta}$ (for example the mean) is obtained from I jackknife replications, $\hat{\theta}_{jack,i}$ (Figure 2.5).

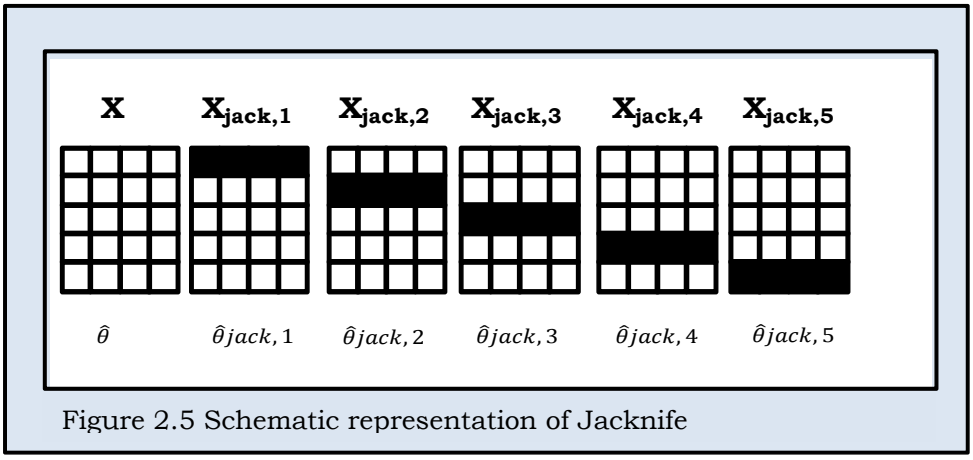

Figure 2.5 Schematic representation of Jacknife

For this, each $\hat{\theta}_{jack,i}$ is computed using its respective jackknife sample $\mathbf{X}_{jack,i}$ which is the original sample $\mathbf{X}$ without the object i, indicated with the black cell in the figure 2.5. This procedure is repeated until all I objects have been removed of the original sample and the I jackknife replications have been computed. Finally, the jackknife estimate of $\hat{\theta}$ is obtained as the mean of the I jackknife replications, as:

$$\hat{\theta}_{jack} = \frac{\sum_{i=1}^{I} \hat{\theta}_{jack,i}}{I} \qquad Eq.2.21$$

The bias and the standard error of an estimate are computed by jackknife using:

$$\widehat{bias}_{jack} = (I-1)(\hat{\theta}_j - \hat{\theta}) \qquad Eq.2.22$$

and

$$\widehat{se}_{jack} = \sqrt{\frac{I-1}{I}\sum_{i=1}^{I}(\hat{\theta}_{j.i} - \hat{\theta}_j)} \qquad Eq.2.23$$

Notice that these formulas differ from the estimations with bootstrap, since an "*inflation factor*" is used in jackknife. Those "*inflation factors*" are $\frac{(I-1)}{I}$ and $(I-1)$ for standard error and bias, respectively [123].

They are included with the aim of building unbiased estimates of the parameter. These biases may occur because all jackknife replications are more similar to the value of the parameter obtained in the original data than in the bootstrap replications, the reason being that the jackknife samples are more similar to the original sample than the

bootstrap samples. For more information about jackknife see [131, 132, 136, 137]. The main disadvantage of jackknife, compared to bootstrap, is that jackknife can only be used for estimating parameters with continuous (smooth) values. This means that jackknife must be used when slight changes in the data cause slight changes in the statistic (e.g. mean) of interest [123]. Bootstrap, however, can be used to compute almost any statistic [138].

Nested bootstrap is based on the principle of resampling from bootstrap samples. For this reason it is also called double bootstrap or bootstrapping the bootstrap [128]. Nested bootstrap is used when bootstrap does not offer correct answers of point or interval estimates.

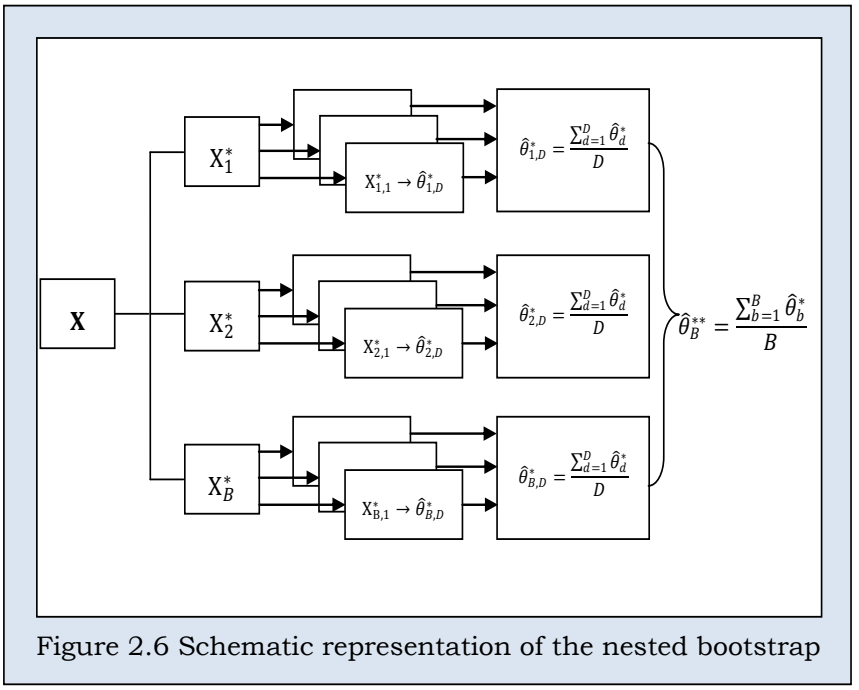

Figure 2.6 Schematic representation of the nested bootstrap

Figure 2.6 shows a scheme of the nested bootstrap, which is performed as follows:

1. From $\mathbf{X}$, a bootstrap sample, $\mathbf{X}_b^*$ is obtained by resampling with replacement.
2. $\mathbf{X}_b^*$ is also resampled to obtain a new bootstrap sample, $\mathbf{X}_d^*$.
3. $\mathbf{X}_d^*$ is used to compute the statistic of interest, $\hat{\theta}_d^*$.
4. Repeat steps 2 and 3 a D times.
5. Use the D different $\hat{\theta}_d^*$, to obtain an approximation to $\hat{\theta}_b^*$.
6. Repeat steps 2 to 5 a large number of times, B.
7. Use the B different $\hat{\theta}_b^*$, to obtain the nested bootstrap statistic, $\hat{\theta}_B^{**}$, which is an approximation to $\hat{\theta}$.

2.6.2.3 *Confidence intervals*

Confidence intervals provide an estimation of the uncertainty of a statistic [139]. In this thesis bootstrap was used to obtain the confidence intervals of predictions with kNN. In a statistical way, the confidence interval of a statistic $\hat{\theta}$, obtained with an α-significance level, can be defined as the interval that will include the true value of θ with a $[(1-2\alpha)\times 100]\%$ confidence [123]. Typically, the confidence interval of a parameter of interest, θ, with a specific degree of confidence, is obtained from the appropriate statistic, $\hat{\theta}$, and using its estimated standard error, $\widehat{se}$. For example, if a normal distribution is assumed, the confidence interval of $\hat{\theta}$ is obtained by:

$$\hat{\theta} \pm z_{(\alpha)} \cdot \widehat{se} \qquad Eq.2.24$$

where $z_{(\alpha)}$ is the $100\alpha th$ percentile of the normal standard distribution.

Calculation of confidence intervals is the major application of nonparametric bootstrap [131] and several methods have been developed: basic, percentile, percentile-*t*, bias corrected and accelerated [123, 128, 140], among others [141, 142]. The aim is to build a confidence interval $(\hat{\theta}_{\text{low}}, \hat{\theta}_{\text{up}})$ of an estimate, $\hat{\theta}$, of a given parameter θ from B bootstrap replications, θ_b^*, obtained using bootstrap samples $\mathbf{X}_b^*$. In all cases the confidence values $(\hat{\theta}_{\text{low}}, \hat{\theta}_{\text{up}})$ are obtained from the values in the bootstrap distribution found in the α-percentiles given by $(B+1)\alpha$-th ordered values of the bootstrap distribution (i.e. bootstrap replications in ascending order, $\hat{\theta}_1^* \leq \hat{\theta}_2^* \leq \cdots \leq \hat{\theta}_B^*$)[127].

Considering that the confidence interval of a given parameter θ should be obtained from its statistic, $\hat{\theta}$ using its percentiles, τ_p, the $1-2\alpha$ interval of $\hat{\theta}$ with both left and right errors equal to α, is limited by [128]:

$$\hat{\theta}_{\text{low}} = \hat{\theta} - \tau_{1-\alpha} \ , \ \ \hat{\theta}_{\text{up}} = \hat{\theta} - \tau_{\alpha} \qquad Eq.2.25$$

which can also be expressed as:

$$\hat{\theta} - \tau_{1-\alpha} \leq \ \ \theta \leq \hat{\theta} - \tau_{\alpha} \qquad Eq.2.26$$

Eq 2.26 is considered the base of the confidence intervals obtained using bootstrap, since it is the reference to obtain them in the different bootstrap confidence intervals methods.

2.6.2.3.1 Bootstrap basic method

This method is based on the same rules used to compute the standard error and the bias. For this reason in this method the distribution of $(\hat{\theta} - \theta)$ is approached using the probability of distribution of $(\hat{\theta}^* - \hat{\theta})$ obtained with bootstrap. Likewise, the percentiles of that distribution, τ_p^*, are approximates to τ_p. They are used to compute the confidence intervals as [127]:

$$\hat{\theta} - \tau^*_{(B+1)(1-\alpha)} \leq \theta \leq \hat{\theta} - \tau^*_{(B+1)(\alpha)} \qquad Eq.2.27$$

where, $\tau^*_{(B+1)(\alpha)}$ and $\tau^*_{(B+1)(1-\alpha)}$ are the percentiles of the distribution of $(\hat{\theta}^* - \hat{\theta})$ for given B and α values.

The percentiles of the distribution of $(\hat{\theta}^* - \hat{\theta})$ can be related to the percentiles of the distribution of the statistic $\hat{\theta}^*_\alpha$ and can be expressed as [127]:

$$\hat{\theta}^*_\alpha = \hat{\theta} + \tau^*_\alpha \qquad Eq.2.28$$

Using Eq 2.28, and replacing the percentile, τ^*_α in Eq 2.27, the confidence interval in the basic method can be obtained as,

$$\hat{\theta} - \left(\hat{\theta}^*_{(B+1)(1-\alpha)} - \hat{\theta}\right) \leq \theta \leq \hat{\theta} - \left(\hat{\theta}^*_{(B+1)(\alpha)} - \hat{\theta}\right) \qquad Eq.2.29$$

or, rearranging, as:

$$2\hat{\theta} - \hat{\theta}^*_{(B+1)(1-\alpha)} \leq \theta \leq 2\hat{\theta} - \hat{\theta}^*_{(B+1)(\alpha)} \qquad Eq.2.30$$

where, $\hat{\theta}^*_{(B+1)(\alpha)}$ and $\hat{\theta}^*_{(B+1)(1-\alpha)}$ are the percentiles of the distribution of $\hat{\theta}^*$ for given B and α values. For example, if $B = 999$ and $\alpha = 0.05$; $\hat{\theta}^*_{(B+1)(1-\alpha)}$ and $\hat{\theta}^*_{(B+1)(\alpha)}$ correspond to the sorted values of $\hat{\theta}^*$ in the positions 50th and 950th. Notice that it is important that $(B+1)(\alpha)$ be integer numbers, otherwise an interpolation must be used [128]. For this reason, it is recommended to use odd numbers for B (e.g. 999 or 1999) to avoid interpolations [127].

2.6.2.3.2 Bootstrap percentile method

This method, also called Efron's percentile method, provides a $(1-2\alpha)$ nonparametric confidence interval for θ. For this, the B different bootstrap replications, $\hat{\theta}^*_b$, obtained using bootstrap are sorted from smallest to largest. Then, the values located in the positions $(B+1)\alpha$ and $B-(B+1)\alpha$ are selected as the upper and lower limits of the confidence interval around θ with an α-significance level. This can be expressed as:

$$\text{BPercentile}\left(\hat{\theta}_{\text{low}}, \hat{\theta}_{\text{up}}\right) = \left(\hat{\theta}^*_{(B+1)\alpha}, \hat{\theta}^*_{((B+1)(1-\alpha)}\right) \qquad Eq.2.31$$

For example, for a $1-2\alpha$ confidence interval of a given $\hat{\theta}$ with $\alpha = 0.05$ and $B = 99$, the values of $\hat{\theta}^*_b$ in the positions 95th and 5th would be the upper and lower limits of the confidence interval.

2.6.2.3.3 Bootstrap t-intervals method

This method, also known as Studentized method or percentile *t*-method [123, 126, 129], shares the form of the classical *t*-interval, but in this case the use of the *t*-table for critical values is not required and the bootstrap method is used to replace it [139]. The confidence interval is obtained as:

$$\mathrm{B}t\left(\hat{\theta}_{\mathrm{low}}, \hat{\theta}_{\mathrm{up}}\right) = \left(\hat{\theta} - \hat{t}_{(1-\alpha)} \cdot \widehat{se}, \quad \hat{\theta} - \hat{t}_{(\alpha)} \cdot \widehat{se}\right) \qquad Eq.2.32$$

A requirement for this method to be applied is that an approach to the statistic *t*-distribution, $\hat{t}$, has to be obtained from the B bootstrap estimates $\hat{\theta}_b^*$. For this, $\hat{\theta}_b^*$ has to be transformed into a standardized variable $\hat{t}_b^*$ by:

$$\hat{t}_b^* = \frac{\hat{\theta}_b^* - \hat{\theta}}{\widehat{se}_b^*} \qquad Eq.2.33$$

where $\hat{\theta}$ is the estimated parameter (statistic) of the original sample (also called observed estimated parameter) and $\widehat{se}_b^*$ is the estimated standard error of $\hat{\theta}_b^*$. $\widehat{se}_b^*$ should be estimated analytically, i.e. using a known formula, using a second level of bootstrap (i.e. nested bootstrap) or using jackknife [123, 126, 127].

2.6.2.3.4 Bootstrap bias-corrected and accelerated method (BCa)

Bias-corrected and accelerated confidence intervals are an improvement of the percentile method [123]. The difference consists in

the form in which the position of the values of the interval is computed. While in the percentile method they are obtained using the number of the bootstrap replication and a given alpha value (Eq 2.31), in BCa the confidence intervals are obtained using:

$$\mathrm{BCa}\left(\hat{\theta}_{\mathrm{low}}, \hat{\theta}_{\mathrm{up}}\right) = \left(\hat{\theta}^{*(\alpha 1)}, \hat{\theta}^{*(\alpha 2)}\right) \qquad Eq.2.34$$

where $\alpha 1$ and $\alpha 2$ are computed as:

$$\alpha 1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{(\alpha)}}{1 - \hat{a}\left(\hat{z}_0 + z_{(\alpha)}\right)}\right)$$
$$\alpha 2 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{(1-\alpha)}}{1 - \hat{a}\left(\hat{z}_0 + z_{(1-\alpha)}\right)}\right) \qquad Eq.2.35$$

$\Phi(\cdot)$ is the standard normal cumulative distribution function, $\hat{z}_0$ is the bias-correction, $\hat{a}$ is the acceleration factor and $z_{(\alpha)}$ is the 100αth percentile of a normal standard distribution. The value of $\hat{z}_0$ is the proportion of bootstrap replications lower than the observed estimated parameter, $\hat{\theta}$, and can be obtained as:

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\#\{\hat{\theta}_b^* < \hat{\theta}\}}{B}\right) \qquad Eq.2.36$$

where $\#\{\hat{\theta}_b^* < \widehat{\theta}\}$ represents the number of $\hat{\theta}_b^*$ lower than $\hat{\theta}$, and Φ^{-1} is the inverse function of a standard cumulative distribution function. The acceleration, $\hat{a}$, indicates the rate of change of the standard error of $\hat{\theta}$ with respect to the true parameter value θ [123, 128]. The acceleration can be computed using jackknife, as:

$$\hat{a} = \frac{\sum_{i=1}^{I}\left(\hat{\theta}_{(\cdot)} - \hat{\theta}_{(i)}\right)^3}{6\left(\sum_{i=1}^{I}\left(\hat{\theta}_{(\cdot)} - \hat{\theta}_{(i)}\right)^3\right)^{3/2}} \qquad Eq\ 2.37$$

where, $\hat{\theta}_{(\cdot)}$ is the statistic estimated by jackknife from the original sample and $\hat{\theta}_{(i)}$ is the jackknife replicate [123]. The acceleration can also be obtained using a nested bootstrap [128].

2.7 Reliability

Reliability is a fundamental concept in Analytical Chemistry. For example, in quality accreditation laboratories, the International Laboratory Accreditation Cooperation (ILAC) and the United Nations Industrial Development Organization (UNIDO), in a working paper about laboratory accreditation in developing economies state that " *reliable results are essential and the role of accreditation is to ensure that within certain acceptable (and quantifiable) limits, tests of any type made on a product in say the Far East can be repeated with confidence in any other country in the world"* [143]. Also, the reliability of analytical data is important for companies that must meet legal requirements (e.g. pharmaceutical, chemical, medical, etc), for organisations responsible of consumer protection (e.g. FDA) and for all laboratories with responsibility for quality control, quality assurance and method development [144]. For these reason, any result obtained with an analytical method should be accompanied by a measure of its confidence, i.e. a reliability value, which is consider a quantitative indication of the quality of a results [145].

The term reliability can be used in various ways, which has led to confusion. In a general sense, the reliability is used to express the probability that a system will operate without fault, i.e. any abnormal incident or accident, of a given period of time, [146]. In analytical data, however, the reliability is used to express a degree of confidence in the results [147]. In classification methods the term reliability is used in both senses, in terms of recognition ability of a classifier [13] and to express the confidence we have in the classification of a particular object [148].

2.7.1 Reliability of classification

The reliability in classification is measured, in a general way, using the non-error rate of classification (see section 2.1.4) [6]. More specifically, it is calculated as the probability of classification for a given object [2]. The generic or specific reliabilities are obtained depending on the classifier used. Not all classifiers provide a measure of the reliability or if it is provided this is not the optimal; for example, the classical *k*NN classifier (section 2.2.3.5). On the contrary, in probabilistic methods, such as the Bayes' rule, a value of the probability of classification is obtained.

2.7.2 Reliability of prediction

A measure of the reliability is required to assess the prediction capabilities of a calibration method. The reliability can be given globally by the RMSEP value or, specifically, by prediction intervals [149], which represent upper and lower confidence limits of the predicted values. Assuming that the prediction has a non-significant bias, the

size of these prediction intervals around the predicted value provides an indication of the precision of this predicted value (i.e. large limits mean less precision) [150].

Several methods have been proposed to compute the confidence limits in multivariate calibration (section 2.4.7)

2.8 References

1. Duda RO, Hart PE, Stork DG (2001) Pattern classification. John Wiley & Sons, New York.

2. González AG (2007) Use and misuse of supervised pattern recognition methods for interpreting compositional data. Journal of Chromatography A 1158:215

3. Brereton RG (2003) Chemometrics: Data analysis for the laboratory and chemical plant. John Wiley & Sons, Chichester

4. Vandeginste BGM, Massart DL, Buydens LM, Lewi PJ, Smeyers-Verbeke J, Jong SD (1998) Handbook of Chemometrics and Qualimetrics. Part B. Elsevier, Amsterdam.

5. Jain AK, Duin RPW, Jianchang M (2000) Statistical pattern recognition: a review. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22:4

6. Berrueta LA, Alonso-Salces RM, Héberger K (2007) Supervised pattern recognition in food analysis. Journal of Chromatography A 1158:196

7. Kowalski BR, Bender CF (1972) Pattern recognition. Powerful approach to interpreting chemical data. Journal of the American Chemical Society 94:5632

8. Wu W, Walczak B, Massart DL, Prebble KA, Last IR (1995) Spectral transformation and wavelength selection in near-infrared spectra classification. Analytica Chimica Acta 315:243

9. Haswell SJ (1992) Practical guide to chemometrics. Marcel Dekker, New York.

10. Sun D (2009) Infrared spectroscopy for food quality analysis and control. Academic Press, Amsterdam

11. Kennard RW, Stone LA (1969) Computer Aided Design of Experiments. Technometrics 11:137

12. Wu W, Guo Q, Jouan-Rimbaud D, Massart DL (1999) Using contrasts as data pretreatment method in pattern recognition of multivariate data. Chemometrics and Intelligent Laboratory Systems 45:39

13. Webb AR (2002) Statistical Pattern Recognition. John Wiley and Sons Ltd, New York

14. Riu J, Bro R (2003) Jack-knife technique for outlier detection and estimation of standard errors in PARAFAC models. Chemometrics and Intelligent Laboratory Systems 65:35

15. Chernick MR (2008) Bootstrap methods: a guide for practitioners and researchers. John Wiley and Sons, New York.

16. Fukunaga K, Hostetler L (1973) Optimization of k nearest neighbor density estimates. IEEE Transactions on Information Theory 19:320

17. Fukunaga K, Flick TE (1984) An optimal global nearest neighbor metric, IEEE Transactions on Pattern Analysis and Machine Intelligence 6:314

18. Short RII, Fukunaga K (1981) The optimal distance measure for nearest neighbor classification. IEEE Transactions on Information Theory 27:622

19. Todeschini R (1989) k-nearest neighbour method: The influence of data transformations and metrics. Chemometrics and Intelligent Laboratory Systems 6:213

20. Fix EaJ (1989) Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. International Statistical Review 57:238

21. Latourrette M (2000) Toward an explanatory similarity measure for nearest-neighbor classification. Machine Learning: ECML 238

22. Kowalski BR, Bender CF (1972) K-Nearest Neighbor classification rule (pattern recognition) applied to nuclear magnetic resonance spectral interpretation. Analytical Chemistry 44:1405

23. Bender CF, Kowalski BR (1974) Multiclass linear classifier for spectral interpretation (pattern recognition). Analytical Chemistry 46:294

24. Koskinen JR, Kowalski BR (1975) Interactive pattern recognition in the chemical laboratory. Journal of Chemical Information and Modeling 15:119

25. López B, Latorre MJ, Fernández MI, García MA, García S, Herrero C (1996) Chemometric classification of honeys according to their type based on quality control data. Food Chemistry, 55:281

26. Shaffer RE, Rose-Pehrsson SL, McGill RA (1999) A comparison study of chemical sensor array pattern recognition algorithms. Analytica Chimica Acta 384:305

27. Ruiz-Jiménez J, Priego-Capote F, García-Olmo J, Castro MDLd (2004) Use of chemometrics and mid infrared spectroscopy for the selection of extraction alternatives to reference analytical methods for total fat isolation. Analytica Chimica Acta 525:159

28. Alonso-Salces RM, Herrero C, Barranco A, Berrueta LA, Gallo B, Vicente F (2005) Classification of apple fruits according to their maturity state by the pattern recognition analysis of their polyphenolic compositions. Food Chemistry 93:113

29. Lukasiak BM, Faria R, Zomer S, Brereton RG, Duncan JC (2006) Pattern recognition for the analysis of polymeric materials. Analyst 131:73

30. Gonzalez-Arjona D, Lopez-Perez G, Gonzalez-Gallero V, Gonzalez AG (2006) Supervised pattern recognition procedures for discrimination of whiskeys from gas chromatography/mass spectrometry congener analysis. Journal of Agricultural and Food Chemistry 54:1982

31. Dragovic S, Onjia A (2007) Classification of soil samples according to geographic origin using gamma-ray spectrometry and pattern recognition methods. Applied Radiation and Isotopes 65:218

32. Chen Q, Zhao J, Vittayapadung S (2008) Identification of the green tea grade level using electronic tongue and pattern recognition. Food Research International 41:500

33. Li B, Wei Y, Duan H, Xi L, Wu X (2012) Discrimination of the geographical origin of Codonopsis pilosula using near infrared diffuse reflection spectroscopy coupled with random forests and k-nearest neighbor methods. Vibrational Spectroscopy 62:17

34. Fix E, Hodges JLJ (1951) Discriminatory Analysis: Nonparametric Discrimination: consistency Properties. International Statistical Review 57:261

35. Muller K-, Anderson CW, Birch GE (2003) Linear and nonlinear methods for brain-computer interfaces, IEEE Transactions on Neural Systems and Rehabilitation Engineering 11:165

36. Breiman L (1996) Bagging predictors. Machine Learning 24:123

37. Bay SD (1998) Combining nearest neighbor classifiers through multiple feature subsets. Proc. 17th International conference on Machine Learning 37-45

38. Coomans D, Massart DL (1982) Alternative k-nearest neighbour rules in supervised pattern recognition: Part 2. Probabilistic classification on the basis of the kNN method modified for direct density estimation. Analytica Chimica Acta 138:153

39. Yuan W, Liu J, Zhou H (2004) An improved KNN method and its application to tumor diagnosis. Machine Learning and Cybernetics 5:2836

40. Lavine BK (2006) Pattern Recognition. Critical Reviews in Analytical Chemistry 36:153

41. Fukunaga K, Hayes RR (1989) Effects of sample size in classifier design, IEEE Transactions on Pattern Analysis and Machine Intelligence 11:873

42. Reunanen J (2004) A pitfall in determining the optimal feature subset size. Proceedings of the Fourth International Workshop on Pattern Recognition in Information Systems 176-185

43. Sima C, Dougherty ER (2008) The peaking phenomenon in the presence of feature-selection. Pattern Recognition Letters 29:1667

44. Yang JM, Yu PT, Kuo BC (2010) A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data. IEEE Transactions on Geoscience and Remote Sensing 48:1279

45. Yang J, Zhang L, Yang J, Zhang D (2011) From classifiers to discriminators: A nearest neighbor rule induced discriminant analysis. Pattern Recognition 44:1387

46. Villegas M, Paredes R (2011) Dimensionality reduction by minimizing nearest-neighbor classification error. Pattern Recognition Letters 32:633

47. Wu W, Massart DL (1997) Regularised nearest neighbour classification method for pattern recognition of near infrared spectra. Analytica Chimica Acta 349:253

48. Parveen P, Thuraisingham B (2006) Face recognition using multiple classifiers. 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)179-186

49. He QP, Jin Wang (2008) Principal component based k-nearest-neighbor rule for semiconductor process fault detection. American Control Conference 1606-1611

50. Zhang Y, Zhou Z (2010) Multilabel dimensionality reduction via dependence maximization. ACM Transactions on Knowledge Discovery from Data 4:1

51. Hart P (1968) The condensed nearest neighbor rule (Corresp.). IEEE Transactions on Information Theory 14:515

52. Gates G (1972) The reduced nearest neighbor rule (Corresp.). IEEE Transactions on Information Theory 18:431

53. Angiulli F (2005) Fast condensed nearest neighbor rule. Proceedings of the 22nd international conference on Machine learning 25-32

54. Kuncheva LI (1995) Editing for the k-nearest neighbors rule by a genetic algorithm. Pattern Recognition Letter 16:809

55. Dasarathy BV, Sánchez JS, Townsend S (2000) Nearest neighbour editing and condensing tools–synergy exploitation. Pattern Analysis & Applications 3:19

56. Sánchez JS, Barandela R, Marqués AI, Alejo R, Badenas J (2003) Analysis of new techniques to obtain quality training sets. Pattern Recognition Letters 24:1015

57. Raicharoen T, Lursinsap C (2005) A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (POC-NN) algorithm. Pattern Recognition Letters 26:1554

58. Kuncheva LI (2004) Combining pattern classifiers: methods and algorithms. J. Wiley & Sons, Hoboken

59. Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics 2:408

60. Peng J, Heisterkamp DR, Dai HK (2003) LDA/SVM driven nearest neighbor classification. IEEE Transactions on Neural Networks 14:940

61. Pan F, Wang B, Hu X, Perrizo W (2004) Comprehensive vertical sample-based KNN/LSVM classification for gene expression analysis. Journal of Biomedical Informatics 37:240

62. Zhang M, Zhou Z (2007) ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition 40:2038

63. Petridis V, Kaburlasos V G. (2003) Finknn: a fuzzy interval number k-nearest neighbor classifier for prediction of sugar production from populations of samples. Journal of Machine Learning Research 4:17

64. Alsberg BK, Goodacre R, Rowland JJ, Kell DB (1997) Classification of pyrolysis mass spectra by fuzzy multivariate rule induction-comparison with regression, K-nearest neighbour, neural and decision-tree methods. Analytica Chimica Acta 348:389

65. Buttrey SE, Karo C (2002) Using k-nearest-neighbor classification in the leaves of a tree. Computational Statistics & Data Analysis 40:27

66. Liu Y, Sun F (2011) A fast differential evolution algorithm using k-Nearest Neighbour predictor. Expert Systems with Applications 38:4254

67. Efron B (1979) Bootstrap Method- Another look at the jacknife. The Annals of Statistics 7:1

68. Hamamoto Y, Uchimura S, Tomita S (1997) A bootstrap technique for nearest neighbor classifier design. IEEE Transactions on Pattern Analysis and Machine Intelligence 19:73

69. Steele B, Patterson D (2000) Ideal bootstrap estimation of expected prediction error for k-nearest neighbor classifiers: Applications for classification and error assessment. Statistics and Computing 10:349

70. Wu W, Mallet Y, Walczak B, Penninckx W, Massart DL, Heuerding S, Erni F (1996) Comparison of regularized discriminant analysis linear discriminant analysis and quadratic discriminant analysis applied to NIR data. Analytica Chimica Acta 329

71. Forina M, Armanino C, Castino M, Ubigli M (1986) Multivariate data analysis as a discriminating method of the origin of wines. Vitis 25:189

72. Todeschini R, Ballabio D, Consonni V, Mauri A, Pavan M (2007) CAIMAN (Classification And Influence Matrix Analysis): A new approach to the classification based on leverage-scaled functions. Chemometrics and Intelligent Laboratory Systems 87:3

73. Martens H, Naes T (1989) Multivariate calibration. Wiley, New York.

74. Forina M, Casolino MC, de la Pezuela Martinez C (1998) Multivariate calibration: applications to pharmaceutical analysis. Journal of Pharmaceutical and Biomedical Analysis 18:21

75. Brereton RG (2000) Introduction to multivariate calibration in analytical chemistry. Analyst 125:2125

76. Bro R (2003) Multivariate calibration: What is in chemometrics for the analytical chemist? Analytica Chimica Acta 500:185

77. Gabrielsson J, Trygg J (2006) Recent developments in multivariate calibration. Critical Reviews in Analytical Chemistry 36:243

78. Forina M, Lanteri S, Casale M (2007) Multivariate calibration. Journal of Chromatography A 1158:61

79. Massart D.L, Vandeginste B.G.M, Deming S.N, Michotte Y, Kaufman L (1988) Chemometrics: A textbook, Data Handling in Science and Technology, Elsevier, Amsterdam.

80. Danzer K, Otto M, Currie L (2004) Guidelines for calibration in analytical chemistry. Part 2: Multispecies calibration (IUPAC Technical Report). Pure and Applied Chemistry 76:1215

81. Coello J, Maspoch S (2007) In: Blanco M, Cerdà V (eds) Temas avanzados de quimiometría, Universitat de les Illes Balears. Servei de Publicacions i Intercanvi Científic, Palma

82. Kalivas JH, Gemperline P (2006) In Practical guide to chemometrics, 2nd edn. CRC/Taylor & Francis, Boca Raton, Florida.

83. Massart DL, Vandeginste BG, Buydens LM, Lewi PJ, Smeyers-Verbeke J, Jong SD (1998) Handbook of Chemometrics and Qualimetrics: Part A. Elsevier Science Inc, New York.

84. Escandar GM, Damiani PC, Goicoechea HC, Olivieri AC (2006) A review of multivariate calibration methods applied to biomedical analysis. Microchemical Journal 82:29

85. Olivieri AC, Faber NM, Ferré J, Boqué R, Kalivas JH, Mark H (2006) Uncertainty estimation and figures of merit for multivariate calibration (IUPAC Technical Report). Pure and Applied Chemistry 78:633

86. Boaz Nadler RRC, (2005) Partial least squares, Beer's law and the net analyte signal: statistical modeling and analysis. Journal of Chemometrics 19:45

87. Wold H (1975) Path models with latent variables: the NIPALS approach. In quantitative sociology: International Perspectives on Mathematical and Statistical Model Building. Academic Press, New York.

88. Wold S, Sjöström M, Eriksson L (2001) PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems 58:109

89. Brereton RG (2006) Consequences of sample size, variable selection, and model validation and optimisation, for predicting classification ability from analytical data. Trends in Analytical Chemistry 25:1103

90. Zeaiter M, Roger J, Bellon-Maurel V, Rutledge DN (2004) Robustness of models developed by multivariate calibration. Part I: The assessment of robustness. Trends in Analytical Chemistry 23:157

91. ISO (2008) International Vocabulary of Metrology – Basic and General Concepts and Associated Terms. JCGM 200:2008

92. De Bièvre P (1997) Measurement results without statements of reliability (uncertainty) should not be taken seriously. Accreditation and Quality Assurance 2:269

93. ISO-IEC (2005) General requirements for the competence of testing and calibration laboratories. ISO 17025.

94. Høy M, Steen K, Martens H (1998) Review of partial least squares regression prediction error in Unscrambler. Chemometrics and Intelligent Laboratory Systems 44:123

95. De Vries S, J.F. Ter Braak C (1995) Prediction error in partial least squares regression: a critique on the deviation used in The Unscrambler. Chemometrics and Intelligent Laboratory Systems 30:239

96. Faber K, Kowalski BR (1996) Prediction error in least squares regression: Further critique on the deviation used in The Unscrambler. Chemometrics and Intelligent Laboratory Systems 34:283

97. Rossel RAV (2007) Robust modelling of soil diffuse reflectance spectra by bagging-partial least squares regression. Journal of Near Infrared Spectroscopy 15:39

98. CAMO ASA (2004) The Unscrambler. 9.0

99. Faber NM (2000) Comparison of two recently proposed expressions for partial least squares regression prediction error. Chemometrics and Intelligent Laboratory Systems 52:123

100. Fernández Pierna JA, Jin L, Wahl F, Faber NM, Massart DL (2003) Estimation of partial least squares regression prediction uncertainty when the reference values carry a sizeable measurement error. Chemometrics and Intelligent Laboratory Systems 65:281

101. Efron B, Tibshirani R (1997) Improvements on cross-validation: The .632+ Bootstrap Method. Journal of the American Statistical Association 92:548

102. Faber NM (2002) Uncertainty estimation for multivariate regression coefficients. Chemometrics and Intelligent Laboratory Systems 64:169

103. Rossel RAV (2007) Robust modelling of soil diffuse reflectance spectra by bagging-partial least squares regression. Journal of Near Infrared Spectroscopy 15:39

104. Serneels V, Pierre J (2005) Bootstrap confidence intervals for trilinear partial least squares regression. Analytica Chimica Acta 544:153

105. Kiers (2004) Bootstrap confidence intervals for three-way methods. Journal of Chemometrics 18:22

106. Singh A (1996) Outliers and robust procedures in some chemometric applications. Chemometrics and Intelligent Laboratory Systems 33:75

107. Marques De Sa, Joaquim P. (2007) Applied Statistics Using SPSS, STATISTICA, MATLAB and R. Springer Berlin Heidelberg

108. Walczak B (1995) Outlier detection in multivariate calibration. Chemometrics and Intelligent Laboratory Systems 28:259

109. Singh A (1996) Outliers and robust procedures in some chemometric applications. Chemometrics and Intelligent Laboratory Systems 33:75

110. Jouan-Rimbaud D, Bouveresse E, Massart DL, de Noord OE (1999) Detection of prediction outliers and inliers in multivariate calibration. Analytica Chimica Acta 388:283

111. Barnard JP, Aldrich C (2000) In: Sauro Pierucci (ed), Detecting outliers in multivariate process data by using convex hulls, Computer Aided Chemical Engineering, European Symposium on Computer Aided Process Engineering-10, Elsevier, Amsterdam.

112. Fernández Pierna JA, Wahl F, de Noord OE, Massart DL (2002) Methods for outlier detection in prediction. Chemometrics and Intelligent Laboratory Systems 63:27

113. Daszykowski M, Kaczmarek K, Vander Heyden Y, Walczak B (2007) Robust statistics in data analysis — A review: Basic concepts. Chemometrics and Intelligent Laboratory Systems 85:203

114. Hsing T (1999) Nearest neighbor inverse regression. The Annals of Statistics 27:697

115. Chee Ng, Yunde Xiao, Wendy Putnam, Bert Lum,Alexander Tropsha, (2004) Quantitative structure-pharmacokinetic parameters relationships (QSPKR) analysis of antimicrobial agents in humans using simulated annealing k-nearest-neighbor and partial least-squares analysis methods. Journal of Pharmaceutical Sciences 93:2535

116. Yap CW, Li ZR, Chen YZ (2006) Quantitative structure–pharmacokinetic relationships for drug clearance by using statistical learning methods. Journal of Molecular Graphics and Modelling 24:383

117. Ter Braak CJF (1995) Non-linear methods for multivariate statistical calibration and their use in palaeoecology: a comparison of inverse (k-nearest neighbours, partial least squares and weighted averaging partial least squares and classical approaches. Chemometrics and Intelligent Laboratory Systems 28:165

118. Nigsch F, Bender A, van Buuren B, Tissen J, Nigsch E, Mitchell JBO (2006) Melting Point Prediction Employing k-Nearest Neighbor Algorithms and Genetic Parameter Optimization. Journal of Chemical Information and Modeling 46:2412

119. Itskowitz P, Tropsha A (2005) k nearest neighbors QSAR modeling as a variational problem: theory and applications. Journal of Chemical Information and Modeling 45:777

120. Oloff S, Mailman RB, Tropsha A (2005) Application of validated QSAR models of D1 dopaminergic antagonists for database mining. Journal of Medicinal Chemistry 48:7322

121. Tino P, Nabney IT, Williams BS, Losel J, Sun Y (2004) Nonlinear prediction of quantitative structure−activity relationships. Journal of Chemical Information and Modeling 44:1647

122. Burba F, Ferraty F, Vieu P (2009) *k*-Nearest Neighbour method in functional nonparametric regression. Journal of Nonparametric Statistics 21:453

123. Efron B, Tibshiran RJ (1993) An Introduction to the bootstrap. Chapman & Hall, New York

124. Hall P (1992) The Bootstrap and Edgeworth expansion. Springer-Verlag, New York.

125. LePage R, Billord L (1992) Exploring the limits of bootstrap. Wiley, New York

126. Mooney CZ, Duval RD (1993) Bootstrapping. A nonparametric approach to statistical inference. Sage, Newbury Park.

127. Wehrens R, Putter H, Buydens MC (2000) The bootstrap: a tutorial. Chemometrics and Intelligent Laboratory Systems 54:35

128. Davison AC, Hinkley DV (1997) Bootstrap methods and their application. Cambridge University Press, Cambridge

129. Chernick MR (1999) Bootstrap methods: a practitioner's guide. John Wiley and Sons, New York.

130. Hall P, Wilson SR (1991) Two Guidelines for bootstrap hypothesis yesting. Biometrics 47:757

131. Henderson AR (2005) The bootstrap: A technique for data-driven statistics. Using computer-intensive analyses to explore experimental data. Clinica Chimica Acta 359:1

132. Miller RG (1974) The jackknife-A review. Biometrika 61:1

133. Tukey JM (1958) Bias and confidence in not quite large samples (Abstracts of Papers). The Annals of Mathematical Statistics 29:614

134. Quenouille MH (1949) Approximate tests of correlation in time-series. Journal of the Royal Statistical Society.Series B (Methodological) 11:68

135. Quenouille MH (1956) Notes on bias in estimation. Biometrika 43:353

136. Wu CFJ (1986) Jackknife, bootstrap and other resampling methods in regression analysis. The Annals of Statistics 14:1261

137. Efron B (1981) Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. Biometrika 68:589

138. Efron B, Tibshirani R (1986) Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical Science 1:54

139. Stine R (1989) An Introduction to bootstrap methods: examples and ideas. Sociological Methods Research 18:243

140. Diciccio TJ, Romano JP (1988) A review of bootstrap confidence intervals. Journal of the Royal Statistical Society. Series B (Methodological) 50:338

141. DiCiccio T, Tibshirani R (1987) Bootstrap confidence Intervals and bootstrap approximations. Journal of the American Statistical Association 82:163

142. Carpenter J, Bithell J (2000) Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. Statistics in Medicine 19:1141

143. ILAC, UNIDO (2003) Laboratory accreditation in developing economies-working paper 2. 2

144. Thomas JD (1996) Reliability versus uncertainty for analytical measurement. Analyst 121:1519

145. Desimoni E, Brunetti B (2011) Uncertainty of measurement and conformity assessment: a review. Analytical and Bioanalytical Chemistry 400:1729

146. Morris AS (2004) ISO 14000 environmental management standards: engineering and financial aspects. Wiley, Hoboken, New Jersey

147. Martens H, Martens M (2001) Multivariate analysis of quality: an introduction. Wiley, Chichester.

148. Cordella LP, Foggia P, Sansone C, Tortorella F, Vento M (1999) Reliability parameters to improve combination strategies in multi-expert systems. Pattern Analysis & Applications 2:205

149. Ortiz MC, Sarabia LA, Sánchez MS, Herrero A (2009) In: Editors-in-Chief: Stephen D. Brown, Romà Tauler, Beata Walczak (eds) Comprehensive Chemometrics, Elsevier, Oxford

150. Baffi G, Martin E, Morris J (2002) Prediction intervals for non-linear projection to latent structures regression models. Chemometrics and Intelligent Laboratory Systems 61:151

CHAPTER 3

RELIABILITY OF k-NEAREST NEIGHBOURS IN CLASSIFICATION

3. Reliability of *k*-Nearest Neighbours in classification

3.1 Introduction

This chapter describes the Probabilistic Bootstrap *k*-Nearest Neighbours (PB*k*NN) method. PB*k*NN combines the *k*NN method and bootstrap, to compute the posterior probability of classification for each classified object. This posterior probability indicates the reliability of classification, i.e. the confidence with which a given object is classified in a certain class. The PB*k*NN method was evaluated with a simulated dataset and with two benchmark datasets: the Iris dataset and the Wine dataset. The results show that PB*k*NN provides reliability values that are comparable to those obtained with the Bayes' rule for the simulated dataset and to those obtained by the Linear Discriminant Analysis (LDA) method for the benchmark datasets.

PB*k*NN uses Hamamoto's bootstrap to obtain new bootstrap training samples. With Hamamoto's bootstrap the results of classification improve because, instead of building bootstrap training samples using the same objects of the original training set, it builds the bootstrap training samples as linear combinations of the objects in the original training set.

The reliability of classification is obtained as a posterior probability. This probability varies continuously (a continuous range of values can be obtained between 0 and 1) depending on the position of the test object in the multivariate space. This measure is more sensitive than in classical *k*NN, which yields the same probability for objects in a similar

position. This reliability value can also be used to derive a new classification rule, i.e., the object is classified in the class whose reliability is the highest.

3.2 Paper. Calculation of the probability of correct classification in probabilistic bagged *k*-nearest neighbours.

Chemometrics and Intelligent Laboratory Systems, Vol 94 No 1 (2008) 51-59.

Edited for format

Calculation of the probability of correct classification in probabilistic bagged *k*-nearest neighbours

Joe Luis Villa, Ricard Boqué*, Joan Ferré
Department of Analytical Chemistry and Organic Chemistry.
Rovira i Virgili University
C/ Marcel·lí-Domingo, s/n. 43007 Tarragona, Catalonia (Spain)

ABSTRACT

This paper presents a new method for computing the probability of correct classification for the *k*-Nearest Neighbours (*k*NN) method. The method uses bootstrap to provide the posterior probability which a new object is classified with. This is a measure of the reliability of the classification; it increases as the test object is closer to the training objects of a given class and is more sensitive to the position of the test object in the calibration space than the classical measure of posterior probability in *k*NN. This reliability of the classification is also used to derive a new rule for classification.

Keywords: classification; nearest neighbours; bootstrap; probability of classification; reliability.

*Corresponding author

E-mail addresses: ricard.boque@urv.cat

3.2.1 Introduction

Classification with the *k*-Nearest Neighbours (*k*NN) classifier [1] is popular because it can be implemented easily and has a good performance without requiring knowledge of the probability distribution function of the data. Chemical applications of *k*NN include the estimation of the quality of chromatograms [2], the classification of pyrolysis mass spectra [3], the determination of drug toxicity from NMR spectra [4] and the characterization of granular products [5] among others [6, 7].

The *k*NN classifier uses a training data matrix $\mathbf{X}$ of J variables measured on I objects. Each object is known to belong to a class c out of C possible classes. This classifier assigns a test object, with measured variables $\mathbf{x}_t = [x_1, x_2, \ldots, x_J]$, to the class to which most of the k nearest neighbours of this object belongs. Those neighbours are found according to a suitable metric, usually the Euclidean distance. There exist variations of the *k*NN method with different distances [8] and decision rule that are used for classification [9-14].

The probabilistic interpretation of *k*NN [11] is that the test object is assigned to the class for which this object has the highest posterior probability $P(class\ c|\mathbf{x}_t)$ which is the probability of the "true" class being c given the measure $\mathbf{x}_t$. Different expressions have been suggested to calculate the posterior probabilities [15] and the expected prediction error [16]. The most common is to calculate the posterior probability as k_c/k, where k_c is the number of nearest neighbours of class c among the k neighbours. A variation consists on combining *k*NN and bootstrap resampling [10-14]. This procedure, called *bagging*

(*Bootstrap AGGregatING*), is a type of ensemble method, which uses bootstrap to improve the performance of the classifier [13]. With bootstrap, many new datasets (called bootstrap training sets) are generated from the original training set. Then, for each bootstrap training set, the test object is classified using *k*NN. As a result of this process, B classification results for each object are obtained. The test object is finally assigned to the class where it was classified most of the times (majority vote).

Independently on the method that is used for assigning the class, it is desirable to have a measure of how certain we are that the assigned class is correct. A measure that is often used in classification is the classification error rate (CER) [11], which is the percentage of objects that are assigned to the wrong category. However, CER measures the performance of the classification rule in terms of *discriminability* i.e., how well the classification method classified test objects [17], and it is not an estimation of the *reliability*, which is a measure of the degree of confidence of the classification of a particular test object [8]. A useful measure of reliability is given by the posterior probability [17]. This probability, as we mentioned before, can be calculated using different approaches. One of them is k_c/k. This measure, however, has the inconvenience that will be the same for all the test objects that have the same k_c. These objects, however, can be in slightly different locations and be more or less close to their neighbours. Intuitively, the closer the test object is to their neighbours, the more reliable the classification is. This, however, is not translated into the value of k_c/k. Moreover, this measure only takes a few discrete values, i.e., for $k = 3$, it only takes values of 0, $1/3$, $2/3$ and 1. We would expect that the reliability would change continuously for similar positions of the test

object in the variable space. Another measure of reliability could tentatively be derived from the majority vote obtained during bootstrap (Bagged *k*NN [13, 15]), by counting how many times the object was assigned to a certain class with respect to total number of bootstraps. This measure of reliability, however, has the inconvenience that an object which has always been assigned to the same class in all the bootstrap iterations, will be assigned a value of reliability of 100%, independently of the number of neighbours of the other classes that were among the k neighbours in each iteration.

Here we present a procedure to estimate the reliability of classification of an object for *k*NN that varies depending on the position of the object in the variable space. The method is based on a modification of *bagged k*NN [13, 15], in which the posterior probability, $P_B(class\ c|\mathbf{x}_t)$, is calculated using bootstrap. Here the bootstrap method proposed by Hamamoto *et al.* [12] is used. While the classical bootstrap uses random sampling with replacement [18, 19], in the Hamamoto's method the bootstrap samples are created (not selected) by locally combining the original training samples, i.e. a new bootstrap sample of a given training object is created by weighing the values of that object and its nearest neighbours. The new bootstrap samples created from all the training objects are used to calculate the *k*NN posterior probability for a test object. This process is repeated B times (i.e., B bootstraps) and finally the mean of the posterior probabilities, $P_B(class\ c|\mathbf{x}_t)$, obtained for the test object is used as a measure of reliability. This reliability value also gives rise to a new classification rule which consists of assigning the test object to the class with the highest $P_B(class\ c|\mathbf{x}_t)$.

This article first describes the *k*NN and the bootstrap method underlying the PB*k*NN method. Next we show the implementation of PB*k*NN. The results of PB*k*NN will be compared with the results of probability obtained with Bayes' Rule [11] and Linear Discriminant Analysis (LDA) [11, 20-22]. The implementation of the algorithm will be illustrated with three datasets: a simulated dataset and two benchmark datasets: Iris dataset and Wine dataset [21, 22].

3.2.2 Methods

3.2.2.1 *k- Nearest Neighbours*

In *k*NN, the posterior probabilities of a given test object to belong to class c is given by [11]:

$$P(class\ c|\mathbf{x}_t) = \frac{k_c}{k} \qquad Eq.3.1$$

where k_c is the number of nearest neighbours of the test object in the training set that belong to class c, and k is the number of neighbours considered for classification. The decision rule for the probabilistic *k*NN is to classify a test object in the class where the posterior probability is the largest:

$$\begin{matrix}\text{test object}, \mathbf{x}_t, \in class\ c \\ \text{if } P(class\ c|\mathbf{x}_t)\end{matrix} = \max_c P(class\ c|\mathbf{x}_t) \qquad Eq.3.2$$

3.2.2.2 *Bootstrap*

Bootstrap is a resampling method [16-18, 23-26] that can be used to estimate a parameter θ of the distribution of a population from a

statistic $\hat{\theta}$ obtained from a sample of the population. In classification, bootstrap is used to improve the accuracy of a given method [11, 13].

There are several possible setups for bootstrap [12, 18, 19, 23]. The classical bootstrapping uses random sampling with replacement [18, 19]. This was already used with *k*NN but without satisfactory results due to the "stability" of the *k*NN [13]. *k*NN is "stable" because small changes in the training data do not lead to significantly different classification results. Here we will consider Hamamoto *et al.* bootstrap II method [12]. This type of bootstrap starts with the training data matrix, $\mathbf{X}$, in which I_c rows correspond to the objects of class c ($\mathbf{X}_c$). $\mathbf{X}_c$ is resampled and locally transformed (i.e. the new object is a combination of this object and its nearest neighbours), to generate a new bootstrap matrix of the class c, $\mathbf{X}_c^b$ where the superscript b indicates the bootstrap replicate.

Resampling is done as follows:

(1) Select the first object of $\mathbf{X}_c$, $\mathbf{x}_i(i = 1)$ and, using the Euclidean distance, find its R nearest neighbours $\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \dots, \mathbf{x}_{i,R}$ among those in $\mathbf{X}_c$.
(2) Compute a new bootstrap sample $\mathbf{x}_i^b$ as a weighted average of these neighbours, including the object i itself $(\mathbf{x}_{i,0})$:

$$\mathbf{x}_i^b = \sum_{r=0}^{R} w_0 \mathbf{x}_{i,0} + w_1 \mathbf{x}_{i,1} +, \dots, w_r \mathbf{x}_{i,r} \qquad Eq.\,3.3$$

where w_r is a weight defined as

$$w_r = \frac{\Delta_r}{\sum_{r=0}^{R} \Delta_r}, 0 \leq \Delta_r \leq 1, \qquad Eq.3.4$$

where Δ_r is a random value from a uniform distribution on [0,1] and $\sum_{r=0}^{R} w_r = 1$.

(3) Steps 1 and 2 are run for all the objects $i = 1, \ldots, I_c$ of $\mathbf{X}_c$, thus obtaining a new matrix $\mathbf{X}_c^b$ for class $c = 1$;

(4) Steps 1 to 3 are repeated for the other classes $c = 2, \ldots, C$.

(5) The bootstrap matrices $\mathbf{X}_c^b$ generated for all the classes are then adjoined to obtain the bootstrap training set $\mathbf{X}^b$ and $\mathbf{X}^b$ is used to classify the test object.

(6) Steps 1 to 5 are repeated B times and the results are finally combined.

Hamamoto's bootstrap II was chosen among the bootstrap methods available in Ref. [12], because in bootstrap II all the objects in the original training set participate in creating the bootstrap training set (each new bootstrap sample is created as a combination of itself and a few neighbours, $Eq.$ 3.3. In contrast, bootstrap methods I and III create new samples by randomly selecting objects of the original training set and hence some objects may never participate in the new bootstrap training set while others may appear more than once. With respect to bootstrap II and IV, note that bootstrap IV uses constant weights in the creation of the bootstrap samples, while bootstrap II uses random weights, which favours the diversity in the training step.

3.2.2.3 *Probabilistic bagged kNN*

The *probabilistic bagged* *k*NN (PB*k*NN) method proposed in this paper combines *k*NN and bootstrap. However, the classification criterion is different than for the bagged *k*NN cited in the Introduction. In PB*k*NN, B bootstrap training sets, $\mathbf{X}^b$ $(b = 1,2,\dots B)$ are generated with Hamamoto's bootstrap II described in section 3.2.2.2. For a given test object $\mathbf{x}_t$, its k nearest neighbours in each $\mathbf{X}^b$ are obtained. Of these k neighbours, k_c belong to class c, so that the posterior probability for the test object can be calculated as:

$$P_b(class\ c|\mathbf{x}_t) = \frac{k_c}{k} \qquad Eq.3.5$$

$Eq.$ 3.5 is calculated for the C classes. This procedure is repeated B times. Finally, the bootstrap posterior probability $P_B(class\ c|\mathbf{x}_t)$ that a test object belongs to class c is computed as:

$$P_B(class\ c|\mathbf{x}_t) = \frac{\sum_{b=1}^{B} P_b(class\ c|\mathbf{x}_t)}{B} \qquad Eq.3.6$$

This value of bootstrap posterior probability $P_B(class\ c|\mathbf{x}_t)$ is used as a new classification rule so that the test object $\mathbf{x}_t$ is finally classified in the class with the highest $P_B(class\ c|\mathbf{x}_t)$. This value is also taken as the reliability of the classification for this object [17].

3.2.2.4 *Bayes' decision rule*

We compared the posterior probabilities from PB*k*NN with those calculated using Bayes' decision rule. Bayesian decision is widely used

in pattern classification [11, 27]. It is based on assigning a test object to the class with the highest posterior probability. The posterior probability is computed as:

$$P(class\ c|\mathbf{x}_t) = \frac{p(\mathbf{x}_t|class\ c) \cdot P(class\ c)}{p(\mathbf{x}_t)} \qquad Eq.3.7$$

where $P(class\ c|\mathbf{x}_t)$ is the posterior probability that the real class is c given that the feature value $\mathbf{x}_t$ has been measured, $p(\mathbf{x}_t|class\ c)$ is a value of the class conditional probability density function, $P(class\ c)$ is the *prior* probability and $p(\mathbf{x}_t)$ is the evidence factor [11], which is computed as:

$$p(\mathbf{x}_t) = \sum_{c=1}^{C} p(\mathbf{x}_t|class\ c) \cdot P(class\ c) \qquad Eq.\ 3.8$$

It is to note that the Bayes' decision requires the probability density functions of the training data, $p(\mathbf{x}_t|class\ c)$ to be known. These will be known for the simulated dataset used in the experimental section, so the results of PB*k*NN and those of the Bayes' decision can be compared.

In real applications, however, the probability distributions may not be known. In that case, the Bayes' formula cannot be used to estimate the probabilities, but the PB*k*NN can still be applied, which is one of the advantages of the PB*k*NN in front of other methods.

3.2.3 Experimental section

3.2.3.1 *Data*

PB*k*NN was evaluated with a simulated dataset and the benchmark datasets Iris and Wine [22, 28-30]. The simulated dataset consists of a training matrix **X** of two classes, with twenty-five objects and two variables. The objects of class 1 were simulated from a bivariate normal distribution $\mathcal{N}(\mu_1, \Sigma_1)$ with mean $\mu_1 = [0,0]$ and covariance matrix $\Sigma_1 = \mathbf{I}$ where **I** is a 2×2 identity matrix. The objects of class 2 were also simulated from a bivariate normal distribution $\mathcal{N}(\mu_2, \Sigma_2)$ with $\mu_2 = [2,2]$ and $\Sigma_2 = \mathbf{I}$. As a result, both classes are slightly overlapped (Fig. 3.1). Additionally, one test object belonging to the class 1 and three test objects of class 2 were also simulated from the above distributions.

The Iris dataset [28, 30] contains measurements on three classes of Iris flowers (Setosa, Versicolor and Virginica) with fifty objects in each class and four variables (petal length, sepal length, petal width and sepal width). The dataset was divided using the Kennard and Stone's algorithm [31] into a training set of one hundred-five (thirty-five per class) objects, and a test of forty-five (fifteen per class) objects. The split was done by running Kennard and Stone's algorithm for each individual class and keeping the first 70 % of the objects of each class for the training set.

The Wine dataset [22, 29, 30] contains the results of chemical analysis of Italian wines of three different cultivars (Barolo, Grignolino, Barbera). This dataset has 27 variables, although only 13 variables are used here so that our results can be compared to those in the

references [20], [32] and [33]. These 13 variables are described for this dataset in UCI repository [30].

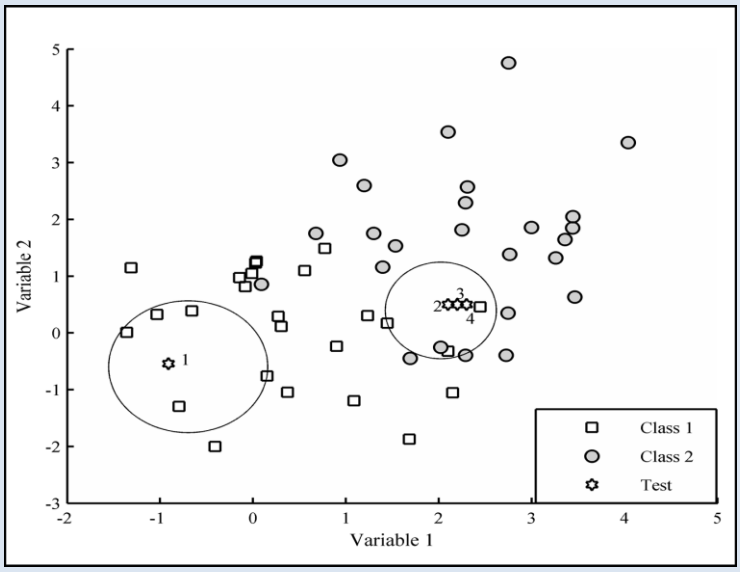

Figure 3.1 Simulated dataset. The circles show the Euclidean space for *k*=5 for test objects 1 and 2

3.2.3.2 *Selection of the optimal parameters for kNN and PBkNN.*

The selection of the optimal k value is critical in both *k*NN and PB*k*NN. In this paper the optimal k was selected as the one that yielded the minimal CER calculated with leave-one-out cross-validation. For *k*NN, cross-validation was done by classifying the object i of $\mathbf{X}$ against the remaining I–1 objects of $\mathbf{X}$ for a given k value. This procedure was repeated until all I objects were classified and, finally, the CER was computed. This procedure was repeated for different values of k. For PB*k*NN, in addition to the value of k, the bootstrap number and the r value that is needed to compute the bootstrap samples ($Eq.3.3$) must also be optimized. This was done by extracting an object i of $\mathbf{X}$ and using the remaining I–1 objects of $\mathbf{X}$ to generate a bootstrap matrix $\mathbf{X}^b$.

Then the object i was classified against $\mathbf{X}^b$ for given k and r values. This procedure was repeated B times for the object i and this object was finally classified into the class in which $P_B(class\ c|\mathbf{x}_t)$ was the highest. This bootstrap procedure was repeated until all I objects were classified and, finally, the CER was computed. This procedure was repeated for different k values. In order to estimate the minimum number of required bootstraps, B, cross-validation was run for different values of B, $B = \{100, 200, 300, 400, 500, 600, 700, 800, 900\ and\ 1000\}$.

3.2.3.3 *Reliability of classification*

The values of posterior probability obtained with PBkNN, $P_B(class\ c|\mathbf{x}_t)$ measure the *reliability* [27] (i.e., the confidence) of the classification of a particular test object. To validate the calculated values of reliability, two strategies were used. First, these values were compared with the Bayes formula for a simulated dataset, simulated from a bivariate normal distribution (section 3.2.2.4). The Bayes rule is considered the best classification rule [11], because it minimises the error rate of classification. The values of posterior probabilities obtained with PBkNN should behave similarly as the posterior probabilities obtained from Bayes. Second, the reliabilities from PBkNN were compared with the values of reliability from a recognized classification method, in this case Linear Discriminant Analysis (LDA). LDA has been used in many chemometrical applications [11, 20-22]. It is based on Bayes' rule and multinormality assumptions [20]. Reliability values of LDA were calculated with the program *class* (V-Class, version 00-01-2008) of the PARVUS [29] software. To compare the results of reliability, we introduce the root mean square of residual of reliability (R_r), which is:

$$R_r = \sqrt{\frac{\sum_{i=1}^{N}(P_{i1} - P_{i2})^2}{N}} \qquad Eq.\,3.9$$

where, P_{i1} and P_{i2} are the probabilities assigned for both methods (LDA, method 1 and PB*k*NN, method 2) to the true class and $i = 1,\ldots,N$ are the objects in the test set. R_r ranges between 0 (same probabilities obtained from both methods) to 1 (opposite results obtained from both methods). Small values of R_r indicate that the values of probability are similar for both methods.

The first strategy was applied with a simulated dataset for which we know the function of distribution for each class and, therefore, the probability values. The second strategy was applied to the Wine dataset. Classification accuracies of 98.9 % have been reported for this dataset using LDA [20], which indicates that the method models well the data. For this dataset, the results of PB*k*NN were also compared to reported results of other methods with respect to the classification accuracy [20, 32] and leave-one-out error rate [33].

3.2.4 Results and Discussion

Here the simulated and the Iris datasets were used in order to show the functionality of PB*k*NN. For both datasets, the values of k for *k*NN and PB*k*NN and B for PB*k*NN were selected by leave-one-out cross-validation. Moreover, for the Wine dataset, besides k and B, also the value of r ($Eq.$ 3.3) was selected for PB*k*NN by leave-one-out cross-validation.

3.2.4.1 *Simulated Dataset*

3.2.4.1.1 Selection of k value and bootstraps number

Classification error rate values were computed using cross-validation for values of k from 1 to 11, both for kNN and for PBkNN. For the even values of k, in case of tie, the sum of the distances of the object to the nearest neighbours of class 1 was calculated, as well as the sum of the distances of the objects to the nearest neighbours of class 2. The object was classified in the class for which the sum of the distances was the smallest. Fig. 3.2 shows that the minimal CER for kNN was obtained for $k = 3$ and $k = 4$. In this case $k = 3$ was selected as optimal because it is the lowest odd value of k. For PBkNN, the optimal value was $k = 5$. This value was decided after running $B = 400$ bootstraps. In order to show the dependency of the optimal k with the number of bootstraps, the CER for different values of k for an increasing number of bootstraps was calculated (Fig. 3.3). It is seen that the CER is stable when the number of bootstraps is larger than 400, so 400 was selected as the sufficient bootstrap number for this dataset. Also note that, for a given k, the variation of the CER for different number of bootstraps is much smaller than the variation of CER among different k values. This indicates that, in this case, although the number of bootstraps influences the classification results, its value is not as critical as the correct choice of k for obtaining a low CER.

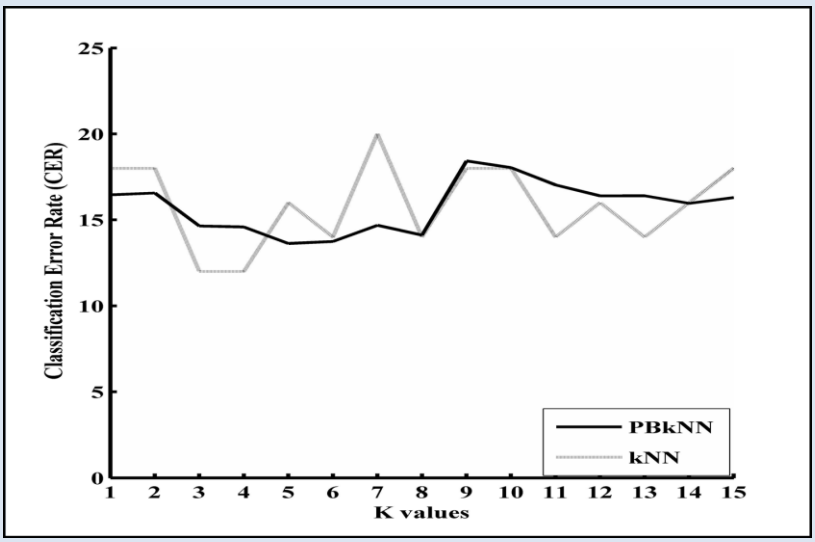

Figure 3.2 Classification Error Rate (CER) for different values of *k* for *k*NN and PB*k*NN (B=400) for the simulated dataset. CER was obtained by leave-one-out cross validation.

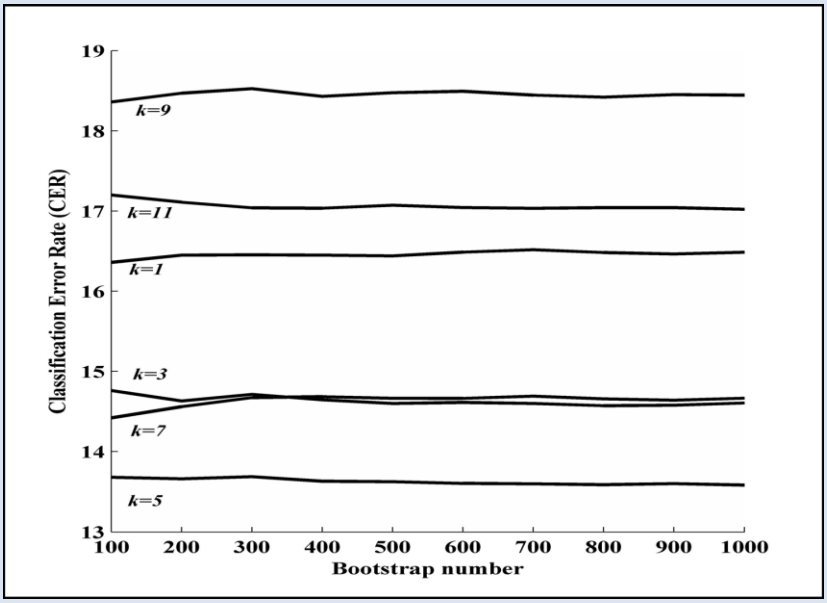

Figure 3.3 Classification Error Rate (CER) for different number of bootstrap, *B*, and different *k* values for PB*k*NN for the simulated data set.

3.2.4.1.2 Reliability of classification

Four test objects were simulated in order to illustrate the calculation of the classification reliability. Test object 1 is at one extreme of the region of class 1 and it was selected as an easy-to-classify object, since all the nearest neighbours belong to class 1. Test objects 2, 3 and 4 are located between the two classes, and hence in a complex position to be classified by kNN. For $k = 5$ (this value of k was used to compare kNN with PBkNN because 5 was the optimal k for PBkNN), objects 2 and 3 have three neighbours of class 1 and two neighbours of class 2, and object 4 has two neighbours of class 1 and three neighbours of class 2. Table 1 shows the classification results and the reliability of classification, measured as posterior probabilities, for PBkNN ($k = 5, B = 400$), kNN ($k = 3$ and $k = 5$), and the Bayes' decision.

The Bayes' decision rule correctly classified object 1 in class 1 and objects 2 to 4 in class 2. The probability values for these objects are plotted in figure 3.4, which also shows the isoprobability contours from $Eq.$ 3.7 for the bivariate distribution that generated the training data of class 1. The contour lines enclose the region of probability values from 0.0 to 1.0 with increments of 0.1. Object 1 has probability of belonging to class 1 of 0.99, while objects 2 to 4 have a probability of belonging to class 1 of 0.26, 0.23 and 0.20 respectively, i.e., they have a higher probability of belonging to class 2.

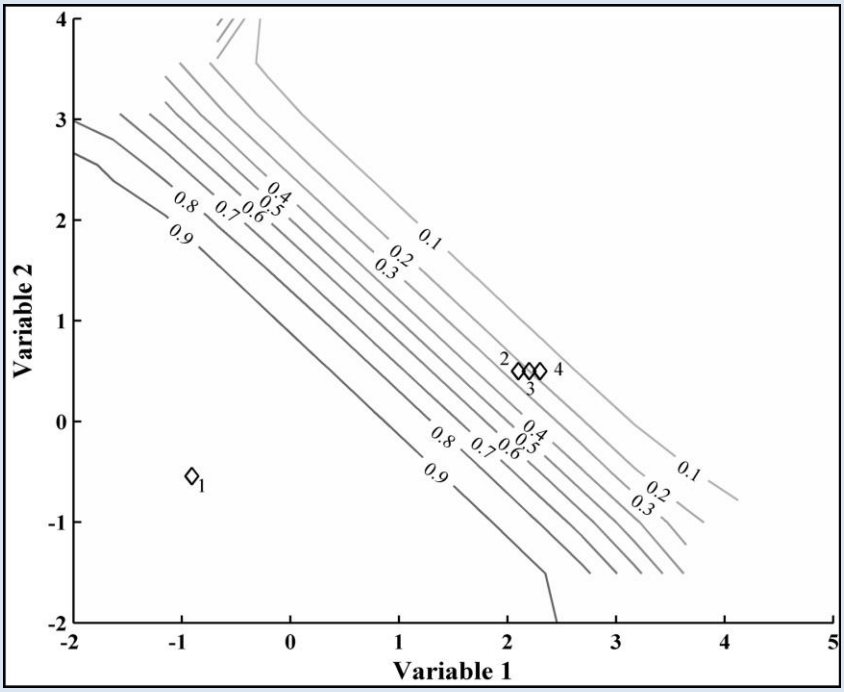

Figure 3.4 Isoprobability contour plot obtained with the Bayes' decision rule for class 1. Each probability value is in one region between the lines. (◊) indicates the test objects.

Table 3.1 Reliability of classification with the Bayes' decision, *k*NN and PB*k*NN for the simulated data set.

Test object	Bayes' decision		*k*-NN (*k*=3)		*k*-NN (*k*=5)		PB*k*NN (*k*=5, *B*=400)	
	Class 1	Class 2	Class 1	Class 2	Class 1	Class 2	Class 1	Class 2
1 (1)	0.99*	0.01	1.00*	0.00	1.00*	0.00	1.00*	0.00
2 (2)	0.26	0.74*	0.67*	0.33	0.60*	0.40	0.47	0.53*
3 (2)	0.23	0.77*	0.33	0.67*	0.60*	0.40	0.41	0.59*
4 (2)	0.20	0.80*	0.33	0.67*	0.40	0.60*	0.37	0.63*

True classes are indicated in brackets and the classes assigned by the methods are indicated with an asterisk.

Table 3.1 also shows the posterior probability values for kNN with $k = 3$ and $k = 5$ ($Eq.$ 3.5). Since the five closest objects to test object 1 belong to class 1, the assigned posterior probability is 1 both for $k = 3$ and for $k = 5$. The large agreement with the Bayesian probability is to be expected because this is an object well in the middle of class 1. This value of probability is also found by PBkNN, which indicates that the randomly generated bootstrap samples always resulted in five objects of class 1 around the test object 1. A more interesting behaviour, however, is observed for objects 2 to 4. Objects 3 and 4 have the same number of nearest neighbours of each class for $k = 3$ (one neighbour of class 1 and two neighbours of class 2). Hence, they are classified by kNN into class 2 and assigned the same posterior probability (0.67). This shows a limitation of the kNN method, because for $k = 3$ the posterior probability value is the same for these objects despite they are located in a slightly different position in the variable space. This behaviour is also observed for $k = 5$. With $k = 5$, however, objects 2 and 3 were classified in class 1 and object 4 was classified in class 2, because they have three neighbours of that class (hence k_c/k = 3/5 = 0.6). Both PBkNN and Bayes' decision assign a different value of probability to these two objects, as it is to be expected from their different position in the variable space. These probability values increase when the test objects are closer to the training objects of the given class and decrease otherwise. It is to note that, in real cases, the probability distribution from which the objects are taken is seldom known, and hence the Bayes' probability cannot be calculated. In that case, the PBkNN seems to provide a more reliable measure of the probability than the k_c/k formula used in kNN. Note, however, that the results for PBkNN, although follow the trends obtained by Bayes' decision, do not yield the same results. The reason is that the PBkNN

values are calculated by resampling the 25 objects available, while the Bayes' probability is calculated from the theoretical distribution that generated those objects. Table 3.1 also shows that *k*NN classified incorrectly the test object 2 in the class 1 since it has two neighbours of class 1 and one of class 2 for $k = 3$ and three neighbours of class 1 and two of class 2 for $k = 5$. Bayes and PB*k*NN, however, classified this object in the class 2. The different B bootstrap training sets generated by linear transformation of the original training set increased the variability around object 2, and its neighbours varied in each iteration.

Figure 3.5 shows this variability. The bar graph shows, for all the bootstrap training sets generated with PB*k*NN, the number of times that 0 to 5 neighbours (i.e., k_c) of object 2 belong either to class 1 or to class 2. Recall that, for the optimal value of $k = 5$, the object is classified into that class if k_c is equal to or larger than 3. That is to say, 164 times the object 2 had three or more neighbours of class 1 (and hence it would be classified into class 1), while 236 times the object 2 had three or more neighbours of class 2 (and hence it would classified into class 2). Hence, object 2 was finally classified into class 2, with a reliability of 0.53.

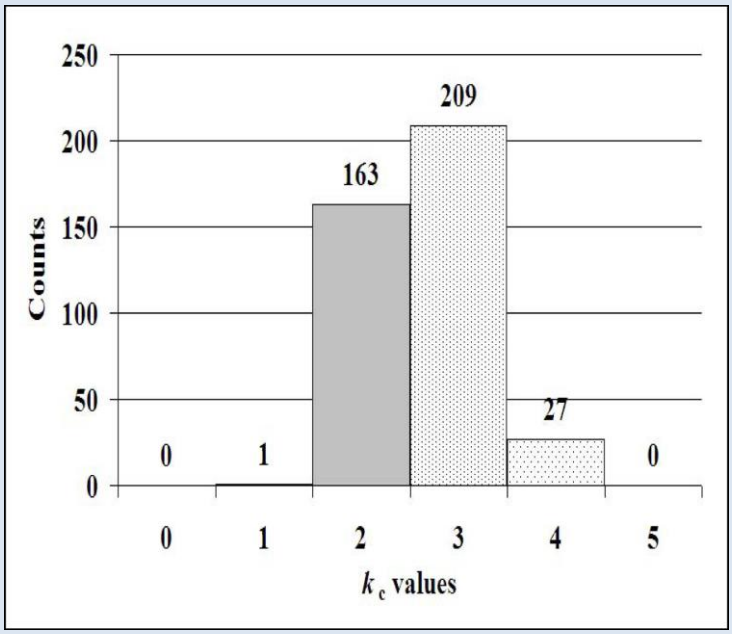

Figure 3.5 Bar graphs of k_c values of the test object 2 for class 2, obtaining during bootstrap in PB*k*NN. The dotted areas indicate that the object would be classified in that class.

3.2.4.2 *Iris Dataset*

Figure 3.6 shows the distribution of the training and test objects on the scores of Principal Component Analysis (PCA) calculated on mean-centered data. The objects of the class Setosa are separated from the other classes, while classes Versicolor and Virginica overlap. Hence, it is to be expected that the probability of correct classification will be higher for objects of class Setosa than for objects of the other two classes.

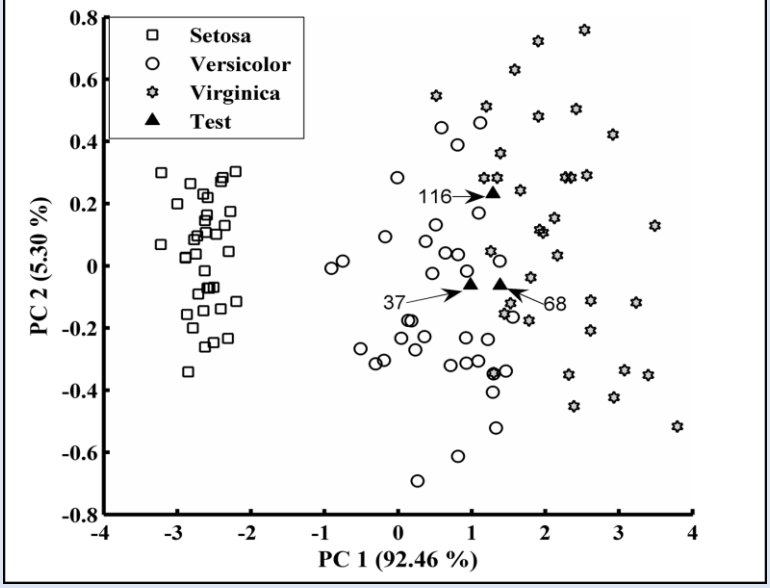

Figure 3.6 PCA of the iris dataset based on mean-centered data.

3.2.4.2.1 Selection of the *k*-value and bootstrap number

For *k*NN and PB*k*NN we evaluated the values of k from 1 to 25 (Fig. 3.7). The classification error rate (CER) was obtained by leave-one-out cross-validation. The optimal number of nearest neighbours was $k = 13$ for *k*NN and $k = 9$ for PB*k*NN. Furthermore, for PB*k*NN we selected the bootstrap number by generating up to 1000 bootstraps and computing the CER (Fig. 3.8). The CER values stabilized for more than 500 bootstraps, so we selected $B = 500$ as the optimal for classification.

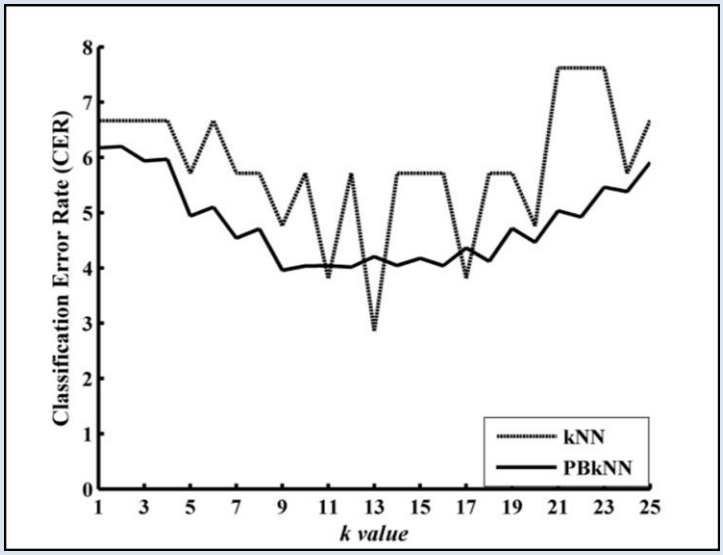

Figure 3.7 Selection of the *k* value for the Iris dataset for *k*NN and PB*k*NN.

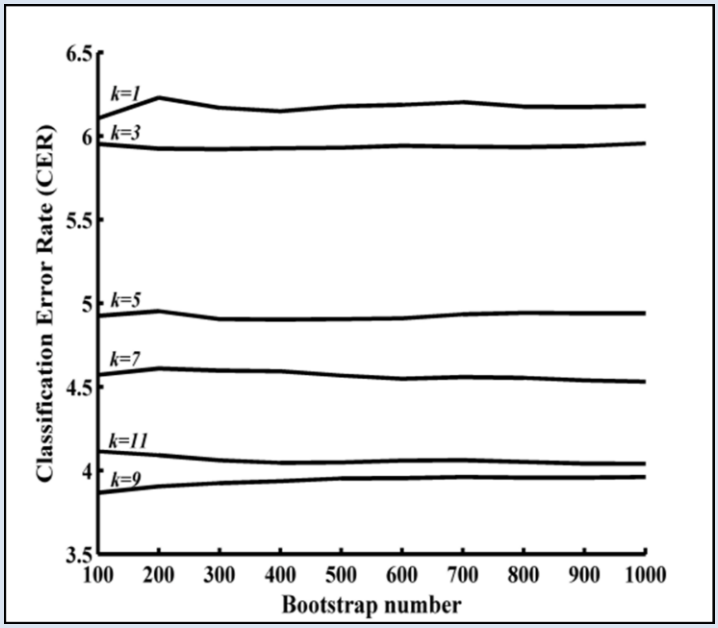

Figure 3.8 Selection of the number of bootstrap, *B*, for PB*k*NN the Iris dataset.

3.2.4.2.2 Reliability of classification

All objects in the test matrix were correctly classified by both *k*NN and PB*k*NN methods. Both methods also gave the same values of posterior probability of classification (1.0) for all test objects of the Setosa class, which is isolated from the other two classes. However, these methods gave different values of posterior probability for some objects in the overlap region of classes Versicolor and Virginica (Fig. 3.6). Table 3.2 shows the classification results for three test objects from that overlapped region: object 37 of class Versicolor, and objects 68 and 116 of class Virginica (the numbers correspond to the ordering of the objects in the original Iris dataset). *k*NN classifies objects 37, 68 and 116 with probability values of 0.77, 0.62 and 0.54 respectively. With PB*k*NN method, the posterior probabilities for these objects are 0.88, 0.66 and 0.75 respectively. These values are higher than for *k*NN, and suggest that the classification results are actually more reliable than what *k*NN suggests. Now we need to show that those higher values are more accurate than the lower values obtained by *k*NN. Since the true probability density functions are not available, a demonstration can be derived from the bar graphs of the numbers of neighbours obtained during bootstrap for a given object.

Figure 3.9 shows the frequency for the number of neighbours of object 116 obtained with PB*k*NN with $B = 500$ and $k = 9$ (the optimal value obtained for PB*k*NN) and also for $k = 13$ (the optimal value obtained for *k*NN). For PB*k*NN (Fig. 9a), object 116 was classified in class Virginica with a reliability of 0.75. This value is obtained from all the probabilities in which k_c is equal to or larger than 5, because the optimal value of k for PB*k*NN is $k = 9$. Of them, the most frequent

number of neighbours is $k_c = 7$ of class Virginica which are found 376 times out of 500. This value is the main contribution to the calculated posterior probability in $Eq.3.6$. On the other hand, kNN classified this object in class Virginica since 5 of the 9 neighbours were of class Virginica. Hence, it was classified with a posterior probability of $k_c/k = 5/9 = 0.56$.

Table 3.2. Reliability of classification with kNN and PBkNN for the Iris data set

Method	Class	Object 37 (Versicolor)	Object 68 (Virginica)	Object 116 (Virginica)
k-NN (k=9)	Setosa	0.00	0.00	0.00
	Versicolor	0.67*	0.44	0.44
	Virginica	0.33	0.56*	0.56*
k-NN (k=13)	Setosa	0.00	0.00	0.00
	Versicolor	0.77*	0.38	0.46
	Virginica	0.23	0.62*	0.54*
PBkNN (B = 500, k=9)	Setosa	0.00	0.00	0.00
	Versicolor	0.88*	0.34	0.25
	Virginica	0.12	0.66*	0.75*
PBkNN (B = 500, k=13)	Setosa	0.00	0.00	0.00
	Versicolor	0.83*	0.37	0.33
	Virginica	0.17	0.63*	0.67*

True classes are within brackets and the classes assigned by the methods are indicated with an asterisk

Figure 3.9a shows that $k_c = 5$ occurred only 5 times in the 500 bootstraps carried out. Hence $k_c = 5$ (and its associated probability) does not represent the underlying distribution of objects around the object Virginica 116, and that the probability value obtained by PBkNN is a more realistic measure. A similar conclusion is obtained if we compare the probabilities for $k = 13$, which in this case is the optimal value of k for kNN. For this object, kNN found 7 neighbours of class

Virginica, so the assigned posterior probability is k_c/k = 7/13 = 0.54. However, this value of $k_c = 7$ only occurred 18 times in the 500 bootstraps (Fig. 3.9b). The most frequent value of neighbours was $k_c = 9$, which appear 232 times in the 500 bootstrap. By combining the values of $k_c = 9$ to 13, the object 116 would be classified with posterior probability value of 0.67.

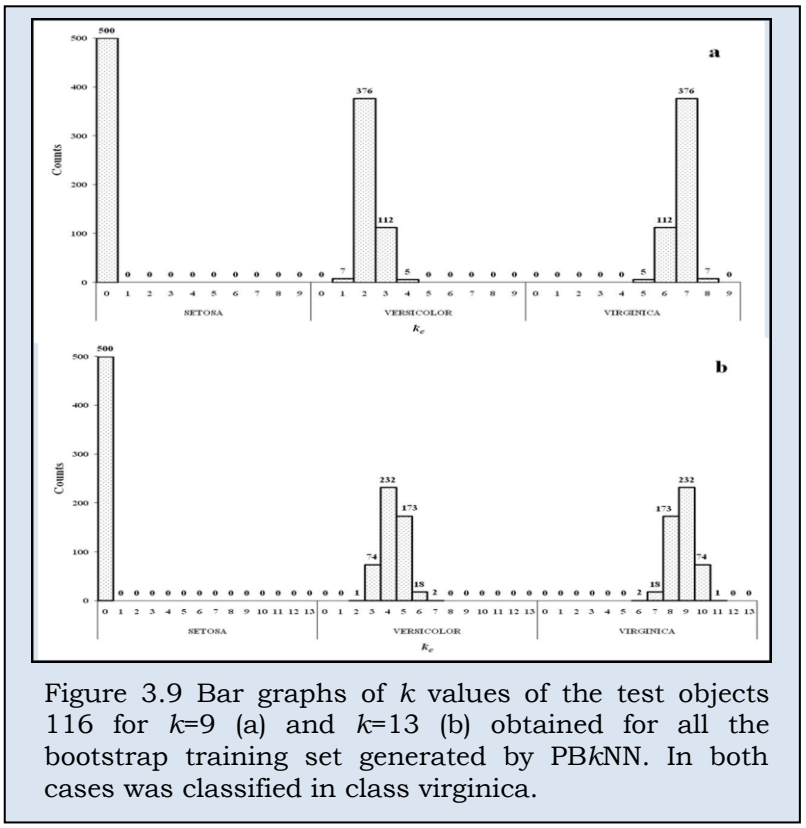

Figure 3.9 Bar graphs of k values of the test objects 116 for k=9 (a) and k=13 (b) obtained for all the bootstrap training set generated by PBkNN. In both cases was classified in class virginica.

Similar results were found for objects 37 and 68. In both cases the probabilities calculated with PBkNN (0.88 with $k = 9$ and 0.83 with $k = 13$) were higher than the probabilities for kNN (0.67 with $k = 9$ and 0.77 with $k = 13$).

3.2.4.3 *Wine Dataset*

3.2.4.3.1 Selection of the k-value, r-value and bootstrap number

For kNN and PBkNN we evaluated the values of k from 1 to 15 (Fig. 3.10). The CER was obtained by leave-one-out cross-validation. The optimal number of nearest neighbours was $k = 11$ for kNN and $k = 3$ for PBkNN. Furthermore, for PBkNN we selected the bootstrap number by generating up to 700 bootstraps and computing the CER.

Figure 3.11 shows the CER values for PBkNN with $k = 1$ and $k = 3$ with bootstrap number up to 700. The results for other values of k (not shown) are very similar to those for $k = 1$ and $k = 3$. CER depends on the number of objects wrongly classified, which, for this dataset, is similar for all values of k in all values of B, except for $k = 3$, which present minimal values of CER. Note that CER values stabilize for more than 200 bootstraps. Hence, the sufficient bootstrap number for this dataset was selected as 200. Since the results of CER were similar for different values of k and B, the selected k and B values were used to select the value of r. The value of r was selected by leave-one-out cross-validation by calculating the CER for r from 1 to 9. The optimal value of r, which gave the minimal CER, was found to be $r = 3$ (Fig. 3.12).

The results of PBkNN were compared with previously reported results [20, 32, 33] and with results from LDA. Both Alpaydin [32] and Viswanath *et al* [33] used the Wine dataset to check the performance of new classification methods based on kNN. They split the dataset into a training of 100 objects and a test set of 78 objects. The classification

accuracy (percentage of the objects correctly classified) reported for their methods is shown in Table 3.3 for completeness.

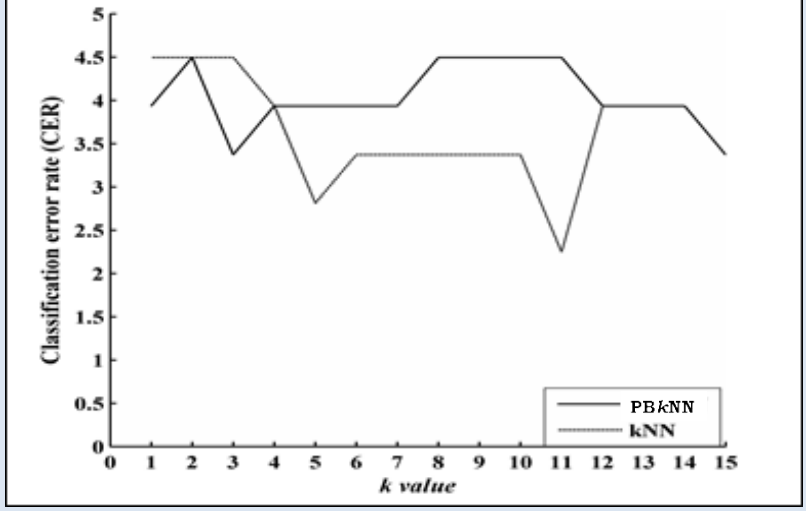

Figure 3.10 Selection of the *k* value for the Wine dataset

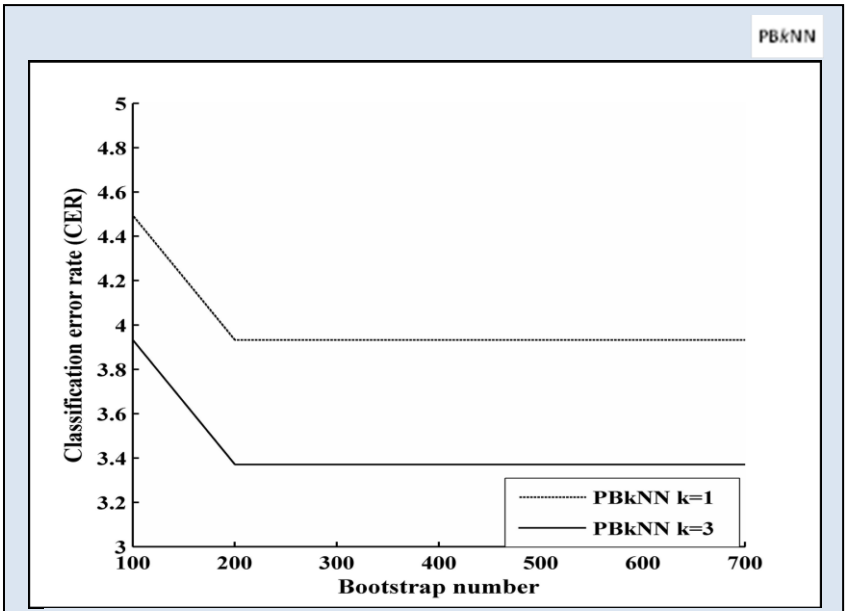

Figure 3.11 Selection of the number of bootstrap, *B*, for PB*k*NN for the Wine dataset.

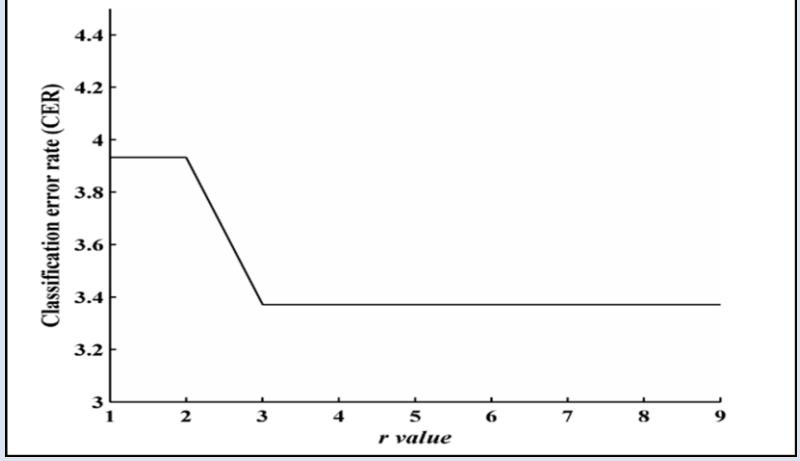

Figure 3.12 Selection of the number of nearest neighbours, *r*, for PB*k*NN for the Wine dataset.

Table 3.3. Comparative of classification accuracy

Reference	Method	Identification	Classification accuracy (%)
Proposed method	Probabilistic bootstrap *k*NN	PB*k*NN	95.96
Alpaydin [32]	1-Nearest Neighbours	NN	94.87
	Condensed Nearest Neighbours	CNN	93.21
	Voting over multiple CNN simple	Voting Simple	93.85
	Voting over multiple CNN weighted	Voting Weighted	95.00
	NN after multiple CNN	NN on union	93.97
Viswanath et al [33]	1-Nearest Neighbours	NNC	91.03
	k-Nearest Neighbours	*k*-NNC	92.31
	Naive Bayes Classifier	NBC	91.03
	NNC with Hamamoto's Bootstrap IV	NNC(BS)	93.29
	Overlap-based pattern -NNC	OLP-NNC	93.60

3.2.4.3.2 Comparison of PB*k*NN to other classification methods

These results must be compared to the classification accuracy from PB*k*NN. Since Alpaydin [32] and Viswanath *et al* [33] do not indicate what objects were assigned to the training set and which ones to the test set, we repeated the random split 100 times. The mean of the classification accuracy of the 100 PB*k*NN is shown in Table 3.3. It is seen that PB*k*NN has an accuracy of 95.96 % with a standard deviation of 1.94. This is comparable to the results reported by Alpaydin but seem slightly better than those reported by Viswanath *et al*. Notice, however, that both Alpaydin and Viswanath *et al*. report different results for the same 1-NN method, which suggests that the selection of the training and test sets influences the results to a certain degree.

The results from PB*k*NN were also compared to those of D-CAIMAN, M-CAIMAN, LDA, QDA, UNEQ, *k*NN, CART, SIMCA and NMC (Table 3.4) [20]. In this case, the reported result is the leave-one-out classification error rate. The PB*k*NN has an error rate of 3.4 %, which is lower than the values reported for *k*NN, CART, SIMCA and NMC. However, it was higher than D-CAIMAN, M-CAIMAN, LDA, QDA and UNEQ. Although the last methods present best error rates than PB*k*NN, these methods require the fulfilment of certain assumptions about the data structure.

For example, in D-CAIMAN and M-CAIMAN the number of class objects should be significantly greater than the number of variables (an objects/variable ratio greater than 2 or 3 is usually suggested). Moreover, LDA, QDA and UNEQ are based on Bayes' rule and/or they use the assumption of multinormality. PB*k*NN, on the other hand, does not require those assumptions.

Finally, the values of reliability obtained with PB*k*NN were compared with those obtained with LDA from PARVUS using the root mean square of residual of reliability (R_r). R_r measures the closeness of the reliability values of the two methods. In this case R_r of PB*k*NN with respect to LDA was 0.14, which indicates that the probabilities are fairly similar in both methods. This value, however, is highly increased by six objects (out of the 178 objects) which were misclassified by PB*k*NN. These objects had very different probabilities in LDA and in PB*k*NN. By removing these six objects from the calculation of R_r, the value decreases to 0.09, which suggests a large agreement between the predicted probabilities obtained with PB*k*NN and those from LDA.

Table 3.4. Comparative of Leave-one-out Error Rate

Reference	Method	Identification	Leave-one-out Error Rate (%)
Proposed method	Probabilistic bootstrap *k*NN	PB*k*NN	3.4
Todeschini *et al* [20]	Discriminant Classification And Influence Matrix Analysis	D-CAIMAN	1.1
	Modelling Classification And Influence Matrix Analysis	M-CAIMAN	1.1
	Linear Discriminant Analysis	LDA	1.1
	Quadratic Discriminant Analysis	QDA	0.6
	Modelling version of QDA	UNEQ	1.7
	k-Nearest Neighbours	KNN	23
	Classification and Regression Trees	CART	11.2
	Soft Independent Model Class Analogy	SIMCA	5.6
	Nearest Mean Classifier	NMC	27.5

3.2.5 Conclusions

We have proposed the *probabilistic bagged k*NN (PB*k*NN) method, which combines *k*NN and bootstrap. The new method provides the reliability of the classification for a particular object. This reliability is obtained as a posterior probability which is calculated by bootstrap. This probability varies smoothly (a continuous range of values can be obtained between 0 and 1) depending on the position of the test object in the multivariate space. This measure is more sensitive than in *k*NN, which might yield the same probability for objects in a similar position. This reliability value can also be used to derive a new classification rule, i.e., the object is classified in the class in which the reliability is the highest. For simulated data, PB*k*NN produced results that vary like the Bayes' decision results do. For the Iris dataset, PB*k*NN classified all the objects as well as *k*NN did, but PB*k*NN yielded higher reliabilities than *k*NN for objects that were located in the overlap region between two classes. Those higher reliabilities were shown to be a more accurate estimation of the actual situation, and hence, for that dataset, *k*NN tended to underestimate the reliability of such classifications.
The reliability values obtained with PB*k*NN are similar that the values obtained with a standard method (LDA). These similarities assess the values of reliability obtained with PB*k*NN respect to LDA.

Acknowledgements

The authors thank support of Department of Universities, Research and Information Society of Catalonia - Spain, for providing Joe Luis Villa's doctoral fellowship and project CTQ2007-66918 of the Spanish Ministry of Education and Science.

3.2.6 References

1. Cover T, Hart PE (1967) Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13:21

2. Pirogov A, Platanov M, Pletnev IV, Shpigun OA (1998) Application of the pattern-recognition method for modelling expert estimation of chromatogram quality. Analytica Chimica Acta 369:47

3. Alsberg BK, Goodacre R, Rowland JJ, Kell DB (1997) Classification of pyrolysis mass spectra by fuzzy multivariate rule induction-comparison with regression, K-nearest neighbour, neural and decision-tree methods. Analytica Chimica Acta 348:389

4. Beckonert O, Bollar ME, Ebbels T, Keun H, Antti H, Holmes E, Lindon J, Nicholson J (2003) NMR-based metabonomic toxicity classification: hierarchical cluster analysis and k-nearest-neighbour approaches. Analytica Chimica Acta 490:3

5. Ros F, Guillaumet S, Rabatel F, Sevilla F, Bertrand D (1997) Combining global and individual image features to characterize granular product populations. Journal of Chemometrics 11:483

6. Todeschini R (1990) Weighted *k*-Nearest Neighbor method for the calculation of missing values. Chemometrics and Intelligent Laboratory Systems 9:201

7. Lukasiak BM, Faria R, Zomer S, Brereton RG, Duncan JC (2006) Pattern recognition for the analysis of polymeric materials. Analyst 131:73

8. Wu W, Massart DL (1997) Regularised nearest neighbour classification method for pattern recognition of near infrared spectra. Analytica Chimica Acta 349:253

9. Parthasarathy G, Chatterji BN (1990) A class of new *k*NN methods for low sample problems. IEEE transactions on systems, man, and cybernetics 20:715

10. Holmes CC, Adams NM (2002) A probabilistic nearest neighbour method for statistical pattern recognition. Journal of the Royal Statistical Society, Series B 64:295

11. Duda RO, Hart PE, Stork DG (2001) Pattern classification. John Wiley & Sons, New York.

12. Hamamoto Y, Uchimura S, Tomita S (1997) A bootstrap technique for nearest neighbor classifier design. IEEE Transactions on Pattern Analysis and Machine Intelligence 19:73

13. Breiman L (1996) Bagging predictors. Machine Learning 24:123

14. Peter Hall RJS, (2005) Properties of bagged nearest neighbour classifiers. Journal of the Royal Statistical Society Series B 67:363

15. Caprile B, Merler S, Furlanello C, Jurman G (2004) Exact bagging with k-nearest neighbour classifiers. Lecture Notes in Computer Science 3077:72

16. Steele B, Patterson D (2000) Ideal bootstrap estimation of expected prediction error for k-nearest neighbor classifiers: Applications for classification and error assessment. Statistics and computing 10:349

17. Gurov SI (2004) Reliability estimation of classification algorithms. I. Introduction to the problem. Point frequency estimates. Computational Mathematics and Modeling 15:365

18. Efron B (1979) Bootstrap Method- Another look at the Jacknife. The Annals of Statistics 7:1

19. Efron B, Tibshiran RJ (1993) An introduction to the bootstrap. Chapman & Hall, New York

20. Todeschini R, Ballabio D, Consonni V, Mauri A, Pavan M (2007) CAIMAN (Classification And Influence Matrix Analysis): A new approach to the classification based on leverage-scaled functions. Chemometrics and Intelligent Laboratory Systems 87:3

21. Todeschini R, Marengo E (1992) Linear discriminant classification tree: A user-driven multicriteria classification method. Chemometrics and Intelligent Laboratory Systems 16:25

22. Forina M, Armanino C, Castino M, Ubigli M (1986) Multivariate data analysis as a discriminating method of the origin of wines. Vitis 25:189

23. Wehrens R, Putter H, Buydens MC (2000) The bootstrap: a tutorial. Chemometrics and Intelligent Laboratory Systems 54:35

24. Hinkley D (1988) Bootstrap methods. Journal of the Royal Statistical Society Series B 50:321

25. Molinaro A, Simon R, Pfeiffer R (2005) Prediction error estimation: a comparison of resampling methods. Bioinformatics 21:3301

26. Lei S, Smith MR (2003) Evaluation of several nonparametric bootstrap methods to estimate confidence intervals for software metrics. IEEE Transactions on Software Engineering 29:996

27. Webb AR (2002) Statistical pattern recognition. John Wiley and Sons Ltd, New York

28. Fisher RA (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics 7:179

29. PARVUS Official Web Site (2008). http://www.parvus.unige.it. Accessed 10-03 2008

30. Asuncion A, Newman DJ (2007) UCI Machine Learning Repository.

31. Kennard R, W., Stone LA (1969) Computer aided design of experiments. Technometrics 11:137

32. Alpaydin E (1997) Voting over multiple condensed nearest neighbors. Artificial Intelligence Review 11:115

33. Viswanath P, Murty N, Bhatnagar S (2005) Overlap pattern synthesis with an efficient nearest neighbor classifier. Pattern Recognition 38:1187

CHAPTER 4

INFLUENCE OF THE MEASUREMENT ERROR ON THE RELIABILITY OF CLASSIFICATION WITH kNN

4. Influence of the measurement error on the reliability of classification with *k*NN

4.1 Introduction

This chapter describes the Bagged *k*-Nearest Neighbours (Bagged-*k*NN) method. Bagged-*k*NN, like PB*k*NN, combines *k*NN and bootstrap to obtain the reliability of classification of a given object, which is calculated as a posterior probability. However, Bagged-*k*NN takes into account the values of uncertainty of the *x*-data to carry out the classification. For this, a new bootstrap method was developed. This new method, called *U*-bootstrap, uses the values of the uncertainty of the *x*-data to generate a new bootstrap dataset with which a given object is classified. The method was evaluated using a simulated dataset and the Wine dataset. The results show that the values of the uncertainty in the *x*-data influence the values of the reliability of classification and this variation is taken into account by the method. In this sense, the reliability, calculated as a posterior probability, varies continuously between 0 and 1, depending on the position of the object to be classified in the multivariate space.

This reliability value is also affected by the value of the uncertainty in the *x*-variables. Hence, the uncertainty in the measured variables should be taken into account in the classification methods.

4.2 Paper. Bagged *k*-nearest neighbours with uncertainty in the variables.

Analytica Chimica Acta, Vol 646 No 1-2 (2009) 62-68

Edited for format

Bagged *k*-nearest neighbours classification with uncertainty in the variables

Joe Luis Villa, Ricard Boqué[a], Joan Ferré
Department of Analytical Chemistry and Organic Chemistry.
Rovira i Virgili University C/ Marcel·lí Domingo, s/n. 43007
Tarragona, Catalonia (Spain)

ABSTRACT

An analytical result should be expressed as $x \pm U$, where x is the experimental result obtained for a given variable and U is its uncertainty. This uncertainty is rarely taken into account in supervised classification. In this paper, we propose to include the information about the uncertainty of the experimental results to compute the reliability of classification. The method combines *k*-nearest neighbours (*k*NN) with a nested bootstrap scheme, in which a new bootstrap training set is generated using the classical bootstrap in the first level (B times) and a new bootstrap method, called *U*-bootstrap, in the second level (D times). Two bootstraps are used to reduce the effect of sampling in the first level and the effect of the uncertainty in the second one. These $B \times D$ new training bootstrap sets are used to compute the reliability of classification for an unknown object using *k*NN. The object is classified into the class with the highest reliability. In this method, unlike the classical *k*NN and Probabilistic Bagged *k*-Nearest Neighbours (PB*k*NN), the reliability of classification changes (increases or decreases) when the uncertainty is increased. These changes depend on the position of the unknown object with respect to the training objects. For the benchmark wine dataset, we found similar values of classification error rate (CER) than for *k*NN (5.57%), but lower than Probabilistic Bagged *k*-Nearest Neighbours using Hamamoto's bootstrap (7.96%) or Efron's bootstrap (8.97%).

4.2.1 Introduction

Multivariate classification assigns an unknown object to class c among the C possible classes based on the values of the J variables $\mathbf{x}_i = [x_1, x_2, \dots, x_j]$ measured for that object. The classification rule is derived from a training set **X,** where $\mathbf{x}_i$ is measured in I objects of known class.

Commonly, the uncertainty of the values in $\mathbf{x}_i$ is not taken into account in the classification step. The uncertainty is defined as a parameter, associated with the result of a measurement that characterizes the dispersion of the values that could reasonably be attributed to the measurand [1]. The uncertainty is an important parameter in analytical science [2], so much that some authors consider that "*a result without reliability (uncertainty) statement cannot be published or communicated because it is not (yet) a result*" [3]. This means that a result should be expressed as $x \pm U$, where x is the value obtained for a given variable and U its uncertainty.

The uncertainty, particularly the expanded uncertainty U, defines an interval around the measured value in which the real value of the variable should be found. For this reason it is important to consider its influence in the classification and in the calculation of reliability of classification results. The reliability, which measures the degree of confidence of the classification of an object [4], is also an important measure of performance of the classification method [5]. It is desirable that the reliability of classification changes (increase or decrease) when the uncertainty of the variables in the training and test dataset changes. This change in the reliability should also depend on the

relative position of the unknown object with respect to the objects in the training set.

The objects in a given dataset have two main sources of uncertainty, sampling and measurement error. The first source is due to the fact that sampling is discrete, i.e. the objects in **X** do not cover the full space of the objects. The uncertainty in this case depends on the number of objects in **X**. The second source is the measurement error of the variables that characterize the objects. Its influence in the uncertainty depends on the size of the measurement error. With the aim of taking into account the effect of these sources of uncertainty during classification, a new method is proposed which combines *k*-nearest neighbours (*k*NN) with a nested bootstrap scheme. Common bootstrap methods only use **X** to generate the new bootstrap samples. However, by only bootstrapping the objects, the measurement noise is fixed during the re-sampling, so it cannot contribute to the estimated uncertainty. The proposed method considers the uncertainty in **X** both in the classification step and in the calculation of the reliability of classification. By nested bootstrap we mean that the classical Efron's bootstrap [6, 7] is used as a first level and an uncertainty bootstrap (*U*-bootstrap) is done for each bootstrap sample obtained using Efron. With Efron's bootstrap B new bootstrap data matrices are created by random selection with replacement of the objects in the original training set. Then, each bootstrap data matrix is used to create D new bootstrap training sets with *U*-bootstrap, by selecting, for each x_{ij}, a random value in the interval $[x_{ij} - U_{ij}, x_{ij} + U_{ij}]$ following a uniform distribution. These D matrices are finally used to classify the unknown object with *k*NN and to compute its reliability of classification. The uniform distribution is used instead of, for example, a normal

distribution, since we do not have a preference for the location of the true value inside the interval. In the usual *k*NN method [8, 9], reliability is computed as k_c/k, where k_c is the number of neighbours of the object in the class c and an object is assigned to the class with the highest reliability. Using the new bootstrap scheme, the reliability is computed after $B \times D$ bootstraps as the mean of the reliability values obtained in each bootstrap iteration, and the object is finally classified into the class with the highest combined reliability. Also, with a nested bootstrap scheme, both the contribution of sampling and measurement error to the final probability can be evaluated. The method was tested with a simulated dataset with two overlapped classes and with increasing uncertainty in the variables. Results of classification reliability were compared with the Bayes' decision rule [9]. Finally, the new methodology was applied to the benchmark Wine dataset, in order to classify the different wines in three regions of origin. The classification results and reliabilities were compared to the classical *k*NN method [8, 9] and to Probabilistic bagged *k*- Nearest Neighbours [10].

4.2.2 Methods

4.2.2.1 *k-Nearest Neighbours*

The *k*NN classifier uses a training data matrix **X**, where each object is known to belong to a class c out of C possible classes [9-12]. This classifier assigns an unknown object $\mathbf{x}_t$, to the class to which most of the *k*-nearest neighbours belong. These neighbours are found according to a suitable metric, usually the Euclidean distance. There are several variations of the *k*NN method, depending on the type of distance used [9, 11] or the decision rule that is used for classification

[9-12] For *k*NN, the posterior probability that a given unknown object belongs to class c is given by[9].

$$P(class\ c\ |\mathbf{x_t}) = \frac{k_c}{k} \qquad Eq.\ 4.1$$

where k_c is the number of nearest neighbours of the unknown object in the training set that belong to class c, and k is the number of neighbours considered for classification. The decision rule for probabilistic *k*NN is to classify an unknown object in the class where the posterior probability is the largest:

$$\begin{matrix}\text{unknown object, } \mathbf{x}_t \in class\ c \\ \text{if } P(class\ c\ |\mathbf{x_t})\end{matrix} = \max_{c\in\{1,\dots,C\}} P(class\ c\ |\mathbf{x_t}) \qquad Eq.\ 4.2$$

4.2.2.2 *Bagging*

Bagging (Bootstrap AGGregatING) is a type of ensemble method which uses bootstrap to improve the performance of the classifier [7, 9-12]. The improvement is obtained because bootstrap combined with a classification method leads to a reduction of the misclassification error [13]. Bootstrap is a resampling method [14-16] that can be used to estimate a parameter θ of the distribution of a population from a statistic $\hat{\theta}$ obtained from a sample of the population. For that, many (*B*) new datasets (called bootstrap samples) are created from the original dataset. These new datasets are used to estimate the statistic of interest. There are several possible setups for bootstrap: resampling with replacement of the objects in the original dataset [6, 7], Hamamoto *et al.* bootstrap [12] and parametric bootstrap [14] among others [15-19]

4.2.2.3 *Bagged k-Nearest Neighbours*

Nested bootstrap is based on the principle of resampling from bootstrap samples [14-16]. It involves two or more levels of bootstrap.

In this paper we use Efron's bootstrap [6, 7] and a newly developed bootstrap method, called Uncertainty bootstrap (*U*-bootstrap). *U*-bootstrap uses the uncertainty in the measured *x*-values to generate the new datasets. For this propose a training matrix $\mathbf{X}(I \times J)$ and an uncertainty matrix $\mathbf{U}(I \times J)$ are needed. **U** contains the uncertainty limits for each variable in each object. The method proceeds as follows:

(1) *U*-bootstrap selects the value of the first variable for the first object of **X**, $x_{ij}(i = 1;\ j = 1)$ and its uncertainty value, U_{ij}.

(2) A new value of x_{ij} is generated by random selection of a value in the interval $[x_{ij} - U_{ij}: x_{ij} + U_{ij}]$. This new value has a number of significant figures consistent with U_{ij}.

(3) The bootstrap matrix $\mathbf{X}^d$ is generated by repeating steps 1 and 2 for all objects and variables in **X**.

Efron's bootstrap does resampling with replacement of **X**, which is useful to limit the effect of sampling on the classification method. Both Efron's bootstrap and *U*-bootstrap are combined with *k*NN to develop the Bagged *k*-Nearest Neighbours (Bagged-*k*NN) method, which is summarized in Figure 4.1 and it is performed as follows:

(1) Select with replacement I_c objects of $\mathbf{X}_c$, where $\mathbf{X}_c$ is the training dataset for class $c = 1$, to obtain a new matrix $\mathbf{X}_c^b$ where b indicates the bootstrap iteration.

(2) Step 1 is repeated for the other classes, $c = 2, \ldots, C$.

(3) The *bootstrap* matrices $\mathbf{X}_c^b$ generated for all the classes are then adjoined to obtain the *bootstrap* matrix $\mathbf{X}^b$.

(4) For matrix $\mathbf{X}^b$, apply *U*-bootstrap *D* times, thus obtaining *D* nested bootstrap matrices (since they are generated from the same $\mathbf{X}^b$), $\mathbf{X}^{bd}$ (d = 1…..D).

(5) A given test object is also submitted to *U*-bootstrap, and, together with $\mathbf{X}^{bd}$ is used to compute the posterior probability, $P_d(\,class\ c|\mathbf{x}_t)$, with *k*NN using $Eq.\,4.1$.

(6) Step 5 is repeated *D* times and the posterior probability, $P_D(\,class\ c|\mathbf{x}_t)$,, is computed as:

$$P_D(\,class\ c|\mathbf{x}_t) = \sum_{d=1}^{D} P_d(\,class\ c|\mathbf{x}_t) \Big/ D \quad Eq.\ 4.3$$

(7) Steps (1) to (6) are repeated *B* times.

(8) The posterior probability, P_e is computed for all classes, as:

$$P_e(\,class\ c|\mathbf{x}_t) = \sum_{b=1}^{B} P_D(\,class\ c|\mathbf{x}_t) \Big/ B \quad Eq.\,4.4$$

(9) Finally the test object is assigned to the class with the highest $P_e(\,class\ c|\mathbf{x}_t)$.

2.4 Probabilistic bagged *k*NN (PB*k*NN)

The new Bagged *k*-Nearest Neighbours is compared here to PB*k*NN [10], which combines *k*NN and bootstrap, without taking into account the uncertainty in the **X**. In PB*k*NN, for a given unknown object $\mathbf{x}_t$, its k nearest neighbours in each $\mathbf{X}^b$ are obtained. Next, the posterior probability for the test object is calculated with $Eq.$ 4.1 for all C classes. This procedure is repeated B times. Finally, the bootstrap posterior probability $P_B(\,class\ c|\mathbf{x}_t)$ that a test object belongs to class c is computed as:

$$P_B(\,class\ c|\mathbf{x}_t) = \sum_{b=1}^{B} P_b(\,class\ c|\mathbf{x}_t) \Big/ B \qquad Eq.\ 4.5$$

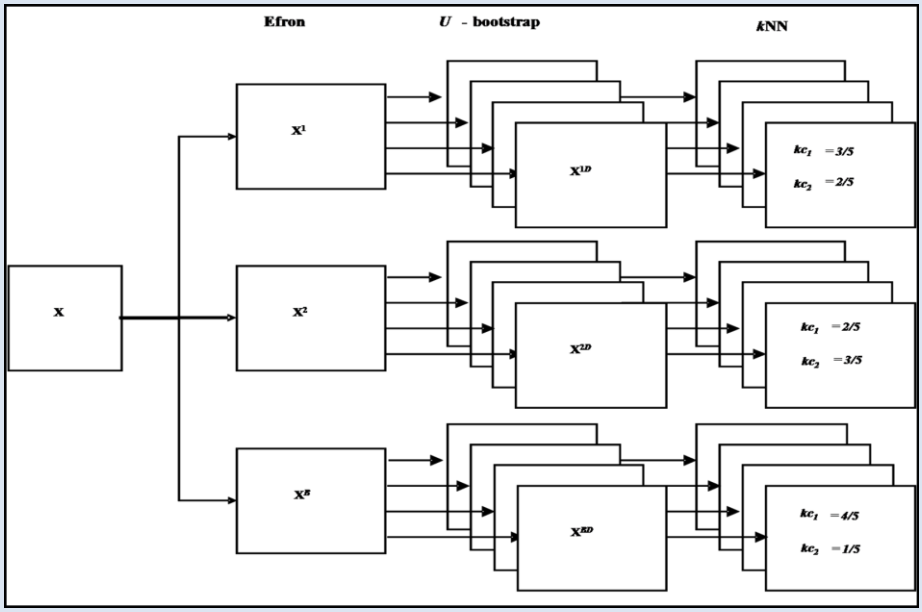

Figure 4.1 Algorithm Bagged-*k*NN. First, *B* Efron's bootstrap data matrices are generated from the original training dataset X. Next, for each bootstrap data matrix, *D* new matrices are generated with *U*-bootstrap. These nested bootstrap matrices are used to calculate the reliability of classification of a given test object using *k*NN, and classify it. In this example *k*=5

The object $\mathbf{x}_t$ is finally classified in the class with the highest $P_B(\,class\,c|\mathbf{x}_t)$. This value is also taken as the reliability of the classification of this object. In the original PBkNN method, B bootstrap training sets $\mathbf{X}^b\,(b = 1,2,\ldots B)$ were generated with Hamamoto's bootstrap II [12]. However, in order to make the results comparable with those of Bagged-kNN, the PBkNN was modified by using Efron's bootstrap instead of Hamamoto's bootstrap.

4.2.3 Experimental section

4.2.3.1 *Data*

Bagged kNN was evaluated with a simulated dataset and the benchmark dataset Wine [20]. The simulated dataset consists of a training matrix **X** of two classes, with fifteen objects and two variables each. The objects were simulated with variable values obtained from the univariate normal distributions,$\mathcal{N}(\mu,\sigma^2)$, where μ and σ^2 are the mean and the variance for the variable. The parameters used for each variable in the simulated dataset are $\mathcal{N}$(4.0,1.0) and $\mathcal{N}$(9.0,6.25) for the variables 1 and 2 of the class 1 and $\mathcal{N}$(7.0,1.0) and $\mathcal{N}$(14.0,6.25) for the variables 1 and 2 of the class 2. As a result, both classes are slightly overlapped (see Fig. 4.2).

The Wine dataset contains the results of chemical analysis of Italian wines of three different cultivars: Barolo, Grignolino and Barbera, with 59, 71 and 48 objects each, respectively. This dataset has 27 variables, although only 13 variables are used here. These 13 variables are described for this dataset in the UCI repository [21]. The Wine dataset was randomly divided into a training set with 100 objects (33 wines of Barolo, 37 wines of Grignolino and 30 wines of Barbera) and a test set

with 78 objects (26 wines of Barolo, 34 wines of Grignolino and 18 wines of Barbera).

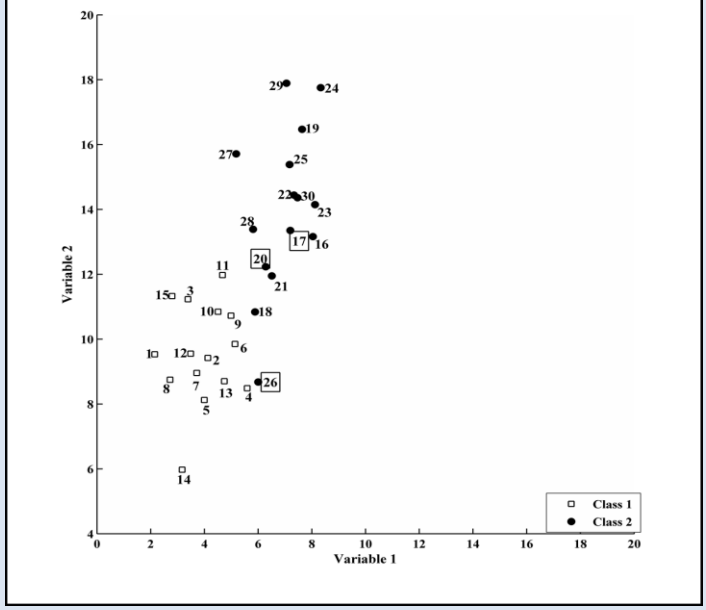

Figure 4.2 Simulated dataset with two classes and fifteen objects in each class.

4.2.3.2 *Uncertainty of the variables*

The ISO 17025 norm [22], which is the quality standard for all testing and calibration laboratories, requires the estimation of the uncertainty of analytical results. However, although the uncertainty in the measured variables should have been measured, it is not reported in the datasets published in the different repositories. Hence, the literature of multivariate classification based on published datasets does not take into account the uncertainty of the variables. In this paper the uncertainty was simulated in order to show the Bagged-*k*NN algorithm. Two simulation strategies were used: for the simulated

dataset, a percentage of the value of each variable was used as a uncertainty, and for the Wine dataset, the Horwitz equation [23] was used:

$$RSD_R = 2^{1-0.5\log c} = 2 \times c^{-0.1505} \qquad Eq.\ 4.6$$

where RSD_R is the relative reproducibility standard deviation of an analytical determination and c is the concentration of the analyte, expressed as mass fraction. Because RSD_R is expressed as a percentage, the uncertainty is transformed into concentration units by multiplying by the concentration value and dividing by 100. RSD_R can be transformed into an expanded uncertainty by multiplying by an appropriate coverage factor, $\mathcal{K}$, normally $\mathcal{K} = 2$ [1]. For the dimensionless variables: colour intensity, HUE and OD280/OD315 ratio, in the wine dataset, we used a 10 % of the variable value as uncertainty estimate (appendix A).

4.2.3.3 *Selection of the optimal parameters for kNN, PBkNN and Bagged-kNN*

The selection of the optimal k value is critical in *k*NN, PB*k*NN and Bagged-*k*NN. In this paper the optimal k was selected as the one that yielded the minimal classification error rate (CER) [9], that is, the percentage of objects that are assigned to the wrong class, calculated with leave-one-out cross-validation (LOOCV) for the simulated dataset and with the randomly selected test set for the Wine dataset. For *k*NN, LOOCV was performed by classifying the object i of **X** against the remaining I–1 objects of **X** for a given k value. This procedure was repeated until all I objects were classified and, finally, the CER was

computed. This procedure was repeated for different values of k. For PB*k*NN and Bagged-*k*NN, the optimal k was also obtained by LOOCV. In addition, 200 bootstrap were selected for PB*k*NN and 100 and 10 bootstraps were selected for the first (Efron's bootstrap) and second level (*U*-bootstrap), respectively in Bagged-*k*NN .

4.2.4 Results and Discussion

4.2.4.1 *Simulated dataset*

The simulated dataset is used to compare the posterior probabilities from Bagged-*k*NN with those calculated using PB*k*NN and *k*NN. The values of k for *k*NN, PB*k*NN were selected computing the CER using LOOCV for $k = 3, 5, 7$ and 9. Bagged-*k*NN was also evaluated using different levels of uncertainty: 1, 5, 10, 20 and 40 %, as a percentage of the variable value.

The CER obtained for the different values of k was the same (6.67%, objects 18 and 26 were misclassified) in *k*NN, PB*k*NN and Bagged-*k*NN. Hence, by simplicity, the minimal k, $k = 3$, was selected for *k*NN, PB*k*NN and Bagged-*k*NN.

Only the results for some of the objects in the overlapped region are commented. These objects are: object 10, which belongs to class 1, and objects 17, 18, 20 and 26, which belong to class 2 (Fig. 4.2). Table 4.1 shows the values of posterior probability for the classification in the true class obtained with *k*NN, PB*k*NN, Bayes and Bagged-*k*NN using increasing values of uncertainty. The Bayes' formula was used to obtain the true posterior probability. The rule in Bayes was to assign an unknown object to the class with the highest posterior probability [5, 9]. Calculation of this posterior probability requires the probability

density function and the a priori probability of each class, which are known in this simulated case but rarely known in a real classification problem.

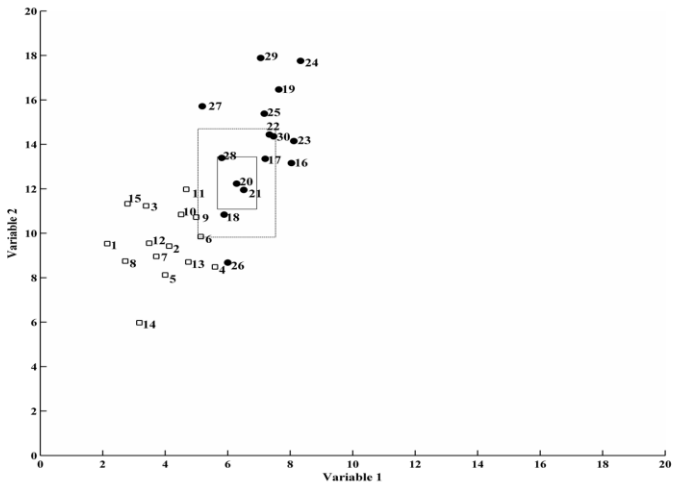

Figure 4.3 Simulated dataset with the uncertainty regions of object 20. The box with continuous line delimits the region where the uncertainty is 10% and the box with dotted line delimits the region where the uncertainty is 20%.

Table 4.1 shows that for object 20, the posterior probability decreases when the uncertainty in the *x*-variables increases. This is because when the uncertainty increases, the objects (both the unknown object and the other objects in the training set) lie in a wider region of the variable space. In this case, this makes the number of nearest neighbours of the class 1 increase, which, in turn, increases the probability of misclassification. This can be better seen in figure 4.3. This figure shows the uncertainty regions for object 20 (class 2). Note

that when the uncertainty is 20%, the number of nearest neighbours of class 1 increases with respect to the case when uncertainty is 10%. When *U*-bootstrap is applied to this object 20, the new bootstrap object can take any value within this region of uncertainty and, hence, the probability that the object 20 is closer to objects of class 1 increases.

We must recall, also, that the training objects also change the position during the bootstrap, which, in turn increases the probability that objects of class 1 are closer to object 20. Hence, the posterior probability of classification, which is obtained as a mean of $B \times D$ iterations, tends to diminish. Figure 4.4 shows the change in the posterior probability of classification due to this effect for different percentages of uncertainty. A similar behaviour is observed for objects 10 and 17.

Objects 18 and 26, which are in the borderline between classes, behave differently than objects 10, 17 and 20, which are not in the borderline (Fig. 4.4). For objects 18 and 26 the posterior probability of classification increases when the uncertainty increases. This is because these objects, which belong to class 2, are close to objects of class 1. This illustrated in figure 4.5 for object 18. Contrary to object 20, when the uncertainty increases, the region of uncertainty includes more objects of class 2 (the class object 18 belongs to); hence, the posterior probability of classification increases.

Table 4.1 Comparative posterior probability in the real class of the objects in the overlapped regions. True classes are indicated in parenthesis and incorrectly classified objects are indicated with an asterisk.

Object	Bayes	*k*NN	PB*k*NN	Bagged-*k* NN (increment of the uncertainty value)				
				1	5	10	20	40
10 (1)	0.960	1.000	0.958	0.904	0.923	0.881	0.850	0.768
17 (2)	1.000	1.000	1.000	1.000	0.999	0.999	0.978	0.818
18 (2)	0.680	0.333*	0.220*	0.340*	0.390*	0.398*	0.428*	0.393*
20 (2)	0.960	1.000	0.940	0.865	0.797	0.795	0.756	0.575
26 (2)	0.470*	0.000*	0.028*	0.035*	0.058*	0.107*	0.216*	0.320*

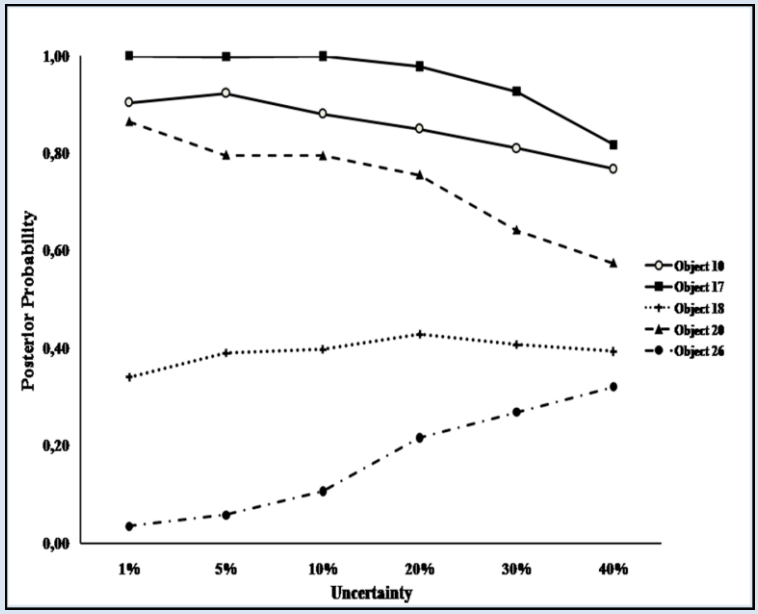

Figure 4.4 Posterior probability of classification versus value of uncertainty for selected objects in the overlapped regions.

The variation of posterior probability is smaller for the objects that are not in the region of class-overlap than for the objects in the overlap region. This is because the objects in the region of non-overlap are near objects of the same class. In this case, the variation of the variable

value due to its uncertainty does not change the class of the neighbours of that object e.g. object 17 in the figure 4.2 is surrounded only for objects of class 2 so the posterior probability is 1.0 when it is classified with Bayes, *k*NN, PB*k*NN and Bagged-*k*NN with uncertainty value of 1%. The value of the posterior probability only changes slightly into 0.999, 0.999 and 0.978 when the uncertainty is 5, 10 and 20 % respectively (see Table 4.1). Only when the uncertainty was increased to 40 % there is a significant decrease in the posterior probability.

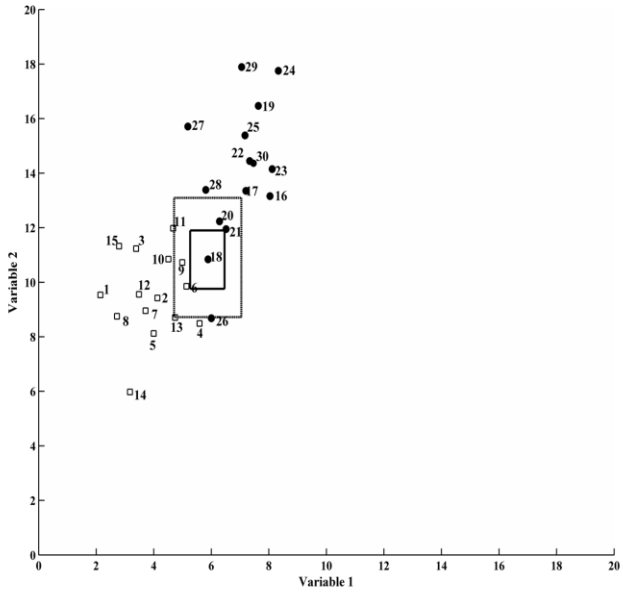

Figure 4.5 Simulated dataset with uncertainty regions of the object 18.

Moreover, with the aim of illustrating the importance of performing a nested bootstrap, we evaluated the effect of each type of bootstrap on the posterior probabilities. For this, we computed the Efron's bootstrap

variance (S_E^2) and the *U*-bootstrap variance (S_U^2), from the analysis of variance for a two-factor fully-nested design [24]. In this case, (S_U^2) is computed using the *D* posterior probabilities obtained with *U*-bootstrap for each Efron's bootstrap ($P_D(\ class\ c|\mathbf{x}_t)$) S_E^2 is computed using the *B* posterior probabilities obtained for all Efron's bootstraps. Figure 4.6 shows S_E^2 (Fig. 4.6a) and S_U^2 (Fig. 4.6b) with respect to the increase of the uncertainty. Figure 4.6a shows that for object 18, S_E^2 decreases when the uncertainty increases, contrary to S_U^2, which increases when the uncertainty increases (see Fig. 4.6b). This is because the posterior probabilities for this object, for each *U*-bootstrap iteration, have a higher variability. This variability is due to the fact that, when the uncertainty increases, the region of uncertainty increases too; therefore, for this object, we find different nearest neighbours each time, and we can classify it with values in a continuous range of probabilities. However, the Efron's bootstrap variance decreases when the uncertainty increases, i.e. the effect of moving the objects in the original training dataset to build a new training bootstrap sample is reduced when the uncertainty increases. This is due to the fact that, when the uncertainty increases, the overlap between classes increases too. This overlapping makes that the range of the posterior probability values, and the posterior probability values itself, decrease.

The increase of S_U^2 is similar for all the objects (see Fig. 4.6b). However, S_E^2 changes differently depending on the original position of the object. For objects 17 and 26, S_E^2 is practically constant (see Fig. 4.6a). For object 17, a slight increase was seen when the uncertainty was increased to 40 %, while for object 26, S_E^2 changes slightly when the uncertainty increases to 10% and 20 %. This is because these objects are surrounded mainly by objects of one class (Fig. 4.2); therefore, the

values of posterior probability obtained for each Efron's bootstrap hardly change.

Finally, the reliabilities of classification were compared with:

$$R_r = \sqrt{\frac{\sum_{i=1}^{N}(P_{i1} - P_{i2})^2}{N}} \qquad Eq.\ 4.7$$

where, P_{i1} and P_{i2} are the probabilities obtained for two methods. In this case, the Bayes' method is always the reference method (method 1) and *k*NN, Bagged-*k*NN, PB*k*NN are method 2. N is the number of evaluated objects. Small values of R_r indicate that the probabilities are similar in both methods. In this dataset, the values of R_r obtained by comparing Bayes and *k*NN, and Bayes and PB*k*NN are similar, 0.12; while R_r obtained between Bayes and Bagged-*k*NN are around 0.10 in the different values of uncertainty used. This means that the values of posterior probability obtained with the proposed method were more similar to Bayes than the values obtained with the other methods.

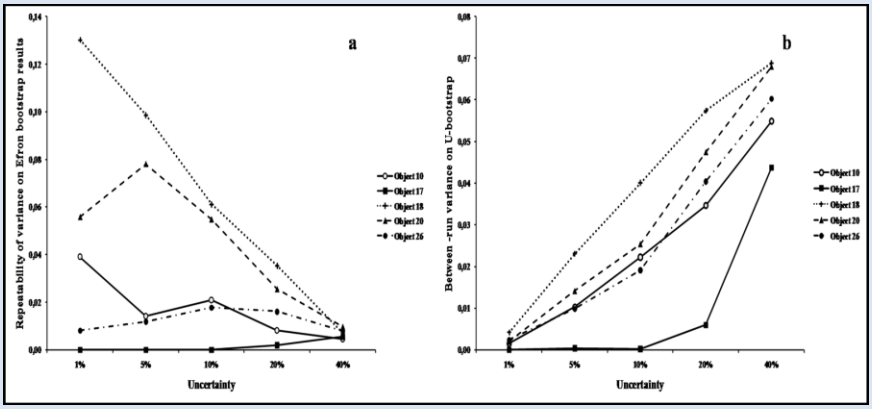

Figure 4.6 Variation of Efron's bootstrap variance s_E^2 (a) and the *U*-bootstrap variance s_U^2 (b) with respect to the increment of the uncertainty.

4.2.4.2 *Wine Dataset*

The Wine dataset was evaluated with Bagged-*k*NN, *k*NN and PB*k*NN. Figure 4.7 shows the distribution of the training and test objects on the scores of the Principal Component Analysis (PCA) calculated with auto-scaled data. Although, for each class, objects are grouped, the classes are slightly overlapped. In the overlap regions we find Grignolino objects 70, 74, 79 and 96 that are in the region of the class Barolo, and Grignolino objects 69, 84 and 97, which are close to the objects of class Barbera. The object number corresponds that in the UCI repository.

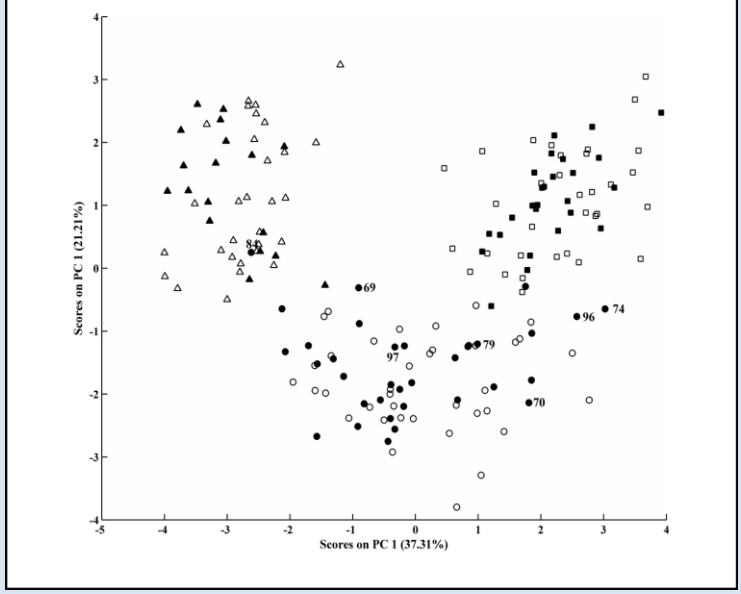

Figure 4.7 PC's for Wine dataset with auto-scaling of the variables for training and test set. Training set: Barolo (□), Grignolino (○) and Barbera (△) and Test set: Barolo (■), Grignolino (●) y Barbera (▲).

The values of k for *k*NN, PB*k*NN and Bagged-*k*NN were selected by computing the CER for the test set and for values of k of 1, 3, 5, 7, 9, 11, 13, 15 and 17. Figure 4.8 shows the values of CER with Bagged-*k*NN, *k*NN and PB*k*NN with Efron and Hamamoto's bootstrap. It is seen that the value of CER in PB*k*NN with Efron's bootstrap is the same for all k values evaluated. However, for Bagged-*k*NN, *k*NN and PB*k*NN with Hamamoto's bootstrap, the minimum value of CER was obtained with $k = 3$ so $k = 3$ was selected for all methods evaluated in this dataset.

Once selected the value of k, the test set was classified using *k*NN, PB*k*NN and Bagged-*k*NN. The CER for Bagged-*k*NN and *k*NN is 5.13 %, which is lower than the CER for PB*k*NN both using Efron's bootstrap (8.97%) and Hamamoto's bootstrap (7.69%). Hence, the proposed Bagged-*k*NN yields better classification results than PB*k*NN. The objects of classes Barolo and Barbera in the test set were classified correctly by all the methods. However, some of the objects of class Grignolino were misclassified, because they are in the overlapped region (Fig. 4.7). Below we will comment on the results for some interesting objects of class Grignolino.

Table 4.2 shows the posterior probability values, in the three classes, for some Grignolino objects obtained with Bagged-*k*NN, *k*NN and PB*k*NN with Efron and Hamamoto. It is seen that *k*NN provided only four different values of posterior probability, 0.000, 0.333, 0.667 and 1.00 while Bagged-*k*NN gave different values, between 0.123 and 0.820. Hence, although Bagged-*k*NN does not perform better than *k*NN, the advantage of Bagged-*k*NN is that the probabilities vary smoothly with the position of the objects in the variable space (a continuous range of probabilities can be obtained between 0 and 1), while the

probability values in *k*NN only take (k + 1) discrete values. Hence, in *k*NN all the objects located at a certain zone of the variable space that have the same neighbours will be assigned the same probability. In Bagged-*k*NN, however, each object has a slightly different value of probability that reflects the different proximity of that object to the objects in the training set.

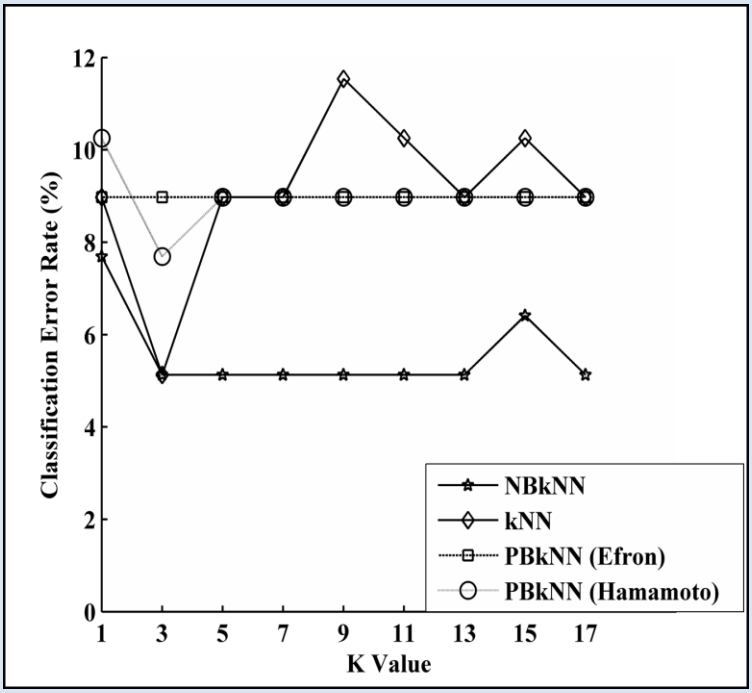

Figure 4,8 Classification error rate for different k values for evaluated methods.

Table 4.2 Posterior probability of classification of wine data set.
All objects in the table belong to class Grignolino.
The Classes correctly assigned by the methods are indicated with an asterisk.

Obj	*k*-NN (*k* = 3)			PB*k*NN - Efron (*k* = 3 ; B = 200)			PB*k*NN - Hamamoto (*k* = 3 ; B = 200)			Bagged-*k*NN (*k* = 3; B = 100 ; D = 10)		
	Barolo	Grignolino	Barbera	Barolo	Grignolino	Barbera	Barolo	Grignolino	Barbera	Barolo	Grignolino	Barbera
69	0.000	0.667*	0.333	0.023	0.520*	0.457	0.000	0.800*	0.200	0.247	0.460*	0.293
70	0.333	0.667*	0.000	0.523*	0.453	0.023	0.477	0.523*	0.000	0.343	0.399*	0.258
74	1.000*	0.000	0.000	0.947*	0.050	0.003	1.000*	0.000	0.000	0.710*	0.147	0.143
79	1.000*	0.000	0.000	0.840*	0.150	0.010	0.537*	0.463	0.000	0.410*	0.343	0.247
84	0.000	0.667*	0.333	0.000	0.450	0.550*	0.000	0.180	0.820*	0.206	0.241	0.553*
96	1.000*	0.000	0.000	0.993*	0.007	0.000	1.000*	0.000	0.000	0.753*	0.124	0.123
97	0.000	0.333	0.667*	0.077	0.353	0.570*	0.467*	0.170	0.363	0.285	0.387*	0.328

Table 4.2 shows that objects 69 and 70 were classified correctly by all the methods except PB*k*NN using Efron's bootstrap, which misclassified object 70. With *k*NN we obtained the same value of posterior probability (0.667) for both objects, while for PB*k*NN and Bagged-*k*NN save different probability because of the already commented higher sensitivity of PB*k*NN and Bagged-*k*NN for the position of the objects in the variable space. The low values of probability obtained with Bagged-*k*NN (lower than 0.460) indicate that additional information must be collected before the objects can be classified reliably. A similar conclusion can be obtained for object 79, which was incorrectly classified by all methods. In this case, kNN and PB*k*NN with Efron, assign high values of posterior probability (>0.840) to this classification which gives a false security to the result. PB*k*NN with Hamamoto's bootstrap and Bagged-*k*NN, on the other hand, assign a low probability (<0.540) to the classification, which warns about the possibility of misclassification. Objects 74 and 96, which are close to each other in the variable space (Fig. 7), are incorrectly classified by all methods. However, *k*NN, again, assigns the same probability to both objects (1.000), which, in addition, is excessively

high. In this case both PB*k*NN methods also assign a high probability to these objects. However, when the uncertainty is taken into account in Bagged-*k*NN, the posterior probabilities are different (because they depend on the position of the object in the multivariate space) and lower. Object 84 is correctly classified with *k*NN but misclassified by all the other methods. However, the correct classification in *k*NN is due to the fact that, of the nine nearest neighbours, only the second and the third belong to class Grignolino, while the others belong to class Barbera. Hence, when the uncertainty is considered, more neighbours of the class Barbera are found. For object 97 the situation is the contrary; it is misclassified by all methods except by Bagged-*k*NN. The reason is that its two nearest neighbours belong to the class Barbera and the third neighbour belongs to class Grignolino. When the uncertainty is taken into account, new neighbours of the class Grignolino appear which is sufficient to obtain a correct classification, although with a low probability.

Finally, despite being a nested bootstrap method, Bagged-kNN was not very time-consuming. LOOCV for the Wine dataset (k=3 , B = 100 and D=10) took less than five minutes in a personal computer with an Intel Core 2 Duo E6750 processor at 2.66 GHz and 3 GB of RAM.

4.2.5 Conclusions

We have proposed the Bagged *k*-Nearest Neighbours (Bagged-*k*NN) method, which combines *k*-Nearest Neighbours, bagging, and the uncertainties of the values of the variables of each object in the training data set. Bagged-*k*NN provides the reliability of classification for a particular object, which is calculated as a posterior probability.

This probability varies smoothly (a continuous range of values can be obtained between 0 and 1) depending on the position of the test objects in the multivariate space. This reliability value is also affected by the value of the uncertainty. In the simulated dataset the value of posterior probability changed for objects in the borderline regions when the uncertainty changes. These changes are due the position of the evaluated objects with respect to the objects in the training set. The reliability increases in the class where the evaluated objects have the most of nearest neighbours. Hence, the uncertainty in the measured variables should be taken into account in the classification methods. This uncertainty influences the reliability of classification, which, in turn can affect the results of classification.

Acknowledgements

The authors thank support of Department of Universities, Research and Information Society of Catalonia - Spain, for providing Joe Luis Villa's doctoral fellowship and project CTQ2007-66918 of the Spanish Ministry of Education and Science.

Appendix A

Mass fraction values used in Horwitz equation (see *Eq.* 4.6, section 4.4.2) for the Wine dataset.

Variable	Units	Mass Fraction
Alcohol	**%v**	0.01
Malic Acid	**g L^{-1}**	0.001
Ash	**g L^{-1}**	0.001
Alkalinity of Ash	[a]**g L^{-1}**	0.001
Magnesium	**mg L^{-1}**	0.000001
Total Phenols	**g L^{-1}**	0.001
Flavanoids	**g L^{-1}**	0.001
Nonflavonoid Phenol	**g L^{-1}**	0.001
Proanthocyanins	**g L^{-1}**	0.001
Color intensity	[b]Dimensionless	
HUE	[b]Dimensionless	
OD208 / OD315 of diluted wines	[b]Dimensionless	
Proline	**mg L^{-1}**	0.000001

[a] In the UCI repository, Alkalinity of ash is expressed in **meq L^{-1}** of NaOH, but in this article these units were changed to **g L^{-1}**.
[b] Dimensionless variables were not estimated by Horwitz equation. Instead, a 10% of the variable value was used as uncertainty value.

4.2.6 References

1. EURACHEM - CITAC (2000) Quantifying uncertainty in analytical measurement. CITAC Guide Number 4:126

2. Ramsey MP (1998) Sampling as a source of measurement uncertainty: techniques for quantification and comparison with analytical sources. Journal of Analytical Atomic Spectrometry 13:97

3. De Bièvre P (1997) Measurement results without statements of reliability (uncertainty) should not be taken seriously. Accreditation and Quality Assurance 2:269

4. Gurov SI (2004) Reliability estimation of classification algorithms. I. Introduction to the problem. Point frequency estimates. Computational Mathematics and Modeling 15:365

5. Webb AR (2002) Statistical pattern recognition. John Wiley and Sons Ltd, New York

6. Efron B (1979) Bootstrap method- Another look at the jacknife. The Annals of Statistics 7:1

7. Efron B, Tibshiran RJ (1993) An introduction to the bootstrap. Chapman & Hall, New York

8. Cover T, Hart PE (1967) Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13:21

9. Duda RO, Hart PE, Stork DG (2001) Pattern classification. John Wiley & Sons, New York.

10. Villa JL, Boqué R, Ferré J (2008) Calculation of the probability of correct classification in probabilistic bagged *k*-nearest neighbours. Chemometrics and Intelligent Laboratory Systems 94:51

11. Holmes CC, Adams NM (2002) A probabilistic nearest neighbour method for statistical pattern recognition. Journal of the Royal Statistical Society, Series B 64:295

12. Hamamoto Y, Uchimura S, Tomita S (1997) A bootstrap technique for nearest neighbor classifier design. IEEE Transactions on Pattern Analysis and Machine Intelligence 19:73

13. Breiman L (1996) Bagging predictors. Machine Learning 24:123

14. Mooney CZ, Duval RD (1993) Bootstrapping. A nonparametric approach to statistical inference. Sage, Newbury Park CA.

15. Davison AC, Hinkley DV (1997) Bootstrap methods and their application. Cambridge University Press, Cambridge

16. Wehrens R, Putter H, Buydens MC (2000) The bootstrap: a tutorial. Chemometrics and Intelligent Laboratory Systems 54:35

17. Hinkley D (1988) Bootstrap methods. Journal of the Royal Statistical Society Series B 50:321

18. Henderson AR (2005) The bootstrap: A technique for data-driven statistics. Using computer-intensive analyses to explore experimental data. Clinica Chimica Acta 359:1

19. De la Rosa, J. I., Fleury GA (2006) Bootstrap methods for a measurement estimation problem. IEEE Transactions on Instrumentation and Measurement 55:820

20. Forina M, Armanino C, Castino M, Ubigli M (1986) Multivariate data analysis as a discriminating method of the origin of wines. Vitis 25:189

21. Asuncion A, Newman DJ (2007) UCI Machine learning repository.

22. ISO-ICE (2005) General requirements for the competence of testing and calibration laboratories.

23. Horwitz W, Albert R (2006) The Horwitz ratio (HorRat): A useful index of method performance with respect to precision. Journal of AOAC International 89:1095

CHAPTER 5

MULTIVARIATE CALIBRATION WITH k-NEAREST NEIGHBOURS

5. Multivariate calibration with *k*-Nearest Neighbours

5.1 Introduction

This chapter introduces Direct Orthogonalization *k*-Nearest Neighbours (DO*k*NN), which uses direct orthogonalization prior to *kNN* to improve the prediction results. DO*k*NN predicts much better than classical *k*NN and its results are comparable to those obtained by PLS. The results improved further when the objects to be predicted are within the domain of the X-variable space of the training samples. Also, compared to PLS, predictions in DO*k*NN appear less influenced by the presence of outliers in the training set.

*k*NN for prediction uses a calibration set **X** of *J* variables measured on *I* objects and a vector of properties **y** ($I \times 1$). The predicted value, $\hat{y}_t$, for a new unknown object, $\mathbf{x}_t$, is computed as the mean or the weighted mean of the **y** values of its k nearest neighbours in **X**. *k*NN for prediction has some interesting properties. First, it has the advantages of a local method. Since prediction is done only using the nearest objects, it is robust to non-linearities that may exist in large regression domains. It is also robust to outliers in **X**, since only the objects that are closer to the object to be predicted are used for quantification. Second, it is non-parametric, so no assumptions have to be made about the probability distribution function of the data. Finally, like in other inverse calibration methods, it is not necessary to know all the constituents in the sample to predict the property of interest.

The closeness in the **X** space is measured, generally, using the Euclidean distance. This distance is affected by the signal from interferences, the noisy or irrelevant variables and by the type of data pre-processing, which can change the nearest neighbours of an object. Direct Orthogonalization (DO) is a pre-processing method used in multivariate calibration, which removes the information in **X** not correlated with **y**, thus improving the correlation between the remaining **X** and **y**. Therefore, nearest objects in **X** would be expected to have similar values in **y**, which favours the predictions with *k*NN.

Here we propose a new methodology to use *k*NN for prediction using DO. With the aim of comparing the ability of prediction of DO*k*NN, PLS and *k*NN were also tested. The method was tested with two spectroscopy datasets: the Fearn's dataset and a pharmaceutical dataset. In both cases DO*k*NN showed similar or even better results than PLS.

*k*NN is a local method, which is based on the fact that objects near in the variable space of **X** have similar values of property, **y**. Therefore, if the object to be predicted is not into or it is far from the boundaries of the variable space of the calibration set (i.e. space spanned by the objects in **X**) *k*NN provide large prediction error values, measured as the difference between the reference and the predicted value. Related to this, we have shown that the use of the Kennard and Stone's algorithm to divide the dataset into a calibration and a validation set improves the predictions by *k*NN. Also, we found that the best results of prediction with *k*NN were obtained when the percentage of variance captured by a PLS model in the **y**-block is higher than 90% in the first two factors. This can be due to that the information explained by the

variables in the first factors is the most influential in the distance calculus. This percentage of the variance in the y-block is increased by applying DO to the data before predicting with *k*NN.

DO*k*NN, like other methods as PLS, is greatly influenced by errors in the reference **y** values. This is because DO*k*NN uses these values to orthogonalize the data. So, errors in the **y** values can affect the identification of the nearest neighbours.

5.2 Paper. Multivariate calibration with *k*-Nearest Neighbours.

Journal of Chemometrics, submitted.

Edited for format

Multivariate calibration with *k*-Nearest Neighbours

Joe Luis Villa, Ricard Boqué*, Joan Ferré
Department of Analytical Chemistry and Organic Chemistry.
Universitat Rovira i Virgili
C/ Marcel·lí Domingo, s/n. 43007 Tarragona, Catalonia (Spain)

ABSTRACT

This paper introduces Direct Orthogonalization k-Nearest Neighbours (DOkNN) as a multivariate calibration method. The property of interest in an unknown sample is predicted as a weighted mean of the properties of the nearest neighbours. Direct orthogonalization is required to remove irrelevant variability in the independent variables and improve the identification of the k neighbours that will be used for prediction. DOkNN was evaluated with the Fearn's dataset in order to predict the protein content from spectral data. DOkNN predicted better (RMSEP of 0.36%) than the classical kNN with SNV preprocessed spectra (RMSEP of 0.96%). After a new split of the original dataset into new training and test sets using the Kennard and Stone's algorithm, the predictions of DOkNN were comparable to those of PLS, with RMSEP values of 0.28% and 0.26% respectively. When the method was tested with a pharmaceutical dataset to predict the amount of an active substance, DOkNN predicted better than PLS, with RMSEP values of 0.20% and 0.30% respectively. The results were also better when the data were split into a training and a test set using the Kennard and Stone's algorithm, with RMSEP values of 0.25% and 0.33% for DOkNN and PLS respectively. Compared to PLS, predictions with DOkNN are less influenced by the presence of outliers in the training set.

5.2.1 Introduction

The *k*-nearest neighbours (*k*NN) method [1] has been used in Chemistry mainly for classification [2-7]. Its use as a multivariate calibration method has been scarcely described [8-10]. When *k*NN is used for prediction, a training data set **X** of *J* variables (e.g., spectra) measured on *I* objects and a vector of properties **y** (*I*×1) are available. The predicted value, $\hat{\mathbf{y}}_t$, for an unknown object $\mathbf{x}_t$ is computed as a weighted mean of the *y* values of its *k* nearest neighbours in **X** [8-10].

Behind this prediction method there is the assumption that objects that have similar x-values have similar values of the property of interest *y*. While this assumption may not always be valid it can be fulfilled by certain datasets containing objects that have a very similar chemical matrix and where the property of interest creates the largest variation in the measured instrumental response. When applicable, predicting with *k*NN has some interesting properties. First, it has the advantages of a local method. Since prediction is done only using the nearest objects, it is robust to non-linearities that may exist in large regression domains. It is also robust to outliers in **X**, since there are usually extremes in the X-domain so they are not selected as neighbours for calculating the prediction. Second, it is non-parametric, so assumptions do not need to be made about the probability distribution function of the data. Confidence intervals can be calculated, for example, using resampling method such as bootstrap. Finally, like in other inverse calibration methods, it is not necessary to know all the constituents in the sample to predict the property of interest. The main limitation of *k*NN is the influence of the interferents' signal on the calculated distance used to identify the neighbours.

Varying amounts of interferents in the unknown sample may shift the sample over the multivariate space and change the sample neighbours, hence changing the final prediction. This drawback is overcome in this work by first removing the irrelevant variability in **X** with direct orthogonalization (DO)[11]and applying *k*NN to the resulting data. The method, called direct orthogonalization *k*-nearest neighbours (DO*k*NN) is used for predicting continuous properties on two different spectral datasets: the Fearn's dataset to predict the protein content from spectral data and a pharmaceutical dataset to predict the amount of an active substance in pharmaceutical tablets.

5.2.2 Methods

5.2.2.1 *Prediction with k-Nearest Neighbours*

Predicting with *k*NN is based on the assumption that objects near to each other in the variable space of **X** have similar values in their property value ***y*.** Hence, the property of an unknown object can be predicted by weighting the *y*-values of its *k*-nearest neighbours:

$$\hat{y} = \sum_{i=1}^{k} w_i \, y_i \qquad Eq.\ 5.1$$

where $\hat{y}$ is the property value assigned to the unknown object, y_i is the property value of the *i*th nearest neighbour (i=1,2,…,k), k is the number of nearest neighbours considered in the prediction, and w_i is a weight factor. Although the arithmetical mean can be used ($w_i = 1/k$) [8, 10], weighing for the inverse of the distance has the advantage of giving a higher weight to the property value of a closer neighbour [9]:

$$w_i = \frac{1}{d_i}\frac{1}{\sum_{i=1}^{k}\frac{1}{d_i}} \qquad Eq.\ 5.2$$

In *Eq.* 5.2, d_i is the distance of the unknown object to its *i*th neighbours in **X**. In addition to the weighted mean the geometrical mean or an exponential weighting scheme [12] can be used. The most used distance is the Euclidean distance. This distance is affected by the signal from interferences, the noise in the variables, the presence of irrelevant variables and by the type of data pre-processing, which can change the nearest neighbours of the object [13]. Hence, predicting with *k*NN improves when adequate pre-processing removes the systematic irrelevant variation in **X** and variable selection can keep only the independent variables that are the most correlated with the property of interest. Pre-processing with direct orthogonalization is described next. Variable selection is out of the scope of this paper. It must be noted that like in other regression methods, the predictive performance is the highest when the unknown objects are within the X-variable space of the training objects. Hence, a careful selection of the training and validation sets, taking into account the position of the objects in the variable space of **X** is required. The Kennard and Stone's algorithm (KS)[14] was used for this purpose.

5.2.2.2 *Direct orthogonalization k Nearest Neighbours (DOkNN)*

Used as a pre-processing method, Direct Orthogonalization (DO) [11] removes the variability in **X** that is not correlated with **y**, thus improving the correlation between the corrected **X** and **y**. When DO is applied in PCR and PLS, the predictive performance of the model is achieved with fewer factors [15]. When it is used with *k*NN, DO

improves the identification of the best neighbours of an unknown object. DO*k*NN is carried out as follows:

1) Apply direct orthogonalization to the column mean-centered training matrix **X** ($\bar{\mathbf{X}}$):

$$\mathbf{X}_{\mathrm{DO}} = \bar{\mathbf{X}}\left(\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathbf{T}}\right) \qquad Eq.\ 5.3$$

where **I** is a properly dimensioned identity matrix and **P** is the loadings matrix for *a* factors, resulting from principal component analysis (PCA) of $\mathbf{X}_0 = \bar{\mathbf{X}} - \mathbf{y}\mathbf{w}^{\mathrm{T}}$, where $\bar{\mathbf{y}}$ is the mean-centered vector of properties **y** and $\mathbf{w} = \bar{\mathbf{X}}^{\mathrm{T}}\bar{\mathbf{y}}(\bar{\mathbf{y}}^{\mathrm{T}}\bar{\mathbf{y}})^{-1}$.

2) Apply direct orthogonalization to the unknown object's **x**

$$\mathbf{x}_{\mathrm{DO}}^{\mathrm{T}} = \bar{\mathbf{x}}^{\mathrm{T}}\left(\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathrm{T}}\right) \qquad Eq.\ 5.4$$

where $\bar{\mathbf{x}}$ is the mean-centred **x** using the mean of the training data, and **P** is the same as in *Eq.* 5.3.

3) Predict the property of interest in the unknown object $\mathbf{x}_{\mathrm{DO}}$ as a weighed mean of the property values of the nearest neighbours in $\mathbf{X}_{\mathrm{DO}}$ (Eq. 5.1).

The optimal number of factors for the DO step (*a*) and of nearest neighbours (*k*) can be decided, for example, by cross-validation.

5.2.3 Experimental Section

5.2.3.1 *Datasets*

The Fearn's dataset [16] consists of NIR spectra of wheat samples collected at six wavelengths in the range of 1680-2310 nm. The property to be predicted is the percentage of protein. The original dataset contained a training set with 24 objects, and a test set with 26 objects. This dataset has often been used in the literature to compare the prediction ability of regression methods [16, 17].

The pharmaceutical dataset consists of 310 NIR transmittance spectra in the range of 7400-10500 cm^{-1}, and the property to be predicted is the amount of active substance content in the pharmaceutical tablets at four different dosages (5, 10, 15 and 20 mg/tablet) [18]. To generate the dataset the authors used 12 pilot and 12 laboratory scale batches. Of them, only the pilot batches were film coated. From each batch ten tablets were taken and analyzed by NIR and HPLC (reference method) and the results were expressed in the relative active substance content (% w/w). This dataset was originally studied with PLS. First, only the pilot batches were used for training and then pilot and laboratory batches were used together. Local models were also developed, i.e. the training was done with only the spectra of one of the dosages.

For these two datasets, the predictive performance of DO*k*NN was compared with that of *k*NN after different types of pre-processing commonly used for spectroscopic data. For the Fearn's dataset, the performance of DO*k*NN was compared to that of *k*NN and PLS regression using either the raw data, or after pre-processing with Standard Normal Variate (SNV) and de-trending [19-21]. The original

data split into training and test set was first considered in order to compare with the reported PLS results. A new split into training and test sets obtained by applying the KS algorithm was also tested. For the pharmaceutical dataset, DO*k*NN models were evaluated using the Root Mean Square Error of Cross Validation (RMSECV), to compare with the original work [18]:

$$\text{RMSECV} = \sqrt{\frac{\sum_{i=1}^{I}(\hat{y}_{cv,i}-y_i)^2}{I}} \qquad Eq.\,5.5$$

where $\hat{y}_{cv,i}$ is the predicted y_i when the object i was left out of the training set. Note that the objects were removed from the training set before any pre-processing (i.e. SNV, DO) was applied. In this case, 10 samples, corresponding to each batch, were left out from the training set. The remaining 300 samples were used to build the model and predict the left out samples. This procedure was repeated until all samples were predicted. Finally, the RMSECV was obtained using *Eq.* 5.5.

5.2.3.2 *Model optimization and prediction ability*

DO*k*NN was run using the Euclidean distance and weighing for the inverse of the distance (*Eq.* 5.2). The optimal k value for *k*NN and the number of factors a used for DO were chosen as the values that yielded the minimal RMSECV. The prediction ability of the models was obtained using the Root Mean Square Error of Prediction (RSMEP):

$$\text{RMSEP} = \sqrt{\frac{\sum_{i=1}^{I}(\hat{y}_i-y_i)^2}{I}} \qquad Eq.\,5.6$$

where $\hat{y}_i$ is the predicted y_i when the object i is from the test set.

5.2.4 Results and discussion

5.2.4.1 *Fearn's dataset*

5.2.4.1.1 Parameter selection for DO*k*NN

The leave-one-out cross-validation (LOOCV) error of the *k*NN models with *k* values from 2 to 15 was calculated. For DO*k*NN, *k* values from 2 to 15 and *a* values from 1 to 6 were evaluated. The original dataset without pre-treatment (RAW) was also tested.

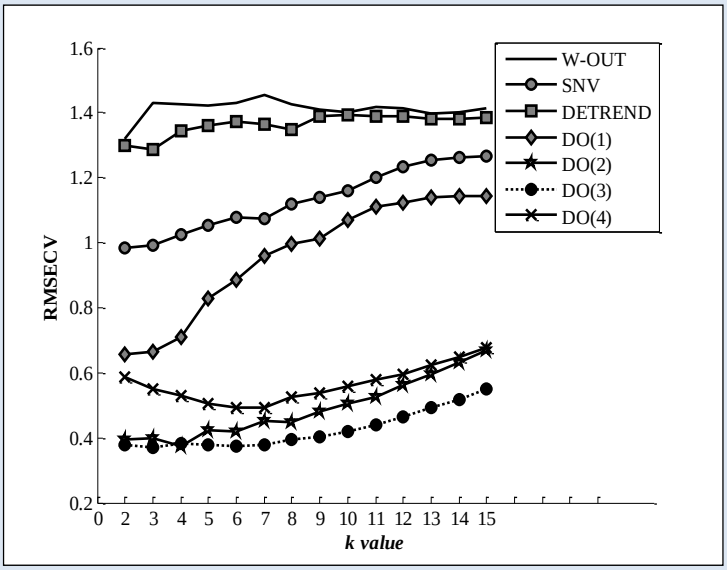

Figure 5.1 Selection of the *k* and *a* for Fearn's dataset with original splitting. The number factors used in the Direct Orthogonalization (DO) step are indicated into parenthesis.

Figure 5.1 shows the results for the original data split. The optimal *k* were *k*=2 for *k*NN without pre-treatment (W-OUT), *k*=2 for *k*NN after SNV and *k*=3 for *k*NN after de-trending. For DO*k*NN the optimal values

were *k*=3 and *a*=3. Note that not pre-processing the data yielded worse results (RMSECV=1.32%) than *k*NN after SNV (RMSECV=0.98%) and *k*NN after de-trending (RMSECV=1.29%). DO*k*NN with *k*=3 and *a*=3 gave three times lower RMSECV values (RMSECV=0.37%) than the best results obtained with *k*NN. Additionally, *k*NN after SNV and DO was evaluated with different *k* (1 to 15) and *a* (1 to 6) values. In this case the lower RMSECV value was obtained for *k*=3 and *a*=2 (RMSECV=0.30%). This value, although lower than the value obtained using only DO, requires two different pre-processing methods to be applied (SNV and DO).

5.2.4.2 *Performance of kNN, PLS and DOkNN*

The test set predictions for SNV-*k*NN (*k*=2) and DO*k*NN (*k*=3, a=3) were compared. The RMSEP of SNV-*k*NN (1.18%) was worse than for DO*k*NN (0.59%). Figure 5.2 shows the reference values vs predicted values in the test set for SNV-*k*NN (Fig 5.2a) and DO*k*NN (Fig 5.2b). For SNV-*k*NN, the determination coefficient after regressing the predicted against the reference values was R^2=0.42, lower than for DO*k*NN, with R^2 = 0.86. Figure 5.2 also shows the abnormal behavior of the prediction outlier object 12 in both methods. After removing object 12 the RMSEP decreased to 0.96% and 0.36% and R^2 improved to 0.44 and 0.94 for SNV-*k*NN and DO*k*NN respectively. DO*k*NN gave better predictions than SNV-*k*NN because the direct orthogonalization step removed the information of **X** not correlated with **y**, making the objects with similar **y** values be closer to each other in the orthogonalized variable space of **X**. For SNV-DO*k*NN RSMEP was 0.60 %. This value is lower than the RSMEP obtained using SNV only (1.18%) but slightly higher than the RSMEP after DO (0.59%). The removal of object 12

decreased the RSMEP from 0.60 % to 0.37%, similar to the value obtained with DO after the object 12 was removed (0.36%). The results obtained using SNV-DO-*k*NN are better than the results obtained by SNV-*k*NN but similar (slight higher) than the results obtained with DO*k*NN.

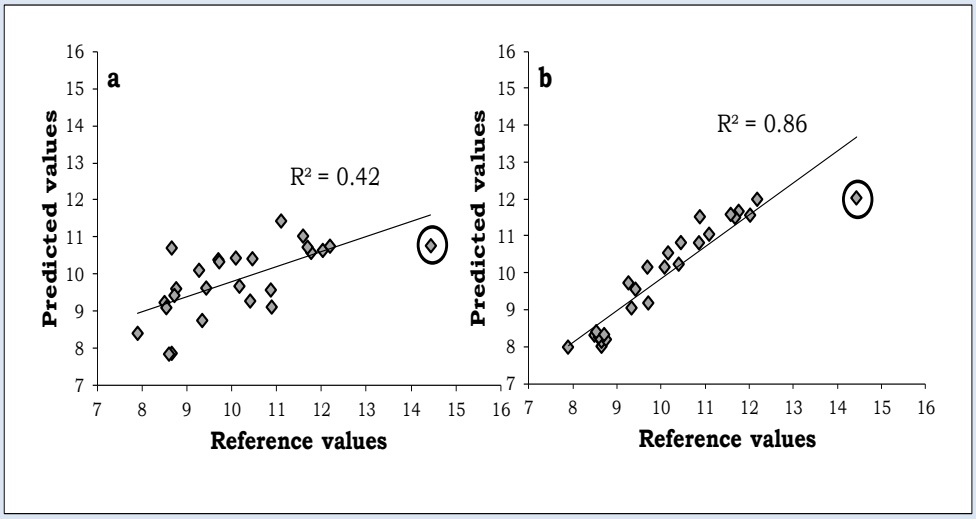

Figure 5.2 (a) Predictions for SNV-*k*NN(*k*=2) (b) Predictions for DO*k*NN (*k*=3, *a*=3). Removing the possible outlier (enclosed in the circle) the R^2 increase to 0.44 and 0.94 for SNV-*k*NN and DO*k*NN respectively.

For the Fearn's dataset, Faber et al.[17] reported an RMSEP value of 0.33% for a PLS model with mean-centred data and four factors. By removing object 1 of the validation set, identified as an outlier, the RMSEP decreases to 0.29%. Note that for PLS the object 1 was identified as a test outlier but not for DO*k*NN. This can be explained because the residual value of prediction (i.e. the difference between the predicted value and the reference value) for object 1 with PLS (-0.92, predicted value was 7.74 while reference value is 8.66) is large. This residual value is related to the leverage, thus objects with large

leverage have residuals of prediction higher than the average of all test samples. Therefore, this large residual can affect the (global) model used by PLS to predict the test object. On the contrary, DO*k*NN uses a local model to predict the test objects, therefore the objects identified as outliers from PLS could be predicted by DO*k*NN without problem.

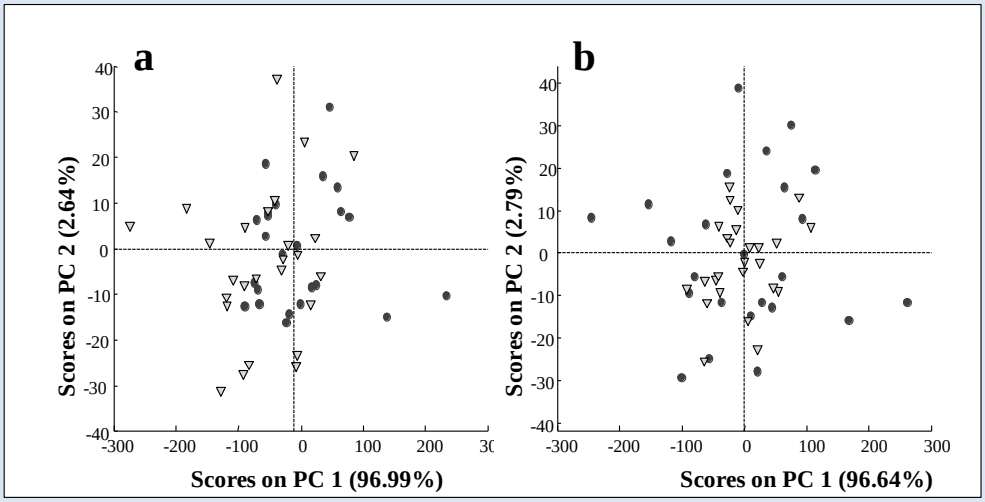

Figure 5.3 Score plot of Fearn's dataset. a) Scores of original data. b) Scores of dataset split by KS. Training samples (circles) and test samples (triangles).

Although the RMSEP is lower for PLS than for DO*k*NN, it must be taken into account that the original training-test set split is severely influencing the predictions with DO*k*NN. The scores plot on the two first principal components of the mean-centred data (99.63% of the training set variance) shows that the original test set is outside the range of the training data (Figure 5.3a). Since *k*NN requires that the test samples be near or within the variable space of the training set, the Kennard and Stone's (KS) algorithm was used to generate a new split into training and test set with the same number of objects for the

training and test sets as in the original split. Figure 5.3b shows the PCA scores plot of the new training set (99.43 % of explained variance). It can be observed that the new test set is now within the space defined by the training set.

For this new split the *k* value for *k*NN and the *k* and *a* values for DO*k*NN were selected again. The same *k* and *a* values and the same data preprocessing used for the original splitting were evaluated. The lowest values of RMSECV were found with *k*=3 for SNV-*k*NN and with *k*=2 and *a*=3 for DO*k*NN. The new test set provided an RMSEP of 1.29% for SNV-*k*NN and 0.28% for DO*k*NN. Again DO*k*NN predicted the best. *k*NN, besides of a favourable splitting of the dataset, also needs that the information in **X** be correlated to **y**. This correlation, in the original split, is higher than in the KS split when SNV is used. The correlation increases when DO is applied. The new sets were evaluated using PLS with five factors, giving an RMSEP of 0.23%. This value is lower than the RMSEP of DO*k*NN (0.28 %), however to build the PLS model was necessary exclude two objects of the training set (object 7 and 11 which correspond to the objects 17 and 25 in the original split). Although in this case the PLS results are lower than the results obtained using DO*k*NN, it can be considered as an alternative multivariate calibration method. Figure 5.4 shows the reference vs. predicted plot for DO*k*NN (5.4a) and PLS (5.4b). This figure shows that similar results are obtained by both methods.

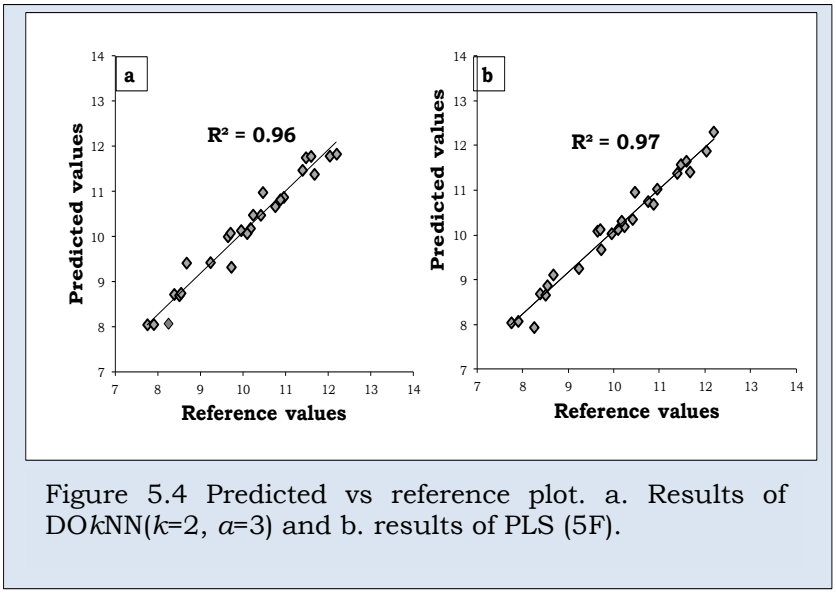

Figure 5.4 Predicted vs reference plot. a. Results of DO*k*NN(*k*=2, *a*=3) and b. results of PLS (5F).

When DO is used as a preprocessing step, the predictions with *k*NN improve. However, the presumed insensitivity of the *k*NN predictions to the presence of outliers must not be taken for granted, since outliers may influence the predictions through their influence in the DO step. To prove this, an artificial outlier was added to the dataset: the variables of the object 1 in the KS training set were arbitrarily multiplied by four.

When DO*k*NN was used, the optimal model (with the artificial outlier) was obtained with *k*=3 and *a*=4, with an RMSECV = 0.52%. The test set gave an RMSEP of 0.26% and R^2=0.96. This shows that DO*k*NN can provide good predictions even if outliers are present in the training set. In contrast, PLS needs to be optimized and the outlier in the training set needs to be excluded to build the model. However, once the model is optimized, PLS provides good prediction results (RMSEP = 0.23 %).

In this case the PLS model was built excluding objects 1, 5, 7 and 15, which correspond to objects 2, 15, 17 and 34 in the original split.

In DO*k*NN, like in PLS, suspicious samples in **X** can be studied by plotting the x-residuals. For DO*k*NN, Figure 5.5 shows the x-residuals as the sum of the squared differences between the original matrix and the matrix after direct orthogonalization. Object 1 is identified as an outlier because of its abnormal high x-residual with respect to the rest of the objects. If the object is removed from the training set, the values of RMSEP and R^2 do not vary (RMSEP = 0.26% and R^2 = 0.96), thus indicating that the outlier in the training set did not influence the prediction by DO*k*NN.

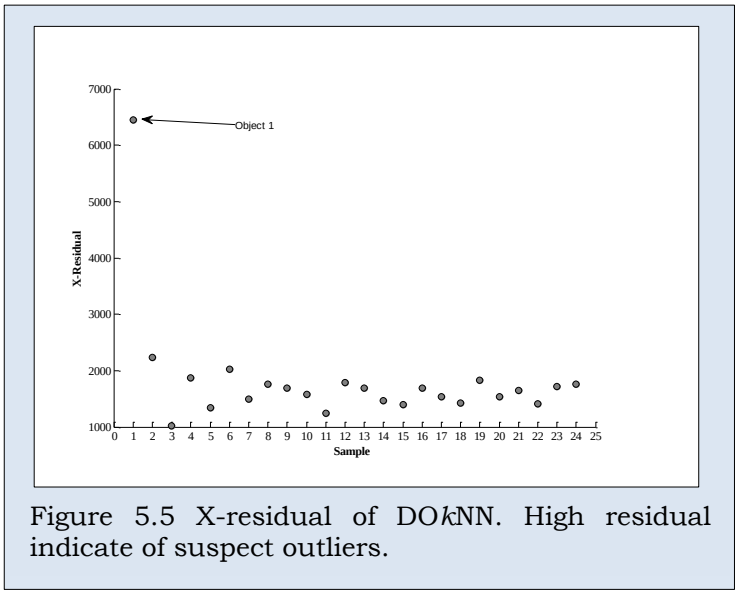

Figure 5.5 X-residual of DO*k*NN. High residual indicate of suspect outliers.

This is assuming that the outlier is present only in the **X** space, i.e. the reference **y** value is correct. Therefore, once the non-correlated information was removed with DO, the object is introduced into the

variable space and it can be used to predict. Figure 5.6 shows the first two PCs of the PCA applied to the Fearn's dataset with the artificial outlier without DO (Fig. 5.6a) and with DO (Fig. 5.6b). We can see that, without DO (Fig. 5.6a) the outlier is out of the variable space in the training set, and far from the variable space of the test set. However, when the DO is applied (figure 5.6b) object 1 is introduced into the variable space.

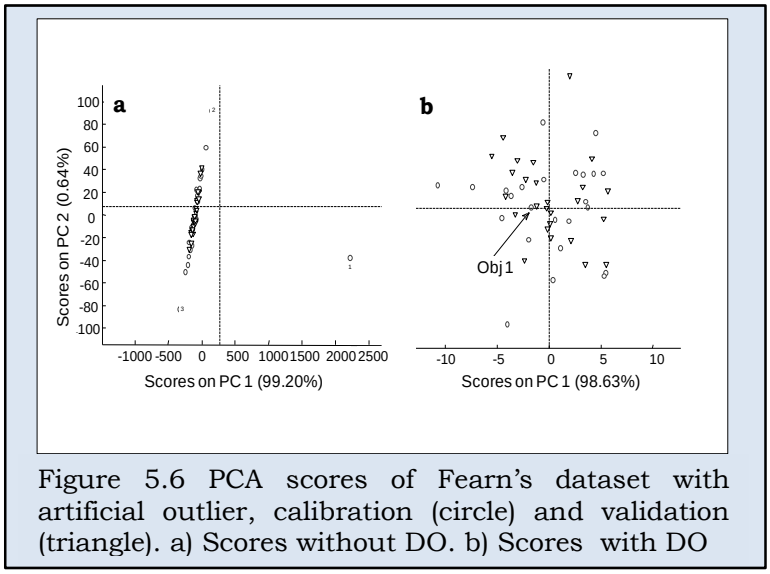

Figure 5.6 PCA scores of Fearn's dataset with artificial outlier, calibration (circle) and validation (triangle). a) Scores without DO. b) Scores with DO

5.2.5 Pharmaceutical dataset

The pilot data were evaluated with *k*NN without pre-treatment and after SNV, de-trending and DO with different values of *a*.

Figure 5.7 shows the RMSECV obtained for k = 2 to 15 and a = 1 to 4. It is seen that *k*NN without pre-treatment or with detrending gives worse results than SNV-*k*NN and DO*k*NN. SNV-*k*NN and DO*k*NN with a=3 provide similar results. In both cases, the RMSECV decreased until k was equal to or lower than 7 with only a slight variation for

values greater than 7. Hence k = 7 was selected as optimal. Both SNV-*k*NN and DO*k*NN yielded similar results, RMSECV=0.20% and R^2=0.97.

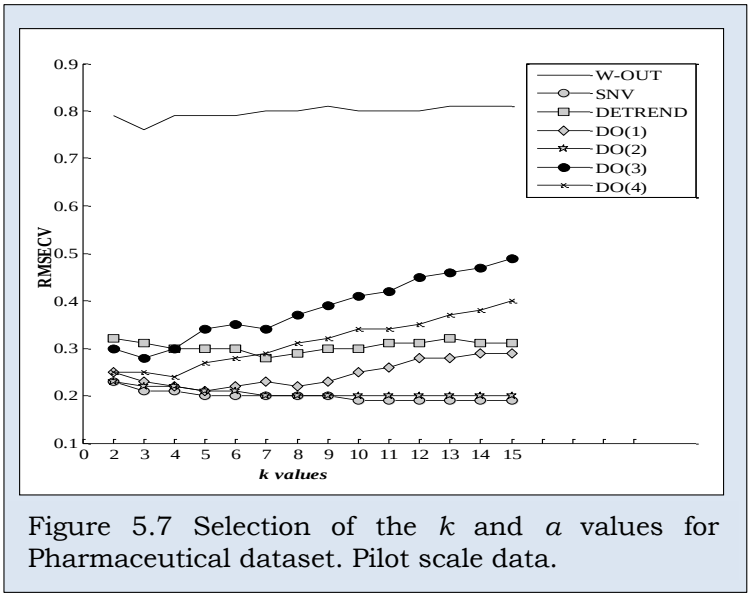

Figure 5.7 Selection of the k and a values for Pharmaceutical dataset. Pilot scale data.

This can be explained by the similar values of the correlation coefficient between each variable in **X** with respect to **y**, when SNV or DO was applied (Figure 5.8). This figure shows that a slight difference was found between the correlation coefficients, mainly for wavenumbers higher than 9100 cm^{-1}. These higher correlations imply that similar nearest neighbours can be found and, therefore, similar values of prediction can be obtained. As the correlation coefficient is a measure of the degree of association between two random variables, this means that if the columns of **X** are correlated with **y** then nearest objects in **X** are related to nearest objects in **y**, thus increasing the performance of the *k*NN method. This figure also shows the correlation coefficients obtained with the dataset before pre-processing. In this case, the correlation coefficients are lower, which is reflected in the higher

RMSECV values. In this figure we can see that the correlation coefficients increase in the region where the characteristic bands of the tablet are found, i.e. 8830 cm^{-1}[18]. This is important because this region can be used to find differences between nearest neighbours, i.e. in this region we can find the principal differences that can have an influence in the calculated distance between nearest neighbours and, therefore, in the predictions. For this dataset, RMSECV=0.30 % and R^2=0.92 were reported in reference [18], obtained with PLS with second derivative data and using a method of variable selection (iPLS). Note that DO*k*NN and SNV-*k*NN provide better predictions (RMSECV=0.20 %) than PLS.

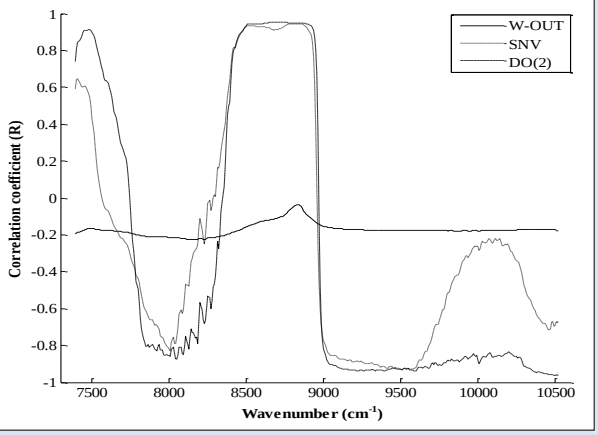

Figure 5.8 Correlation coefficient of each variable in X with respect to y

A second dataset, including pilot, full and laboratory scale batches was also evaluated by *k*NN with the same pre-treatment and *k* and *a* values. In this case, Dyrby et al [18] reported an RMSECV = 0.33% for a PLS model (using one component) with multiplicative scatter corrected (MSC) and first derivative data.

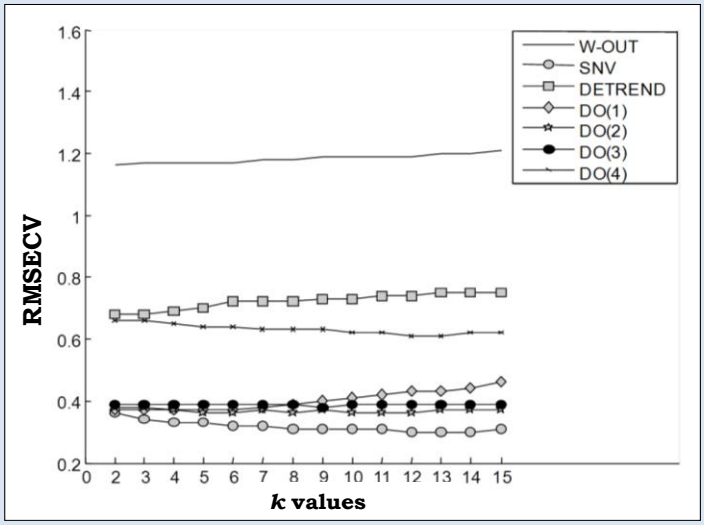

Figure 5.9 Selection of *k* and *a* for the pharmaceutical dataset. All data included.

Figure 5.9 shows the results obtained for *k*NN. The worst result was obtained with *k*NN without pre-treatment. For DO*k*NN, *k*=5 and *a*=2 were selected as the optimal values. *k*NN using de-trending gave RMSECV values lower than *k*NN without pre-treatment, but higher than those obtained for DO*k*NN and SNV-*k*NN, with a minimal value of RMSECV=0.68% with *k*=2. For this dataset the best results were obtained with SNV-*k*NN(RMSECV=0.33%with *k*=4), which is slightly lower than the obtained with DO*k*NN, i.e. RSMECV with DO*k*NN is only 0.03 % higher than the obtained with *k*NN with SNV. The results obtained with SNV-*k*NN are comparable to the results obtained with PLS (RMSECV=0.33%). The higher RMSECV obtained with DO*k*NN can be explained by the higher *y*-residual of certain samples with respect to the overall of the samples analyzed. These samples correspond to the batches of the dosage of 5 and 20 mg/tablets done at full scale, and 5 and 10 mg/tablets done at laboratory scale (figure 5.10). This figure

shows that, while most of the samples have random residuals near zero, the batches pointed above, i.e. encircled in the figure 5.10, have atypical residuals. This can be due to the fact that the RMSECV was obtained by predicting a batch each time, i.e. the variability of each specific batch to be predicted is not represented in the model used to predict, which can generate bad predictions.

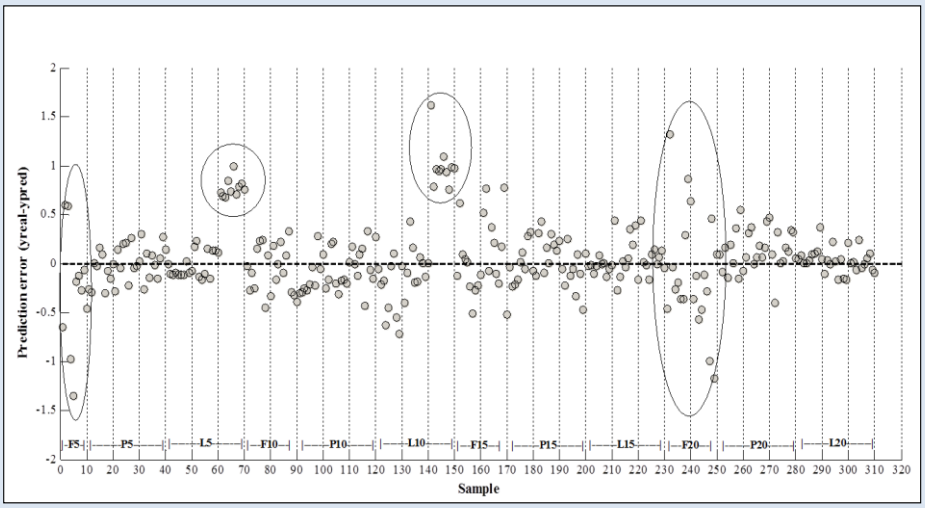

Figure 5.10 Prediction error of Pharmaceutical dataset in the global model by DO*k*NN. Grid divided the samples for each batch. Letter F, P and L indicate the batch obtained by full, pilot or laboratory scale respectively and the numbers indicate the dosage of the analyzed sample.

Finally the complete dataset was divided using KS into training and test set with 186 and 124 samples respectively. In this case, only SNV-*k*NN and DO*k*NN were compared. The optimal values obtained were k=4 for SNV-*k*NN and k=5 and a=2 for DO*k*NN. In both cases, the values of RMSEP and R^2 were 0.25% and 0.96 respectively. Equal values are obtained because for this dataset there exists a high correlation between **X** and **y**, which makes that SNV-*k*NN tends to

DO*k*NN. These values were compared with the results obtained with PLS using different pre-treatments (Table 5.1). This table shows that in all cases PLS gives worse predictions than DO*k*NN and SNV-*k*NN. The highest value of RMSEP was found using detrending (RMSEP=0.40% and R^2=0.91), while SNV and MSC showed similar results (RMSEP=0.33% and R^2=0.94).

Table 5.1. Result for pharmaceutical dataset using Kennard and Stone's for defining the training and test sets

Pretreatment	Model parameters	RMSEP (%)	R^2
PLS-SNV	a = 3	0.33	0.94
PLS-Detrend	a = 2	0.40	0.91
PLS-MSC	a = 3	0.33	0.94
*k*NN-SNV	k = 4	0.25	0.96
DO*k*NN	k = 5, a = 2	0.25	0.96

5.2.6 CONCLUSIONS

The use of DO*k*NN as a local prediction method has been presented. The method is appropriate when the objects that are close in the variable space of **X** have similar values of the property of interest **y**. The prediction error of *k*NN is improved by removing the variability in **X** not correlated with **y** using direct orthogonalization. If the object to be predicted is not into or it is far from the boundaries of the variable space of the training set (i.e. space spanned by the objects in **X**) DO*k*NN gives large prediction errors. For the Fearn's dataset, the split of the dataset into training and a test set with the Kennard and Stone's algorithm improved the predictions by DO*k*NN.

Although SNV can also improve the prediction by *k*NN, this is only possible if the correlation between each variable in **X** and **y** is increased once SNV is applied to the dataset. This is not always the case, e.g. for the Fearn's dataset. Also, contrary to SNV-*k*NN, DO*k*NN allows outlier detection in **X**, although these outliers may not affect the predictions obtained with DO*k*NN. Finally, DO*k*NN can provide similar or better predictions than PLS.

DO*k*NN, like other methods as PLS, is largely influenced by errors in the reference *y* values because the direct orthogonalization step uses these values to orthogonalize the data. So, errors in the *y* values can affect the identification of the nearest neighbours.

Work being developed will show the calculation of confidence intervals for predictions obtained from DO*k*NN.

Acknowledgements

The authors thank support of Department of Universities, Research and Information Society of Catalonia - Spain, for providing Joe Luis Villa's doctoral fellowship and project CTQ2007-66918 of the Spanish Ministry of Education and Science.

5.2.7 References

1. Cover T, Hart PE (1967) Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13:21

2. Todeschini R (1990) Weighted *k*-nearest neighbor method for the calculation of missing values. Chemometrics and Intelligent Laboratory Systems 9:201

3. Alsberg BK, Goodacre R, Rowland JJ, Kell DB (1997) Classification of pyrolysis mass spectra by fuzzy multivariate rule induction-comparison with regression, *k*-nearest neighbour, neural and decision-tree methods. Analytica Chimica Acta 348:389

4. Ros F, Guillaumet S, Rabatel F, Sevilla F, Bertrand D (1997) Combining global and individual image features to characterize granular product populations. Journal of Chemometrics 11:483

5. Pirogov A, Platanov M, Pletnev IV, Shpigun OA (1998) Application of the pattern-recognition method for modelling expert estimation of chromatogram quality. Analytica Chimica Acta 369:47

6. Beckonert O, Bollar ME, Ebbels T, Keun H, Antti H, Holmes E, Lindon J, Nicholson J (2003) NMR-based metabonomic toxicity classification: hierarchical cluster analysis and k-nearest-neighbour approaches. Analytica Chimica Acta 490:3

7. Lukasiak BM, Faria R, Zomer S, Brereton RG, Duncan JC (2006) Pattern recognition for the analysis of polymeric materials. Analyst 131:73

8. Ter Braak CJF (1995) Non-linear methods for multivariate statistical calibration and their use in palaeoecology: a comparison of inverse (*k*-nearest neighbours, partial least squares and weighted averaging partial least squares) and classical approaches. Chemometrics and Intelligent Laboratory Systems 28:165

9. Chee Ng, Yunde Xiao, Wendy Putnam, Bert Lum,Alexander Tropsha, (2004) Quantitative structure-pharmacokinetic parameters relationships (QSPKR) analysis of antimicrobial agents in humans using simulated annealing k-nearest-neighbor and partial least-square analysis methods. Journal of Pharmaceutical Sciences 93:2535

10. Yap CW, Li ZR, Chen YZ (2006) Quantitative structure–pharmacokinetic relationships for drug clearance by using statistical learning methods. Journal of Molecular Graphics and Modelling 24:383

11. Andersson CA (1999) Direct orthogonalization. Chemometrics and Intelligent Laboratory Systems 47:51

12. Nigsch F, Bender A, van Buuren B, Tissen J, Nigsch E, Mitchell JBO (2006) Melting point prediction employing *k*-nearest neighbor algorithms and genetic parameter optimization. Journal of Chemical Information and Modeling 46:2412

13. Todeschini R (1989) *k*-nearest neighbour method: The influence of data transformations and metrics. Chemometrics and Intelligent Laboratory Systems 6:213

14. Kennard R, W., Stone LA (1969) Computer aided design of experiments. Technometrics 11:137

15. Fernández Pierna JA, Massart DL, de Noord OE, Ricoux P (2001) Direct Orthogonalization: some case studies. Chemometrics and Intelligent Laboratory Systems 55:101

16. Faber K, Kowalski BR (1996) Prediction error in least squares regression: Further critique on the deviation used in The Unscrambler. Chemometrics and Intelligent Laboratory Systems 34:283

17. Faber NM, Song XH, Hopke PK (2003) Sample-specific standard error of prediction for partial least squares regression. Trends in Analytical Chemistry 22:330

18. Dyrby M, Engelsen SB, Norgaard L, Bruhn M, Lundsberg-Nielsen L (2002) Chemometric quantitation of the active substance (containing CN) in a pharmaceutical tablet using Near-Infrared (NIR) transmittance and NIR FT-Raman spectra. Applied Spectroscopy 56:579

19. Vandeginste BGM, Buydens LMC, De Jong S, Lewi PJ, Smeyers-Verbeke J (1998) Handbook of chemometrics and qualimetrics. Part B. Elsevier, Amsterdam, The Netherlands

20. Wu W, Walczak B, Massart DL, Prebble KA, Last IR (1995) Spectral transformation and wavelength selection in near-infrared spectra classification. Analytica Chimica Acta 315:243

21. Luypaert J, Heuerding S, Heyden YV, Massart DL (2004) The effect of preprocessing methods in reducing interfering variability from near-infrared measurements of creams. Journal of Pharmaceutical and Biomedical Analysis 36:495

CHAPTER 6

UNCERTAINTY OF PREDICTIONS WITH k-NEAREST NEIGHBOURS

6. Uncertainty of predictions with *k*-Nearest Neighbours

6.1 Introduction

In the previous chapter we introduced Direct Orthogonalization *k*-Nearest Neighbours (DO*k*NN) for predicting continuous properties with *k*NN. *k*NN had already been used to predict quantitative properties in Analytical Chemistry. However, the uncertainty of those predictions was not reported. Uncertainty is a fundamental parameter of an analytical result, and any method should include the procedure for estimating the uncertainty of its results. In this chapter we present the estimation of the uncertainty of predictions obtained with DO*k*NN, based on the bootstrap bias corrected and accelerated (BCa) method.

Bootstrap confidence intervals estimation is one of the most important areas of study of bootstrap methods and the Bias Corrected and accelerated (BCa) method is considered the best bootstrap method to compute the confidence intervals. BCa improves the accuracy, in terms of coverage and length of the intervals, of other bootstrap methods because BCa is obtained using a bias correction step.

To obtain the uncertainty of the prediction of an unknown object, first the object is predicted using DO*k*NN. For this Direct Orthogonalization is used to remove irrelevant variability in the independent variables and improve the identification of the *k* neighbours which are used for prediction using *k*NN. Then the uncertainty of this prediction is obtained using bootstrap. For this a bootstrap sample, **X***, is generated by resampling with replacement from the original training matrix **X**.

This new bootstrap sample is used to predict the unknown object using DO*k*NN and the predicted value is stored. This procedure is repeated *B* times, and then the *B* predictions are used to obtain the confidence intervals.

The method was evaluated using the Fearn's dataset to predict the protein content in ground wheat samples from NIR spectra. The predictions and their uncertainties, were compared with the predictions of PLS models. PLS confidence intervals were obtained using BCa like in DO*k*NN. DO*k*NN required the optimization of the *k* value used in *k*NN, and the number of factors *a* of the Direct Orthogonalization step. The prediction uncertainties were compared using the mean of the interval length values and the value of coverage probability for the test set. The length is the difference between the superior and lower values of the confidence interval and the coverage probability should be close to the standard 100(1-α)%. The uncertainties obtained using DO*k*NN are better than those obtained by PLS in terms of coverage probability; however, they are slightly higher in terms of interval length average.

6.2 Paper. Uncertainty of predictions with *k*-Nearest Neighbours

In preparation

Uncertainty of predictions with *k*-Nearest Neighbours

Joe Luis Villa, Joan Ferré, Ricard Boqué*
Department of Analytical Chemistry and Organic Chemistry.
Universitat Rovira i Virgili
C/ Marcel·lí Domingo, s/n. 43007 Tarragona, Catalonia (Spain)

ABSTRACT

We recently developed a new method for predicting continuous properties with the *k*-Nearest Neighbours (*k*NN) method called Direct Orthogonalization *k*NN (DO*k*NN). In this paper we use bootstrap to estimate the uncertainty of predictions with DO*k*NN. The method was evaluated using the Fearn's dataset to predict the protein content in ground wheat samples from NIR spectra. Predictions together with their uncertainties were compared to the ones provided by PLS. DO*k*NN provided similar prediction results than PLS with RSMEP values of 0.28% and 0.23% respectively. The uncertainty values obtained by DO*k*NN were compared with those obtained by PLS. The results show that the uncertainties obtained for DO*k*NN include a larger number of the reference values than those obtained by PLS, with coverage values 69.5% and 61.5% respectively. On the other hand, the average interval length in DO*k*NN is 0.74%, which is higher than the average interval length for PLS, which is 0.45%.

Keywords: Prediction methods; Nearest neighbours; Bootstrap; Uncertainty; BCa; Reliability.

*Corresponding author
E-mail addresses: ricard.boque@urv.cat

6.2.1 Introduction

The Direct Orthogonalization *k* Nearest Neighbours (DO*k*NN) was recently proposed [1]. This method provides predictions of continuous properties calculated as a weighted average of the property values of the *k* nearest neighbours of a sample. The method compared favourably with other classical approaches, including Partial Least Squares (PLS) regression. In this paper we show the estimation of the sample specific uncertainty of the DO*k*NN predictions.

In Analytical Chemistry *k*NN has been used to predict pharmacokinetic properties of drugs [2-4], but the uncertainty of the predictions was not reported. In the fields of mathematics and statistics [5, 6] an estimate of the variance of the *k*NN predictions was developed. This variance can be used to compute the uncertainty but it has not been used in analytical applications. In addition, the variance estimate by Altman [6] requires assumptions such as normality and low clustered data to be fulfilled, which are not always accomplished in chemical datasets.

Uncertainty is a fundamental parameter of an analytical result. It is considered that "*a result without reliability (uncertainty) statement cannot be published or communicated because it is not (yet) a result*" [7]**.** According to the ISO-GUM norm [8] the uncertainty of a calculated value is defined as a parameter, associated with the result of a measurement, which characterizes the dispersion of the values that could reasonably be attributed to the measurand. The two main methods for estimating uncertainty are error propagation and resampling strategies, such as jack-knife or bootstrap [9]. Bootstrap is a resampling method that can estimate a parameter θ of the

distribution of a population from the samples collected from it [10-15]. A sample is here defined in a statistical way, that is, a random selection of a given number of objects of the population. By bootstrap, the estimated parameter, $\hat{\theta}$, is obtained as the mean of B estimations done from B bootstrap samples, where each bootstrap sample is obtained by random sampling with replacement from the original dataset.

Confidence interval estimation is one of the areas where bootstrap has achieved major success, and different procedures are available [10-15]. Applications of bootstrap confidence intervals in the chemical and pharmaceutical fields can be found elsewhere [18-19].

In this paper we use Direct Orthogonalization *k*-Nearest Neighbours (DO*k*NN) to predict a property of an unknown object and then we use bootstrap to provide the uncertainty of that prediction. DO*k*NN is a variation of *k*NN for predicting continuous properties. The use of direct orthogonalization (DO) prior to *k*NN removes irrelevant variability in the independent variables and improves the selection of the k neighbours and, consequently, the prediction ability of the method. By bootstrapping, many new datasets $\mathbf{X}^*$ (called bootstrap training sets) are generated from the original training set $\mathbf{X}$. Then, for each bootstrap training set, DO is applied and a given unknown object is predicted using *k*NN. This procedure is repeated B times. These B predictions, obtained for all the bootstrap training sets, are used to compute the uncertainty of prediction.

Several approaches have been developed to compute bootstrap confidence intervals, namely: basic, percentile, percentile-t and bias

corrected and accelerated [10, 20, 21], among others [22, 23]. Of them, the bias corrected and accelerated (BCa) method is considered to be the best in terms of coverage [11]. The coverage of BCa intervals is closer to the nominal value, i.e. the coverage obtained with BCa intervals is similar to the coverage to be obtained with a parametric method.

According to Efron, BCa is better than other bootstrap methods in terms of accuracy [11]. This is because BCa is obtained using bias correction, which improves the results by increasing the coverage and reducing the length of the intervals. In this article the BCa method has been used to estimate the uncertainty of prediction from DO*k*NN.

The proposed method was tested against the Fearn's dataset to predict the protein content in ground wheat samples. Additionally, the uncertainties obtained were compared to the ones obtained with PLS regression. To compare the results, the BCa bootstrap method was used to obtain the uncertainty of prediction using DO*k*NN and PLS.

6.2.2 Methods

6.2.2.1 *Direct Orthogonalization k-Nearest Neighbours*

With *k*NN a given property of an unknown object is predicted from the property values of its nearest neighbours in a calibration set, **X**. For this, the distances between the unknown object and all the objects in **X** are calculated and the *k* first nearest neighbours are used to predict the property of the unknown object as a weighted mean of the property values of the *k* nearest neighbours, *Eq* 6.1 [24].

$$\hat{y}_t = \sum_{i=1}^{k} wd_i y_i \qquad Eq.\ 6.1$$

where

$$wd_i = \frac{1}{d_i} \frac{1}{\sum_{i=1}^{k} \frac{1}{d_i}} \qquad Eq.6.2$$

$\hat{y}_t$ is the property value assigned to the unknown object, y_i is the property value of the *i*th nearest neighbour (i=1,2,…,k), k is the number of nearest neighbours considered in the prediction, and d_i is the distance of the unknown object to the *i*th nearest neighbour in **X**.

This method applies direct orthogonalization (DO) prior to the prediction with *k*NN [1]. DO removes the variability in **X** that is not correlated with **y**, thus improving the correlation between the transformed **X** and the variation in **y** [25]. In this way, the multivariate distances of the neighbours to a given object are better related to the property being modelled. In DO*k*NN, the value k used by *k*NN and the number of factors used in the DO step must be optimised.

6.2.2.2 *Uncertainty of prediction*

In multivariate calibration, three approaches are mainly used to estimate the uncertainty of predictions: the U-deviation approach [26-27]; the Error-in-variables (EIV) approach [13, 28-29] and bootstrap resampling methods [21,30-31].

The U-deviation and EIV approaches are based on the regression model and make use of some estimates (e.g., leverage) obtained from the model to compute the uncertainty. These methods, however, cannot be

applied to DO*k*NN to estimate the uncertainty of predictions; given that in DO*k*NN do not use regression estimates to do the predictions. Instead, resampling methods have to be used. In this paper we used the bootstrap resampling method.

Bootstrap is used to generate empirical estimates of a statistic,$\hat{\theta}$ and uses them to make inferences about the population [7]. In a predictive model the statistic to be estimated is the predicted value [31-32]. For this, B bootstrap samples, **X***, are created by resampling with replacement from the original data. Then, each bootstrap sample is used to compute the statistic,$\hat{\theta}^*$. Finally, the B statistics $(\hat{\theta}_1^*, \dots, \hat{\theta}_B^*)$ obtained are used to infer about the population from which the data were taken.

Calculation of confidence intervals is the major application of bootstrap [34]. The aim is to calculate a confidence interval $(\hat{\theta}_{\text{low}}, \hat{\theta}_{\text{up}})$ of an estimate, $\hat{\theta}$, of a given parameter θ from B bootstrap replications, $\hat{\theta}_b^*$, obtained using bootstrap samples, $\mathbf{X}_b^*$. In all cases the confidence interval $(\hat{\theta}_{\text{low}}, \hat{\theta}_{\text{up}})$ is obtained from the values in the bootstrap distribution found in the α-percentiles given by $(B+1)\alpha$-th ordered values of the bootstrap distribution (i.e. bootstrap replications in ascending order, $\hat{\theta}_1^* \leq \hat{\theta}_2^* \leq \cdots \leq \hat{\theta}_B^*$)[14].

6.2.2.3 *Bootstrap bias-corrected and accelerated method (BCa)*

In the BCa method the confidence intervals are obtained using:

$$\text{BCa}(\hat{\theta}_{\text{low}}, \hat{\theta}_{\text{up}}) = (\hat{\theta}^{*(\alpha 1)}, \hat{\theta}^{*(\alpha 2)}) \qquad Eq.\ 6.3$$

where $\hat{\theta}^{*(\alpha 1)}$ is the bootstrap parameter, $\hat{\theta}^{*}$, and $\alpha 1$ and $\alpha 2$ are the position of the Bth order values of the bootstrap parameter, $\hat{\theta}^{*}$, obtained with the B bootstrap samples, and they are used to obtain the confidence intervals. $\alpha 1$ and $\alpha 2$ are obtained as:

$$\alpha 1 = \Phi(\hat{z}_0 + \frac{\hat{z}_0 + z_{(\alpha)}}{1 - \hat{a}(\hat{z}_0 + z_{(\alpha)})}$$

$$\alpha 2 = \Phi(\hat{z}_0 + \frac{\hat{z}_0 + z_{(1-\alpha)}}{1 - \hat{a}(\hat{z}_0 + z_{(1-\alpha)})} \qquad Eq.\ 6.4$$

$\Phi(\cdot)$ is the standard normal cumulative distribution function, $\hat{z}_0$ is the bias-correction, $\hat{a}$ is the acceleration factor and $z_{(\alpha)}$ is the 100αth percentile of a normal standard distribution. The value of $\hat{z}_0$ is the proportion of bootstrap replications lower than the observed estimated parameter, $\hat{\theta}$, and can be obtained as:

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\#\{\hat{\theta}_b^* < \hat{\theta}\}}{B}\right) \qquad Eq.\ 6.5$$

where $\#\{\hat{\theta}_b^* < \widehat{\theta}\}$ represents the number of $\hat{\theta}_b^*$ lower than $\hat{\theta}$, and Φ^{-1} is the inverse function of a standard cumulative distribution function. The acceleration, $\hat{a}$, indicates the rate of change of the standard error of $\hat{\theta}$ with respect to the true parameter value θ [7, 20]. The acceleration is used to correct the bias of the confidence intervals and it can be computed, as:

$$\hat{a} = \frac{\sum_{i=1}^{I}\left(\hat{\theta}_{(B2)} - \hat{\theta}_{(b2)}\right)^3}{6\left(\sum_{i=1}^{I}\left(\hat{\theta}_{(B2)} - \hat{\theta}_{(b2)}\right)^3\right)^{3/2}} \qquad Eq.\ 6.6$$

where, $\hat{\theta}_{(B2)}$ is the statistic estimated by a second level of bootstrap from the original sample and $\hat{\theta}_{(b2)}$ is the second level of the bootstrap replicate.

Within the bootstrap methods used to obtain confidence intervals, BCa is considered the best [5] because BCa uses the bootstrapped sampling distribution to estimate the constants $\hat{z}_0$ and $\hat{a}$, and uses them to remove the bias the confidence intervals generated using the percentiles of the parameters estimated using bootstrap. If $\hat{z}_0$ and $\hat{a}$ are zero the BCa method becomes the bootstrap percentile method [12, 14], where the confidence intervals are obtained as 1-(1-α) and 1-α Bth ordered bootstrap parameter, $\hat{\theta}^*$. Details about calculation of the confidence intervals using bootstrap can be found elsewhere [11-12, 14, 23, 35-36].

6.2.2.4 *Calculation of confidence intervals for DOkNN*

The steps to calculate the confidence intervals of the predictions by DO*k*NN are the following:

i. Predict the test object, *i*, using DO*k*NN and the original training set, **X**, to obtain $\hat{y}_i$.
ii. Generate a bootstrap sample, **X***, by resampling with replacement from **X**.
iii. Predict the test object, *i*, using DO*k*NN and the bootstrap sample, **X***, to obtain $\hat{y}_i^*$
iv. Repeat step ii and iii B times.
v. Use the B predictions of the test object, $\hat{y}_i^*$, to compute the bias-correction $\hat{z}_0$ using equation 6.5.

vi. Use the B predictions of the test object, $\hat{y}_i^*$, and $\hat{z}_0$ to obtain the acceleration using equation 6.6.

vii. Obtain the values of $\alpha 1$ and $\alpha 2$ using equation 6.4

viii. Obtain the BCa confidence intervals of prediction. $\hat{y}_i^*$. For this the values of the B predictions of the test object, $\hat{y}_i^*$, are ordered and the $\hat{y}_i^*$ predictions in the position $\alpha 1$ and $\alpha 2$ are selected as the confidence intervals.

6.2.2.5 *Partial Least Squares confidence intervals*

In PLS regression the uncertainty of prediction is usually obtained using an estimation of the standard deviation of the prediction error. The most commonly used methods are the U-deviation [37] and the Error-In-Variables (EIV) method [29]. Of them, the U-deviation is the most frequently used because it is user-friendly. However, bootstrap resampling methods have also been used [38].

6.2.2.6 *Selection of the optimal parameters for prediction with DOkNN*

In DO*k*NN two parameters have to be optimized: the optimal k value for *k*NN and the optimal number of factors, a, used in the DO step. They were selected by leave-one-out cross-validation (LOOCV) in the training set, as in reference [1], using the criterion of minimal Root Mean Square Error of Cross-Validation (RMSECV), computed as:

$$\text{RMSECV} = \sqrt{\frac{\sum_{i=1}^{I}(\hat{y}_{CV,i} - y_i)^2}{I}} \qquad Eq.\ 6.7$$

$\hat{y}_{CV,i}$ is the estimate for y_i when the object i was predicted and I is the number of training objects. Values of k between 2 and 15 and values of a between 1 and 6 were evaluated.

According to Wherens *et al.* [14], to compute the bootstrap confidence intervals, a minimum of 1000 bootstraps is required. Particularly they used 1999 and in this paper we used this value as the optimal.

6.2.2.7 *Comparison of the confidence intervals*

DO*k*NN prediction uncertainties were compared to the ones obtained with PLS regression [26, 28]. The prediction uncertainties were compared using the mean of the interval length values [12] and the value of coverage probability obtained for the test set in each dataset evaluated. For a given test object, the length is computed as:

$$\text{length} = \hat{\theta}_{\text{up}} - \hat{\theta}_{\text{low}}$$

where $\hat{\theta}_{\text{up}}$ and $\hat{\theta}_{\text{low}}$ are the upper and lower levels of the confidence interval, respectively. The percentage of coverage probability is computed as the percentage of the test objects for which the "actual" value of the property lies within the confidence interval found. Better results are obtained with lower values of the length average, which are indicative of the precision of the results, and values of coverage probability close to the standard 100(1-α)% [13] , which are indicative of the accuracy of the results, i.e. the confidence intervals computed include the real value of the evaluated objects in the test set.

6.2.3 Results and Discussion

6.2.3.1 *Fearn's dataset*

In this section we compare the confidence intervals obtained with DO*k*NN with those obtained by PLS, both were calculated using BCa bootstrap. First, the Kennard and Stone's algorithm was used to split the data into training and test sets.

6.2.3.1.1 Selection of the optimal parameters for DO*k*NN and PLS.

The *k* value for *k*NN and the number of factors, *a*, used in the DO step of DO*k*NN, were optimized in [1] for the Fearn's dataset. For the KS split, *k*=2 and *a*=3 were selected as optimal. For PLS the optimal number of factor was 5 and the calibration model was optimized excluding the objects 7 and 11 in the training set.

6.2.3.1.2 Comparison of the uncertainties of prediction

Figures 6.1 and 6.2 show the predictions obtained with DO*k*NN and PLS, respectively, for the Fearn's dataset. The reference value and the uncertainty of prediction obtained for each method are also indicated.

The PLS model (mean-centered data and 5 factors) with the Kennard and Stone's split yielded RSMEP values of 0.23 % and 0.28 % for DO*k*NN. Table 6.1 compares the average interval length and the coverage probability computed for both DO*k*NN and PLS.

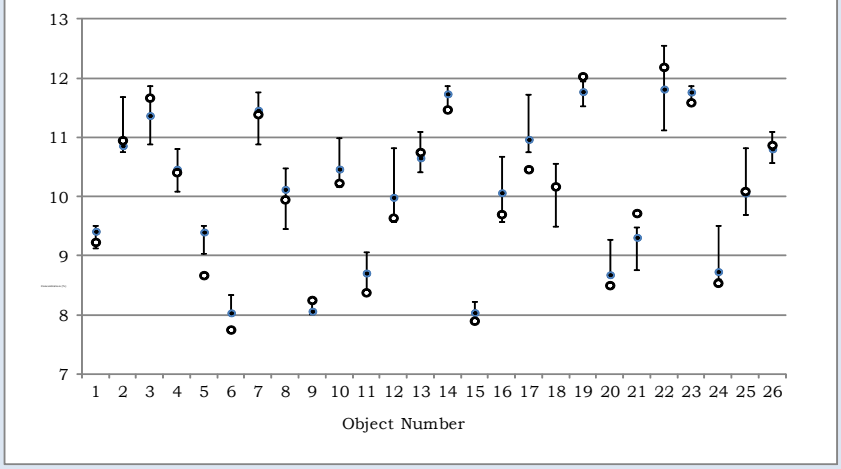

Figure 6.1 DO*k*NN prediction (black point), Reference values (white point) and uncertainty measure obtained by Bootstrap using DO*k*NN.

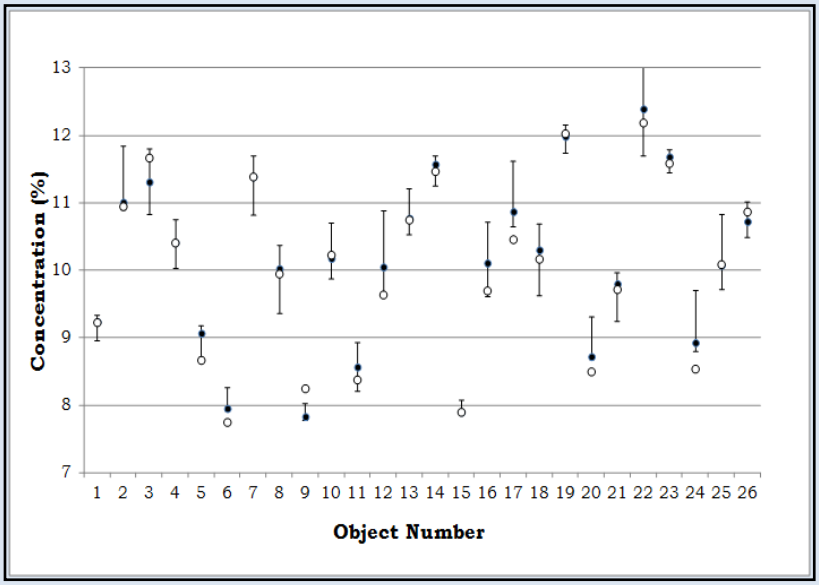

Figure 6.2 PLS prediction (black point), Reference values (white point) and uncertainty measure obtained by Bootstrap using PLS.

Table 6.1. Coverage Error and Length Average for DO*k*NN and PLS for Fearn's data Set with Kennard and Stone's Split

Method	Fearn's Data Kennard and Stone's Split	
	coverage probability	Average of the Length Values
PLS* (BCa)	61.5%	0.45%
DO*k*NN	69.2%	0.74%

For PLS 61.5% of the confidence intervals computed included the reference value. DO*k*NN had a coverage probability of 69.5 %, i.e. the reference values of 17 out of the 26 predicted objects were within the confidence intervals computed. The standard coverage probability was obtained as 100(1-α)%. In this case, given that α =0.05, the standard coverage probability was 95.0%. This means that, concerning coverage probability, the confidence intervals obtained using DO*k*NN and bootstrap BCa were better than those of PLS, because the coverage probability obtained with DO*k*NN was closer to the standard 95.0%. The average interval length for PLS was 0.45%, while for DO*k*NN was 0.75%. This means that the confidence intervals obtained with DO*k*NN are largest than the obtained with PLS. This difference can explain the difference observed in the coverage probability. The small differences in the RSMEP of both DO*k*NN and PLS models means that the predictions are very similar, therefore slightly higher interval lengths increase the probability that the reference value be contained in the interval. The differences in the length of the confidence intervals can be explained because PLS modelled the training data using all objects and, therefore, the small changes done in the training dataset when bootstrap is applied almost do not affect the model and the prediction results. In this sense, the predictions obtained for each object for each bootstrap set are very similar, thus making that the uncertainties

obtained and, therefore, the length are narrower. DO*k*NN, however, uses only the nearest objects of the unknown object to compute the predictions and small changes in the dataset can change significantly the prediction results and therefore their uncertainties.

6.2.4 Conclusions

DO*k*NN can be used for predicting continuous properties with a performance comparable to PLS. In this paper we have shown how to compute the uncertainty of the predictions of DO*k*NN using bootstrap. The uncertainties obtained are better than those obtained by PLS in terms of coverage probability and slightly higher in terms of average of length.

DO*k*NN combined with BCa bootstrap is useful to build multivariate predictive models and to calculate the uncertainty of the predicted values. Finally, a disadvantage of DO*k*NN with BCa Bootstrap is that the length of the uncertainties obtained can be higher than those obtained using PLS.

6.2.5 References

1. Villa J.L, Boque R, Ferre J (Submited). Multivariate calibration with *k*-Nearest Neighbours. Journal of Chemometrics, submitted.

2. Ter Braak CJF (1995) non-linear methods for multivariate statistical calibration and their use in palaeoecology: a comparison of inverse (k-nearest neighbours, partial least squares and weighted averaging partial least squares) and classical approaches. Chemometrics and Intelligent Laboratory Systems 28:165

3. Chee Ng, Yunde Xiao, Wendy Putnam, Bert Lum,Alexander Tropsha, (2004) Quantitative structure-pharmacokinetic parameters relationships (QSPKR) analysis of antimicrobial agents in humans using simulated annealing k-nearest-neighbor and partial least-square analysis methods. Journal of Pharmaceutical Sciences 93:2535

4. Yap CW, Li ZR, Chen YZ (2006) Quantitative structure–pharmacokinetic relationships for drug clearance by using statistical learning methods. Journal of Molecular Graphics and Modelling 24:383

5. Mack YP (1981) Local properties of *k*-NN regression estimates. SIAM Journal on Algebraic and Discrete Methods 2:311

6. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46:175

7. De Bièvre P (1997) Measurement results without statements of reliability (uncertainty) should not be taken seriously. Accreditation and Quality Assurance 2:269

8. ISO (1995) Guide to the expression of uncertainty in measurement (GUM).

9. Olivieri AC, Faber N(M, Ferré J, Boqué R, Kalivas JH, Mark H (2006) Uncertainty estimation and figures of merit for multivariate calibration (IUPAC Technical Report). Pure and Applied chemistry 78:633

10. Efron B, Tibshiran RJ (1993) An introduction to the bootstrap. Chapman & Hall, New York

11. Efron B (1987) Better bootstrap confidence intervals. Journal of the American Statistical Association 82:171

12. DiCiccio TJ, Efron B (1996) Bootstrap confidence intervals. Statistical Science 11:189

13. Faber N(M (2000) Comparison of two recently proposed expressions for partial least squares regression prediction error. Chemometrics and Intelligent Laboratory Systems 52:123

14. Wehrens R, Putter H, Buydens MC (2000) The bootstrap: a tutorial. Chemometrics and Intelligent Laboratory Systems 54:35

15. Wood (2005) Bootstrapped confidence intervals as an approach to statistical inference. Organizational research methods 8:454

16. Mooney CZ, Duval RD (1993) Bootstrapping. A nonparametric approach to statistical inference. Sage, Newbury Park.

17. Kiers (2004) Bootstrap confidence intervals for three-way methods. Journal of chemometrics 18:22

18. Serneels V, Pierre J (2005) Bootstrap confidence intervals for trilinear partial least squares regression. Analytica Chimica Acta 544:153

19. Pigeot (2001) The bootstrap percentile in food and drug administration regulations for bioequivalence assessment. Drug information journal 35:1445

20. Davison AC, Hinkley DV (1997) Bootstrap methods and their application. Cambridge University Press, Cambridge

21. Diciccio TJ, Romano JP (1988) A review of bootstrap confidence intervals. Journal of the Royal Statistical Society.Series B (Methodological) 50:338

22. DiCiccio T, Tibshirani R (1987) Bootstrap confidence intervals and bootstrap approximations. Journal of the American Statistical Association 82:163

23. Carpenter J, Bithell J (2000) Bootstrap confidence intervals: when, which, what? a practical guide for medical statisticians. Statistics in Medicine 19:1141

24. Nigsch F, Bender A, van Buuren B, Tissen J, Nigsch E, Mitchell JBO (2006) Melting point prediction employing k-Nearest Neighbor algorithms and genetic parameter optimization. Journal of Chemical Information and Modeling 46:2412

25. Andersson CA (1999) Direct orthogonalization. Chemometrics and Intelligent Laboratory Systems 47:51

26. De Vries S, J.F. Ter Braak C (1995) Prediction error in partial least squares regression: a critique on the deviation used in The Unscrambler. Chemometrics and Intelligent Laboratory Systems 30:239

27. Høy M, Steen K, Martens H (1998) Review of partial least squares regression prediction error in Unscrambler. Chemometrics and Intelligent Laboratory Systems 44:123

28. Faber K, Kowalski BR (1996) Prediction error in least squares regression: Further critique on the deviation used in The Unscrambler. Chemometrics and Intelligent Laboratory Systems 34:283

29. Faber NM, Song XH, Hopke PK (2003) Sample-specific standard error of prediction for partial least squares regression. TrAC Trends in Analytical Chemistry 22:330

30. Efron B (1979) Bootstrap method- another look at the jacknife. The annals of statistics 7:1

31. Efron B (1986) How biased is the apparent error rate of a prediction rule? Journal of the American Statistical Association 81:461

32. Bonate PL (1993) Approximate confidence intervals in calibration using the bootstrap. Analytical Chemistry 65:1367

33. Jones G, Wortberg M, Kreissig SB, Hammock BD, Rocke DM (1996) Application of the bootstrap to calibration experiments. Analytical Chemistry 68:763

34. Henderson AR (2005) The bootstrap: A technique for data-driven statistics. Using computer-intensive analyses to explore experimental data. Clinica Chimica Acta 359:1

35. Efron B, Tibshirani R (1986) Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical Science 1:54

36. Hall P (1988) Theoretical comparison of bootstrap confidence intervals. The Annals of Statistics 16:927

37. CAMO ASA (2004) The Unscrambler. 9.0

38. Fearn T (1983) A misuse of ridge regression in the calibration of a near infrared reflectance instrument. Applied Statistics 32:73

CHAPTER 7

CONCLUSIONS

Conclusions

7. Conclusions

7.1 Introduction

This chapter contains the conclusions of this thesis and some suggestions for future work about the reliability of classification and prediction using *k*-nearest-neighbours (*k*NN).

Multivariate classification models and multivariate calibration models are commonly used in Analytical Chemistry. The output of these models, i.e., a predicted class label, or a predicted value of a certain property, must be reported together with the degree of certainty of the result. Due to the complexity of the mathematics involved, the calculation of the reliability is not a direct task and it is a current concern for those applying multivariate methods.

This thesis has focussed on the calculation of the reliability values for the *k*NN classification method, and for a new calibration method that uses *k*NN for predicting continuous properties. Progress was made on the use of resampling methods (specifically bootstrap) to estimate the reliability of classification and prediction results. The classification and prediction method chosen was *k*NN, but some of the ideas produced in this thesis might be applied to other multivariate methods of classification and prediction.

7.2 About the reliability of classification using *k*NN

In the classical *k*NN method, a probability of correct classification is calculated only from the total number of evaluated neighbours (k) and the number of neighbours that belong to the selected class (k_c). This measure takes the same value for any unknown object over the variable space that has the same value of k_c. In order to obtain a value of reliability of classification that depends on the particular location of every unknown object, the probabilistic bagged *k*NN (PB*k*NN) was proposed. This method uses bootstrap to provide, for each evaluated object, a class label and a reliability value between 0 and 1 indicating the certainty of the assignment. This reliability value can, at the same time, be used in a classification rule so that the most reliable label is assigned. The use of bootstrap in this procedure allows the method to provide a different reliability value depending on the position of the object in the variable space with respect to the objects of the training set. With this approach, PB*k*NN provided better classification results than classical *k*NN and results that are comparable to those of LDA.

A second piece of work showed the influence of the uncertainty on the independent variables in the dataset on the reliability of classification. A new classification method, U-bootstrap, that combines *k*NN and bootstrap was presented. This method takes into account the uncertainty in the independent variables and uses it to generate a new dataset that is used to classify an unknown object. This procedure is repeated B times and the results are used to obtain the reliability of classification. With this method, the reliability of classification was seen to vary depending on the uncertainty in the dataset and depending on the position of the unknown object in the variable space

of the training set. For a given unknown object, the reliability of classification decreased when the uncertainty in the dataset increased.

7.3 About the reliability of prediction using *k*NN

A modification of *k*NN was proposed to predict continuous properties. Direct Orthogonalization *k*-Nearest Neighbours (DO*k*NN) uses a preliminary direct orthogonalization (DO) step, followed by *k*NN to predict continuous properties as a weighted mean of the property values of the neighbours. The prediction error in DO*k*NN decreases as the variability in **X** that is not correlated with **y** is removed. For this reason, a preliminary orthogonalization step was added. This local prediction method also performs better when the unknown object is inside the boundaries of the variable space of the training set (i.e. space spanned by the variables in **X**). In addition, DO*k*NN allows outlier detection in **X**, although the method itself is largely insensitive to the presence of outliers due to its local character. With the tested datasets, DO*k*NN provided similar or better predictions than PLS. DO*k*NN, like other methods as PLS, was largely influenced by errors in the reference *y* values because the DO makes use of these values. Hence, large errors in the *y* values can adversely affect the identification of the nearest neighbours.

Finally, DO*k*NN combined with BCa bootstrap was able to provide the uncertainty of the predicted values. BCa uses *B* bootstrap samples from the training set to predict the unknown object. Then these *B* prediction values are used to obtain both the constant used to correct the bias and the confidence interval of the prediction. The uncertainties obtained were lower than those obtained by PLS in terms of coverage

probability and slightly higher in terms of average of the length of the interval.

7.4 Future work

Although the proposed methods were able to improve the classification and prediction results obtained with *k*NN, there is still room for improvement on aspects such as:

- Probabilistic bagged *k*NN (PB*k*NN) and DO*k*NN work optimally for large training sets. Future works might focus on strategies for training the classifier using only a few training samples. Bootstrap strategies can be developed in this direction.
- BCa Bootstrap calculated uncertainties for DO*k*NN are higher than those obtained with PLS, for this reason future work can focus on the development of new methods to estimate the variance of prediction in order to improve the uncertainty measures for DO*k*NN.