A Comparison of Methods for Binary Classification problems

Raffaello Camoriano
Final Project for the Artificial Intelligence Course, A.A. 2011/2012
MSc in Robotics Engineering
Università degli Studi di Genova

Introduction

The present project report pursues the aim of describing the various steps, choices and results of a typical data analysis and supervised binary classification problem.
All the work has been developed using the R statistical computing environment.

A compendium of the phases in which the work has been divided is hereby reported:

1. Dataset import and normalization.
2. Preliminary qualitative analysis of correlation and noisy features.
3. Automated feature selection with Relief algorithm.
4. Quantitative and qualitative normality tests.
5. Feature extraction with Principal Component Analysis (PCA) and Multidimensional Scaling (MDS), comparison of the results.
6. Training, tuning and testing of different classifiers, comparison of the error rates.
7. Final validation.

In the following sections, these steps will be analysed in major detail.

Dataset overview

The provided benchmark dataset (number 18) is composed of 121 observations of 25 features, together with a column containing the class labels (1 or 2). All the features are numerical, with real values, and neither NAs, nor NaNs are present.
All the attribute values have been normalized between -1 and 1. The low number of samples in the dataset implies that the curse of dimensionality will thoroughly affect further analysis and classification.

Preliminary analysis

The first step has been to perform a qualitative statistical analysis of the distributions for each feature, including the class-conditional ones.
This task has been carried out by drawing standard and class-dependent boxplots, and by analysing the main statistical indexes. It has been possible to identify irrelevant and noisy features, which are not significant for telling the classes apart, and very significant features. Two indicative examples are shown in figures 1 and 2.
It has been concluded that the following features are probably the least relevant for classification:

x1, x2, x6, x11, x12, x14, x20, x21, 23, 25

However, the removal of the irrelevant and redundant features has been delegated to the Relief algorithm, which will be illustrated later.

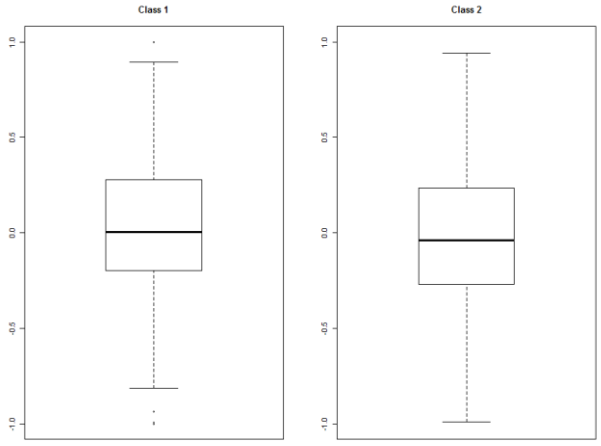

Figure 1: *class conditional boxplots for feature x2. The difference is not segnificant, this feature is most probably a noisy one.*

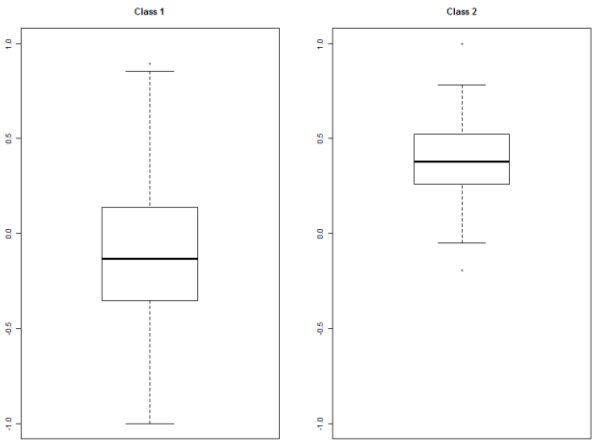

Figure 2: *class conditional boxplots for feature x19. The difference is very segnificant, this feature is the most significant for telling apart samples.*

Correlation between different features has also been studied, by means of the correlogram shown in figure 3. The most correlated couples of features, which might be redundant, are the following:

x1	**x23**
x2	**x6**
x2	**x20**
x2	**x23**
x6	**x8**
x6	**x13**
x11	**x21**
x12	**x17**
x12	**x24**
x12	**x25**
x18	**x20**

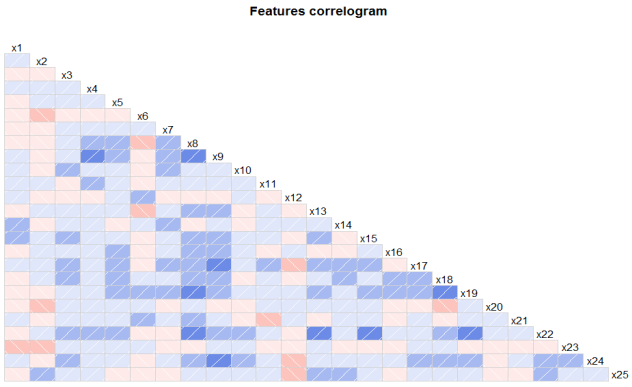

Figure 3: *features correlogram. Red: high correlation. Blue: low correlation*

Feature selection procedure

In order to automatically remove irrelevant and reduntant features, which might affect the performances of the classifiers, an enhanced version of the Relief algorithm for binary class problems has been used.

First of all, an extended dataset with three new dummy features has been created. The observations of these new features are normally distributed, in order to simulate noise. Then, the Relief algorithm is applied to the entire extended dataset, and the columns are sorted in decreasing order with respect to the computed Relief quality index.

In the following step, all the dummy features and the original features whose quality index is lower than the one of the second-ranked dummy feature are removed.

The information content of the remaining columns is evaluated by computing the classification error of a KNN-1 classifier.

As a rule of thumb, the algorithm is executed for N times, where N is the number of samples in the dataset. Only the set of features with the lowest error rate is kept for the next steps.

In figure 4, the boxplot showing the final number of significant columns for 100 itetarions of the feature selection procedure is reported. It must be noted that the best performances of the classifiers have been achieved with a number of features varying between 18 and 20, and that this totally automated procedure for feature selection sometimes removes too many or too few columns. However, the median of the number of kept columns is 18, which is a rather satisfactory result.

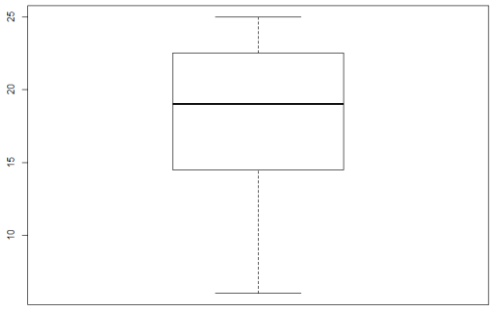

Figure 4: *Number of chosen columns over 100 iterations*

Two possible approaches for trying to increase the robustness of such procedure might be:

- The use of a different classifier. Naïve Bayes has already been tried, but the performance have not been affected significantly.
- Increasing the number of dummy features.

Normality tests

Both qualitative and quantitative tests for assessing the normality of the distributions of the features have been performed. These tests are important, because some of the methods which have been used in the analysis assume normal distributions.

The quantitative tests are the following (a threshold of 0.05 has been used for both of them):

- Shapiro-Wilk Normality Test.
- Lilliefors (Kolmogorov-Smirnov) Normality Test.

The results of both tests show that there is a small number of features for which normal distribution can not be assumed. Q-Q plots have been used for visualising these results. Two explanatory examples are reported below in figures 5 and 6.

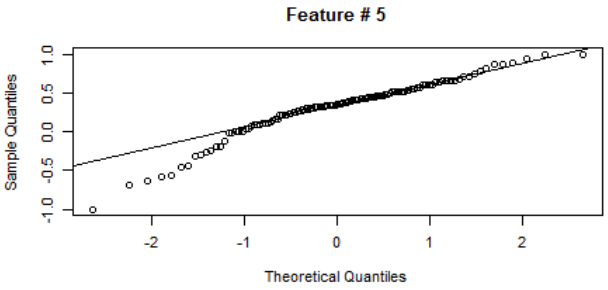

Figure 5: *Feature x5, test not passed.*
Shapiro-Wilk Normality Test result: $2.15*10^{-5} < 0.05$

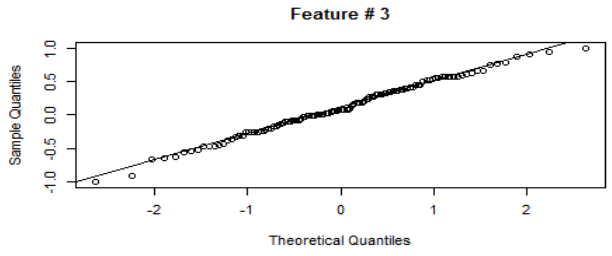

Figure 6: *Feature x3, test passed.*
Shapiro-Wilk Normality Test result: 0.79 > 0.05

Feature extraction

Two feature extraction methods have been applied:

- Multidimensional Scaling (MDS)
- Principal Component Analysis (PCA)

These methods allow a better visualization of the dataset and can help to reduce the effects of the curse of dimensionality by concentrating as much information as possible in a small number of features.

PCA is optimal in the case in which all the features have a normal distribution, while MDS is less binding in this sense. In our case, where not all the features have a normal distribution, for theoretical reasons it has been decided to use the results of MDS in the classification phase instead of the ones of PCA. This theoretical speculation is confirmed by results, as reported in figures 7 and 8. In fact, the mapping performed by MDS portrays two clearly separated clusters, one for each class.
The PCA variance histogram is reported in figure 9.

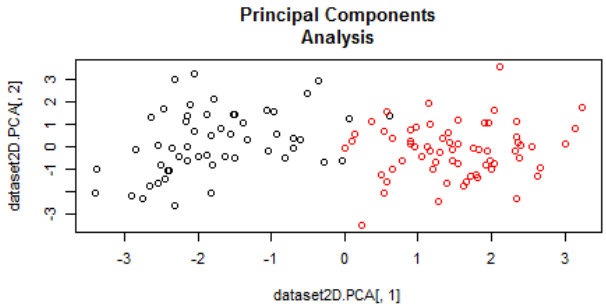

Figure 7: *PCA bidimensional mapping, the result is good, but the dataset is not linearly separable.*

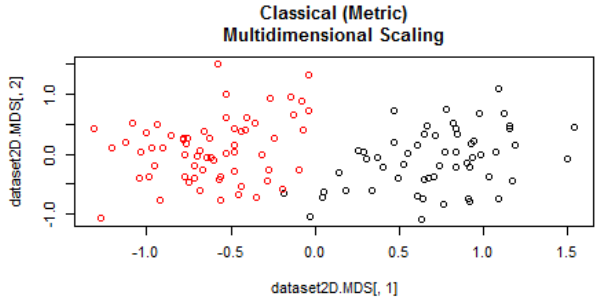

Figure 7: *MDS bidimensional mapping, the result is excellent, and the dataset seems to be linearly separable.*

Figure 7: *Standard deviations of the principal components.*

Classification

The dataset has been divided in various subsets for training, testing, optimization and validation of the chosen classifiers.
First of all, the training and test set have been divided randomly from the validation set, with a proportion of 80% vs 20%. It has been decided to dedicate only 20% of the patterns to validation because the number of patterns is very limited for training purposes.

Then, the training and test set has been used by each classifier in different ways. The ones for which optimization of hyperparameters is performed use 50% of the training and test set for optimization, and the remaining 50% for error rate estimation with bootstrap (B=100). These classifiers suffer from the curse of dimensionality and for the low number of pattern. This is why their performances are generally worse than the ones of other classifiers.

The following classifiers have been tested:

- Naïve Bayes
- Multivariate Parametric Bayes
- Multivariate Non-parametric Bayes
- K Nearest Neighbours
- Linear Discriminant Analysis
- Single Hidden Layer Neural Network
- Support Vector Machine
- Decision Tree (rpart)

A brief description of the characteristics, performances and tuning of each classifier follows.

Naïve Bayes (NB)

Assumes independent features and gaussian distributed likelihoods. No tuning has been performed, but the performances of this classifier are among the best. This is probably due to the inherent simplicity of the method, as there are features which are very good estimators for the class even by themselves (e.g. x19, as shown in figure 2).

Multivariate Parametric Bayes (MPB)

Assumes dependent features and normally distributed likelihoods. No tuning has been performed. At first, it has been trained using all the features selected by Relief.
However, after training it with only the first two features extracted by MDS, the error rate has decreased substantially, making MPB one of the best classifiers. The decision regions and the samples are shown in figure 8 (left).

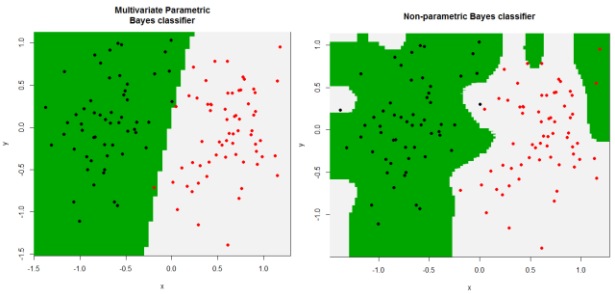

Figure 8: *MPB (left) and MNP (right) decision regions.*

Multivariate Non-parametric Bayes (MNB)

It is the least binding of the Bayesian classifiers, making no assumptions about the likelihood distributions. No tuning has been performed, and the model has been trained on the first two features extracted by MDS, because of the lack of a multi-dimensional implementation of the R Kernel Density Estimation function *kde2d* (MASS package).
The performance of the classifier is quite poor, most probably because of the low number of features for likelihood density estimation. The 2D decision regions are shown in figure 8 (right).

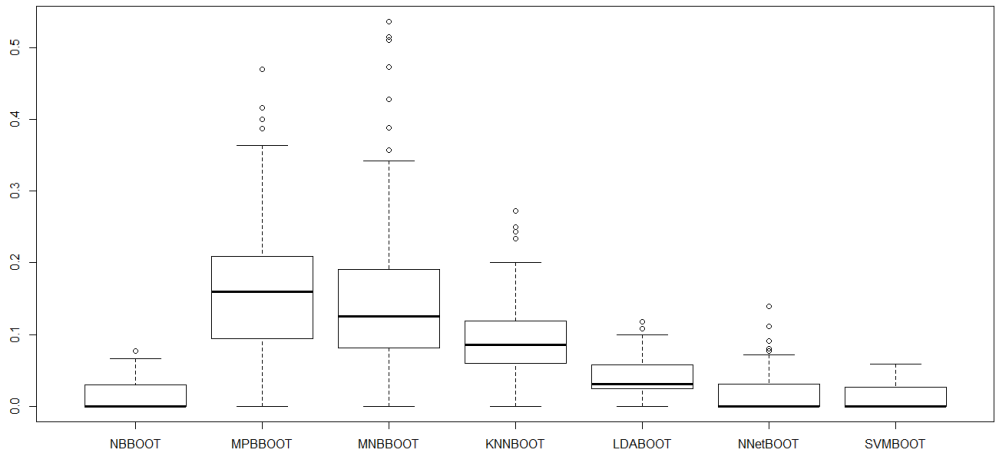

Figure 9: *Final comparison of the error rates of the classifiers, with N = 20 (best case).*

K Nearest Neighbours (KNN)

Tuning of parameter *k* has been performed via grid-search on this classifier (see code output for graph). The error rates of KNN are not among the best.

Linear Discriminant Analysis (LDA)

LDA assumes that the independent variables are normally distributed, which is not exact in this case. LDA has not been tuned, but its error rate is among the best.

Single Hidden Layer Neural Network (NNet)

The number of hidden neurons has not been tuned. As a rule of thumb, its value has been set to F/2, where F is the number of features of each sample.
Results are among the best.

Support Vector Machine (SVM)

The value of gamma has been tuned. After many iterations, the best value which has been found is 0.023, suggesting that the non-linearity of the gaussian kernel should be very limited. Its results are often the best.

Decision Tree (rpart)

The *cp* and *minsplit* parameters have been tuned via grid search. The error rates of the classifier are high (around 20%), it would probably need more samples for training.

The trained tree model is extremely simple. As we could expect, it often uses only feature *x19* as a discriminant.
However, the pruning of the tree seems to be too heavy.

Conclusions

The comparison between the error rates of the classifiers with N = 20 is shown in figure 9. The best results are obtained by SVM, closely followed by Naïve Bayes and Single Hidden Layer Neural Network.
However, as N could vary due to the relative instability of the feature selection procedure, Naïve Bayes seems to be a more robust solution.
The error rates of the final validation can be observed by launching the code, and they substantially confirm the ones shown in figure 9.

Bibliography

[1] Theoretical and Empirical Analysis of ReliefF and RRelief, Robnik-Sikonja & Kononenko, 2003
[2] Multi-dimensional Density Estimation, David W. Scott & Stephan R. Sain, 2004
[3] A Classification Learning Algorithm Robust to Irrelevant Features, H. Altay Güvenir
[4] Valutazione del dataset Wine utilizzando metodi di Machine Learning, Piana Stefano, 2011
[5] Pattern classification, R. Duda, P. Hart, and D. Stork, John-Wiley, 2nd edition, 2001