Machine learning methods in wine classification

Antti Paloposki, Aalto University 2014
Joel Huttunen, Aalto University 2014

Abstract

Background: Rating the wine quality is a task usually given to experts with and experience from thousands of wines. In this project, we attempted to use machine learning method for predicting the quality and type of different wines based on their physical qualities. **Results:** Types of different wines were easy to determine, as there was high correlation with for example sulphur dioxide and the probability for wine typebeing white. Quality for considerably more difficult to predict and neither of the methods used in that prediction succeeded in giving accurate predictions. **Conclusions:** It seems that wine and red wines have different physical attributes that make them easy to distinguish from each other. As quality was significantly more difficult to predict, it would suggest that either there is a very weak link with physical attributes and wine quality, the parameters given to us were not useful for quality-predictions, or that the quality of wine is highly subjective and thus very difficult to predict with an algorithm-based solution.

I. Introduction

The aim of the project was to predict types (white or red) and quality of wines based on their chemical properties. The dataset for building a predictor was inspired by classic Portugese 'Vinho Verde' wines and consisted of ~6000 wines with 11 chemical properties each. The dataset also included the type and quality of the wine for teaching a predictive algorithm.

Classifying wines provides a good opportunity to test data-analysis methods studied at the course. As we studied the differences between white- and red-wines, it became apparent that the difference between red and white wine is relatively easy to determine using the amount sulfur dioxide as a strong predictor for whether wine is red or white. However, using a single constant as a limit to predict the type of wine did not give very accurate results and therefore we decided to use some clustering algorithm to predict the type. Quality of the wine was slightly more complicated and did not yield very accurate results, since there is not always a clear relation with a particular chemical property and the quality of the wine.

During the project we gained valuable experience in practical data-analysis and applied the theoretical knowledge acquired during the course into a practical problem.

II. Methods

In our solution we used both support vector machines and k-nearest neighbor algorithms for predicting the type of wine. Mathworks documentation suggested that support vector machines would outperform k-nearest neighbors and other classification algorithms and that k-means would perform poorly, with high-dimensional data. [2] Nevertheless, we decided to also implement a k-nearest neighbor to see if it can fit data in eleven dimensions at all and to get some hands on experience with both methods. Support vector machine can be implemented on binary classification problems, where the data examined has two possible classes. SVM classification works by finding a hyperplane from data that has the largest margin between two classes. This is illustrated in figure 1. Support vectors are the data points closest to margin boundaries.

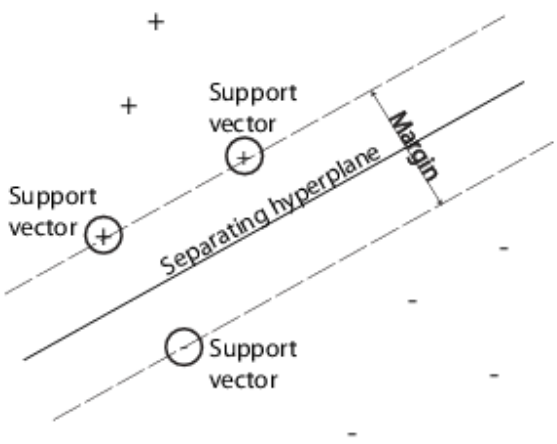


Figure 1: SVM visualized [3]

The course did not feature support vector machines, but we were confident in our ability to implement a solution with it. We tested support vector machines with different combinations of variables but finally went with a solution that used all the variables from the training data, since it gave the most accurate results. Theoretically it is possible, that some variables do not correlate at all with the type of the wine and decrease the effectiveness of the prediction, but this did not seem to be the case with our data.

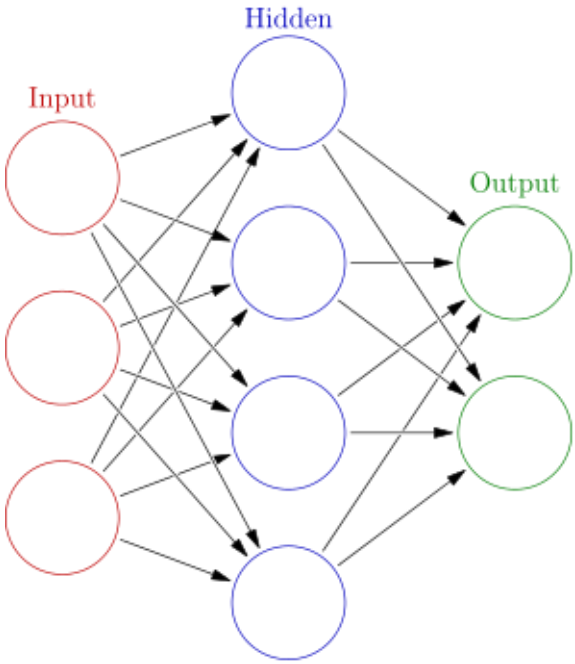
To get some redundancy in our results, we also tested wine type classification with k-nearest neighbor algorithm. We expected this algorithm to perform poorly with high-dimensional data and it usually requires dimensionality reduction, for example principal component analysis to work with high dimensional data. Nevertheless it was good to have some comparison with SVM.

Originally support vector machines were developed for binary classifications, but they have been extended to handle multi-class cases. This kind of multiclass classification with extended SVM was slightly challenging to implement, so we decided to predict wine quality with a neural networks method. K-nearest neighbor was again used for the sake of comparison.

To predict the quality of wines, we used pattern recognition with neural networks, since they are a powerful tool for multivariate complex decision boundary problems. [6]

In an artificial neural network consists of interconnected group of nodes that are connected to form a network which simulates biological neural network, for example the brain connections between these nodes are adaptive weights, which can be trained in supervised learning, as we already know the wine quality. In this problem we employed Matlabs neural network toolbox to train a neural network for wine quality predictions.

Figure 2: Artificial neural network



As illustrated in Figure 2, neural network determines the output value from input parameters. In this classification, we had eleven inputs (data for wine classification as provided in the assignment) and seven outputs (Probabilities that wine quality is same as the number of output node.) We can now use the highest probability as a prediction for the wine class and assign the wine quality as that number.

III. Experiments

As we started analyzing the wine types, we randomly selected a training set for SVM algorithm. We taught multiple SVMs with 30% validation and 70% training. With each iteration the sets were selected completely randomly. SVM model created with the training data was then compared against the validation data and the error was calculated. We ran multiple tests with different data dimensionality to see if some parameters were irrelevant/redundant, but finally went with all parameters, since that gave us the most accurate prediction. Cross-validation was done by running the script multiple times in a row. Since the training set was always selected randomly, the training data was different every time. We decided to first go with a linear kernel function and quickly discovered that it gave excellent results.

Classification with k-nearest neighbor was done with exactly the same set-up, but instead of SVM, we trained a k-nearest neighbor algorithm.

As we moved to predict the quality of wine, we switched to Matlabs neural networks toolbox.

To solve the quality of wine with neural networks, we used input of eleven parameters, the physical characteristics of wines, and gave it to hidden layer of 15 nodes. This gave us the 7 output classes which represented each of the possible values for wine quality

IV. Results

As we predicted the wine type, SVM with all parameters gave us a very good, roughly 1% error in validation set with multiple test runs. We also tried decreasing the dimensionality of training data when creating models with SVM, but this steadily increased the error in validation phase, so we decided to use all possible parameters in our classification. Theoretically it is possible that some parameters would be redundant, or irrelevant with the type of wine, but it would seem that this was not the case with the test data given to us.

K-nearest neighbor was also used to predict the type of wine, but as Matlab documentation suggested, with around 6-7% error margin it was significantly outperformed by SVM.

Predicting wine quality with k-nearest neighbor had error rate of 58%, this was probably because k-nearest neighbor is not very strong method when classifications with high-dimension data. Usually in a situation like this, dimensionality reduction is performed prior to applying k-nearest neighbor. [5]

Neural networks estimated wine quality with error of ~45%, which was still rather bad result, but considerably better than the result from k-nearest neighbors.

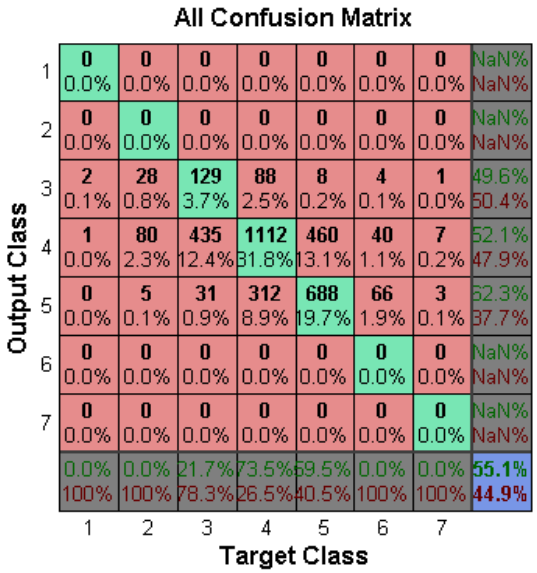


Figure 3: Confusion matrix of neural networks prediction

We also used F-score to measure the accuracy of our neural networks classification. F-score can be interpreted as the weighted average of precision and recall, where the best value is one and worst zero. In our results, the F-score is 0.495

F-score is calculated as:

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

V. Discussion

From the results we can see that predicting the type of wine succeeded very well, even with a method that typically performs poorly with high dimensions. The other attribute, quality, was not so easy to predict and both neural networks and k-nearest neighbor failed to predict quality accurately.

These results would suggest that the type of wine is strongly related to the chemical properties of wine. We did some research and found out this is true, as the amount of sulfur dioxide is a very good predictor for the type of wine by itself. [1]

Since predicting the quality of wine gave very vague results with high error rate, it would seem that contrary to the type of wine, quality does not have any particular physical properties or that those properties were not featured in the dataset we were given. It is also possible that quality of wine is highly subjective. When people with different tastes rate wines, the results will not have any clear characteristics for a good wine. This means that predicting quality will always yield very inaccurate results, as good and bad ratings will not have any consistent characteristics in the training and validation datasets.

References

[1] http://web2.slc.qc.ca/jmc/w05/Wine/conclusion.html

[2] http://www.mathworks.se/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html

[2] http://www.mathworks.se/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html

[3] http://www.mathworks.se/help/stats/support-vector-machines-svm.html

[4] http://www.mathworks.se/help/stats/classification-using-nearest-neighbors.html

[5] http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#Dimension_reduction

[6] http://se.mathworks.com/help/nnet/examples/wine-classification.html

[7] http://en.wikipedia.org/wiki/File:Colored_neural_network.svg

Appendices

See Matlab-files for the source code. Main.m contain the main sequence, which compares different methods by using ProjectModuled-properties as a framework.