# Class-Based Weighted NB for Text Categorization

C

**Mahsa Paknezhad**
*Shiraz University of Technology, Iran*

**Marzieh Ahmadzadeh**
*Shiraz University of Technology, Iran*

## INTRODUCTION

Naïve Bayes classifier is a supervised and probabilistic learning method (Manning, Raghavan, & Schuetze, 2008) which greatly simplifies learning by making the assumption that provided features are conditionally independent. Although this assumption usually does not hold, this classifier proves to compete well with other more sophisticated techniques (Rish, 2001). Moreover, being fast and easy to implement has resulted in frequent use of Naïve Bayes for text classification (Rennie, Shih, Teevan, & Karger, 2003). Studies comparing classification algorithms prove that Naïve Bayes is comparable in performance with decision trees and neural network classifiers (Han, & Kamber 2006). Many enhancements have been proposed so as to relax this unrealistic assumption. These enhancements are mainly in the area of feature selection and feature weighting (Lee, Gutierrez, & Dou, 2011). Feature selection is the process of selecting a subset of proposed features and using only these selected features in text categorization. Feature selection results in two main advantages: Firstly, by decreasing the amount of the effective vocabularies it makes classification more efficient. Secondly, it eliminates noise features and consequently makes classification more accurate (Manning et al., 2008). Feature weighting which obviously assigns a weight to each feature is more flexible than feature selection since feature weighting assigns continuous weights to features while feature selection assigns only 0/1 values (Lee et al., 2011). Many improvements have been proposed in both areas, but weight adjusting considering class attribute has rarely been investigated. In this chapter, we will propose the class-based weighted Naïve Bayes algorithm. In this algorithm, weight adjustment is performed for all samples with the same class attribute in the training dataset. Weight adjustment is achieved by examining different weights for each and every feature in the dataset and selecting the weight which contributes to the best improvement in the classification result. This mechanism will be elaborated further in section 3.

This chapter is structured as follows. In the next section, we will provide a brief review of other enhancements proposed to improve Naïve Bayes classifier. In section 3, we will introduce our proposed algorithm, class-based weighted Naïve Bayes algorithm and show the results of our experiments. Finally, a direction for future research and conclusion are given.

## BACKGROUND

In what follows, we will review some enhancements carried out in order to improve the performance of Naïve Bayes algorithm.

Joshi and Nigam (2011) conducted Naïve Bayes classification in two different ways: flat and hierarchical. In flat classification general Naïve Bayes approach was used, but in hierarchical classification classes in the training dataset were arranged in a hierarchical order according to the relationship among classes. This approach did not decrease the training time of the algorithm, but it

DOI: 10.4018/978-1-4666-5202-6.ch041

made classifying new documents faster since less comparison was required. Experiments showed that the hierarchical technique performed better than the flat technique except in some especial cases in which they were the same in performance.

Lee et al. (2011) proposed a feature weighting method using information gain to measure the significance of features. That is "a feature with a higher Information gain deserves higher weight". Furthermore, in order to remove the bias toward features with a wide range of values they considered split information measure while defining the feature weight. This measure which is also utilized in decision trees such as C4.5 assigns large split information to features with a lot of values. They proved that this algorithm outperforms the regular naïve Bayesian, Tree Augmented Naïve Bayes, NBTree and decision tree.

Similarly, Turhan, and Bener (2007) proposed utilization of heuristics to improve software defection prediction performance. They examined GainRatio, InfoGain and PCA to measure the level of importance of software metrics and evaluated them by weighted Naïve Bayes classifier. The results showed that InfoGain and GainRatio outperform standard Naïve Bayes and PCA based heuristic. Generally speaking, they proved that "linear methods lack the ability to improve the performance of Naïve Bayes while non-linear methods give promising results".

Zhang, Pena, and Robles (2009) introduced a multi-Label Naïve Bayes classification algorithm so as to be able to learn from instances with multiple labels. Also, the principle component analysis (PCA) technique was utilized as a feature extraction technique to improve the performance of the algorithm by eliminating redundant features. Moreover, feature subset selection techniques based on a genetic algorithm (GA) were used to choose more appropriate features for classification. The experiments showed that in comparison with other multi-label learning algorithms, this method gains a convincing performance.

Another approach proposed by Ratanamahatana and Gunopulos (2002) to improve Naïve Bayes was to use C4.5 algorithm. This new version of Naïve Bayes algorithm was called selective Bayesian classifier (SBS) and used only those features that C4.5 uses in constructing its decision tree. This way the algorithm needed fewer training instances to achieve high classification accuracy. It showed that SBS almost always gives a better result than Naïve Bayes algorithm and outperforms C4.5 in many cases which C4.5 outperforms Naïve Bayes.

Finally, Frank, Hall, and Pfahringer (2009) suggested a lazy approach named locally weighted Naïve Bayes. This approach does not build any Naïve Bayes model from the training dataset until the classification time. It merely uses a weighted set of training instances which are at the neighborhood of the test instance. Having no strong dependency at the neighborhood of the test instance makes this technique to perform well. They showed that locally weighted naïve Bayes evenly outperforms the standard one.

In brief, a wide range of actions have been taken to improve the performance of Naïve Bayes. These efforts focus on improving different aspects of the algorithm, including feature extraction, feature weighting, or on utilizing the algorithm for more general applications. However, in this paper our focus is on improving weighted Naïve Bayes to gain more accuracy in text classification. The next section gives a deep explanation about our approach.

## MAIN FOCUS

### Class-Based Weighted Naïve Bayes Algorithm

Now we will introduce our own approach for improving the performance of Naïve Bayes classifier in text classification. What we have done is similar to what Han, Karypis, and Kumar (1999) proposed to improve K-nearest Neighbor algorithm except that we have considered the classes of instances when defining the weights of features. In other