Machine Learning Project

Classification of news articles

Jakob Sahlström
jasa5691@student.uu.se

Anders Lindström
anli6945@student.uu.se

Jesper Dürebrandt
jedu6357@student.uu.se

Uppsala University

Friday 23rd May, 2014

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1 Introduction

2 Theory

2.1 Naive Bayes classifier

Let c be the class and $A = a_1, \dots, a_n$ be the attributes of a document. Then with Bayes Theorem

$$p(c|A) = \frac{p(A|c)p(c)}{p(A)}, \tag{1}$$

the attributes A is classified as class C if and only if

$$f_b(A) = \frac{p(c|A)}{p(\neg c|A)} \geq 1, \tag{2}$$

where $f_b(A)$ is called a *Bayesian* classifier. Assuming all attributes are independent given the class,

$$p(A|c) = p(a_1, \dots, a_n|c) = \prod_{i=1}^{n} p(a_i|c)$$

the final classifier can be written as

$$f_{nb}(A) = \frac{p(c)}{p(\neg c)} \prod_{i=1}^{n} \frac{p(a_i|c)}{p(a_i|\neg c)} \tag{3}$$

where f_{nb} is called the *Naive Bayesian* (NB) classifier and is a binary classifier,

Two models that uses the Naive Bayes assumption are the *multi-variate Bernoulli* model and the *multinomial* model. The main difference is that they are different assumptions regarding the distribution of $p(a_i|c)$. In the Bernoulli model the attributes are binary, indicating if a word from a vocabulary has occurred at least once or not. In the multinomial model the frequency of words are taken into account.

2.2 Random Forest

Random Forest (RF) is based on building several considerably small decision trees. Consider having a feature vector that is of length N, then randomly select $n << N$ of those features. Build the trees with some kind of algorithm (e.g. C4.5), where information gain is taken into consideration when splitting, and no pruning is done (i.e. expand the tree fully). Then repeat selecting n new variables from N until the wanted number of trees are built.

After all the trees are built, they can be used to let a new vector of data pass through all the trees and then letting each tree *vote* on what class the vector most probably should be a part of.

Algorithm 1 Random Forest

```
Let $x = (x_1, x_2, ..., x_N)$ be a set of features;
while not enough trees do
    Randomly pick with replacement a subset containing $n << N$ features;
    Use training set to build a decision tree using a classification algorithm, e.g. C4.5, except no pruning is done.
end while
```

2.3 Support Vector Machine Classifier

Support vector machine classifiers (SVM's) classifies data belonging to two classes by finding the hyperplane with the widest margin that separates the classes. The data vectors that restrict the margin of the hyperplane are referred to as suport vectors. This results in a maximization problem, where the objective function describes the width of the margin. This is solved using quadratic programming. An advantage with this approach is that the maximization problem is convex, meaning that the maximum found is guaranteed to be the global maximum. This requires, however, that the classes are linearly separable.

2.4 Multi-class problem

Classifiers like NB and SVM are binary classifiers, separating two classes. For a multi-class problem the strategy *One-vs-The-Rest* is used which creates one classifier for each target class. Then for each classifier fit the feature vector to that class vs not that class. The classifier with highest confidence will be the final class of the feature vector.

3 Method

3.1 In general

3.2 About Naïve Bayes

- Started with Naïve Bayes Classifier
- Downloaded about 200 articles to classify
- Divided into 2/3 training set, 1/3 test set
- Parsed the words, removed non-alphabetical letters, kept -
- Stemmed the words in the array
- Removed empty articles
- Removed stop words (such as "a","the" and so on.)
- Used information gain to sample a good feature vector smaller than the one used earlier.

3.3 About SVM

4 Results

Write some introducing text about result here.

4.1 Naive Bayes Classifier

Here goes specific results for Naïve bayes.

4.2 Support Vector Machine Classifier

Here goes specific results for SVM.

5 Related work

Maybe write some introducing text about related work here.

5.1 Naive Bayes Classifier

Naive Bayes models are widely used because of it's simplicity and efficiency. A. McCallum and K. Nigam compares the two most common models, the multi-variate Bernoulli and the multinomial model, in the realm of document classification. They are explained in detailed both theoretically and empirically and in general the multinomial model outperforms the Bernoulli model [7].

5.2 Random Forest Classifier

Random Forest is a decision tree based classification model. It has been used to classify web documents by keywords, where it for five and seven topics performed better than e.g. Naive Bayes and MLP [2]. In I. Kopriska, J. Poon, J. Clark,

J. Chan's paper, they use Random Forest for classifying e-mails. Random Forest was able to out perform other methods, such as DT, SVM and NB[3].

5.3 Support Vector Machine Classifier

Support vector machine classifiers performs well on data that is linearly separable and is guaranteed to find the optimal hyperplane that separates the data. They can however only separate data into two classes, but if combined they are able to perform multi-class classification. A simple approach is to use k SVMs to solve a k-class classification problem. The SVMs may also be combined in a more sophisticated fashion, so that less than k SVMs can be used. Both methods are investigated in [5].

6 Conclusions & Future work

- Fuzzy match example for improvement or testing.

References

[1] R. Berwick. An Idiots guide to Support vector machines, 2003.

[2] Myungsook Klassen and Nikhila Paturi. Web document classification by keywords using random forests, 2010.

[3] Irena Koprinska, Josiah Poon, James Clark, and Jason Chan. Learning to classify e-mail, 2006.

[4] V. Tampakas M. Ikonomakis, S. Kotsiantis. Text Classification Using Machine Learning Techniques, 2005.

[5] Eddy Mayoraz and Ethem Alpaydin. Support Vector Machines for Multi-class Classification, 1999.

[6] Andrew McCallum and Kamal Nigam. A comparison of event models for Naive Bayes text classification, 1998.

[7] Harry Zhang. The Optimality of Naive Bayes, 2004.