A Simple KNN Algorithm for Text Categorization

Pascal Soucy Guy W. Mineau
Department of Computer Science
Université Laval, Québec, Canada
{Pascal.Soucy, Guy.Mineau}@ift.ulaval.ca

Abstract

Text categorization (also called text classification) is the process of identifying the class to which a text document belongs. This paper proposes to use a simple non-weighted features KNN algorithm for text categorization. We propose to use a feature selection method that finds the relevant features for the learning task at hand using feature interaction (based on word interdependencies). This will allow us to reduce considerably the number of selected features from which to learn, making our KNN algorithm applicable in contexts where both the volume of documents and the size of the vocabulary are high, like with the World Wide Web. Therefore, the KNN algorithm that we propose becomes efficient for classifying text documents in that context (in terms of its predictability and interpretability), as will be demonstrated. Its simplicity (w.r.t. its implementation and fine-tuning) becomes its main assets for on-the-field applications.

1. Introduction

Text categorization (also called text classification) is the process of identifying the class to which a text document belongs. This generally involves learning, for each class, its representation from a set of documents that are known to be members of that class. This paper proposes to use a simple non-weighted features KNN algorithm to achieve this task. The simplicity of this algorithm makes it efficient w.r.t. its computation time, but also w.r.t. the ability for non expert users to use it efficiently, that is, in terms of its prediction rate and the interpretability of the results, as will be demonstrated below.

This paper presents a simple KNN algorithm adapted to text categorization that does aggressive feature selection. This feature selection method allows the removal of features that add no new information given that some other feature highly interacts with them, which would otherwise lead to redundancy, and features with weak prediction capability. Redundancy and irrelevancy could harm a KNN learning algorithm by giving it some unwanted bias, and by adding additional complexity. By taking into account both the redundancy and relevancy of features, we aim at providing solid ground for the use of KNN algorithms in text categorization where the document set is very large and the vocabulary diverse.

The KNN algorithm that we propose is presented in section 2. Our feature selection method is described in section 3. We tested the whole method on various corpora of text documents and we discuss the results of these experiments in section 4. In brief, it proved to be more accurate than Naive Bayes classifiers, and proved to be as accurate as other variants of the KNN approach but with a much smaller set of features, which helps the interpretability of the results and the speed of classifying new documents. Therefore we conclude that the KNN algorithm presented in this paper is highly relevant for the categorization of large corpora of text documents, and by its simplicity, could be easily implemented by non expert users.

2. Learning to categorize texts using KNN

Conceptually, each example document x, called an instance, is represented as a vector of length $|F|$, the size of the vocabulary:

$$<w_1(x), w_2(x), w_3(x), ..., w_{|F|}(x)>$$

where $w_j(x)$ is the weight of the jth term. That weight may be set according to different criteria, such as : frequency, *TFIDF* or a score assigned to the feature for its capability to divide the examples into some set of classes (e.g., the Information Gain). The simplest feature weighting function is to assign the same value to each term that occurs in the instance (let us say 1 for instance), and 0 to those that do not, which amounts to a non-weighted features approach. Our algorithm uses this latter simple weighting approach.

In our KNN algorithm, we used distance as a basis to weight the contribution of each k neighbor in the class assignment process. We define the confidence of having document d belonging to class c as:

0-7695-1119-8/01 $17.00 © 2001 IEEE

$$\text{Confidence}(c,d) = \frac{\sum_{k_i' \in K | (Class(k_i')=c)} Sim(k_i', d)}{\sum_{k_i \in K} Sim(k_i, d)} \qquad (1)$$

where *Sim* is the value returned by the similarity function used to compare the instance with its neighbors. That is, for each neighbor in the neighbor set *K* (of size *k*) belonging to a particular class *c*, we sum up their similarities to document *d* and divide by the summation of all similarities of the *k* neighbors with regard to *d*.

To compare document *d* with instance *i*, we choose the CosSim function which is particularly simple using our binary term weight approach :

$$\text{CosSim}(i,d) = \frac{C}{\sqrt{A * B}} \qquad (2)$$

where *C* is the number of terms that *i* and *d* share, *A* is the number of terms in *i* and *B* the number of terms in *d*. The neighbor set K of *d* thus comprises the k instances that rank the highest according to that measure.

3. Feature Selection

Feature selection is the process of selecting a subset of all available features. With text documents, as mentioned before, text categorization may involve hundreds of thousands of features, most of them being irrelevant. Relevancy of a feature in text categorization is particularly hard to define since there may be much feature interaction that keeps irrelevant features from being identified as such.

Our findings are: 1) that the entropy-based Information Gain metric is good at finding relevant features in terms of its capability to discriminate between classes, and 2) that removing every feature occurring only once does not increase the error rate and should be used as a fast filter before computing the Information Gain. These results are consistent with similar findings in [1,2]. However, these latter methods do not allow feature removal based on the interaction between features. In our KNN experiments, we used *μ-Cooccurrence*, a recent feature selection method described in [3]. That feature selection method reduces aggressively the vocabulary size using feature interaction.

4. Results and conclusion

The data sets used in our experiments are summarized in Table 1. Tasks accompanied by * are data sets we have manually created, while the others have been studied in the literature. Each task involves two classes.

We conducted tests using a naïve bayesian classifier, and we ran comparisons with our KNN algorithm, as shown in Table 2. The best Bayes results are obtained using many features, but not all, a good value being the 2000 best features selected by the Information Gain.

Task name	Dataset	Train/Test Balance
WebKBCourse	WebKB	80/320
ReutersCornWheat	Reuters-21578	40/360
LingSpam	Ling-Spam	40/80
Prisoner*	WWW	40/20
Beethoven*	WWW	40/20
News1*	Usenet	40/27

Table 1 : Data sets used in our experiments.

However, despite the fact that Bayes performs quite well for these tasks, KNN does better (for any number of features). From these results, we claim that our very simple KNN algorithm can reach impressive results using very few features.

Task name	μCooc + KNN		μCooc + Bayes		Bayes 2000 Best	All Bayes	
	#f	**%**	**#f**	**%**	**%**	**#f**	**%**
Course	35	96.9	35	95	95,6	12150	94.2
Reuters1	8	98.3	8	97.4	89,2	3393	81.1
Spam	77	95	77	86.3	90	4484	86.3
Prisoner	9	90	9	70	80	5076	70
Beethoven	8	85	8	90	85	3327	90
News	130	85.2	130	92.7	96,3	8522	92.6
McrAvrg	**95.5**		92.7		93.1	90.3	
# feature	**267**		267		12000	36952	

Table 2 : Results summary. #f is the number of features, while % is the classification accuracy.

Future works include the use of meta-information such as the structure of the document, author name, presence of highlights, figure or image included, etc. KNN appears to be particularly fit to include meta-level information into its classification mechanism.

References

[1] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos & P. Stamatopoulos. Learning to Filter Spam E-Mail : A Comparison of a Naive Bayesian and a Memory-Based Approach. In *Proceedings of Machine Learning and Textual Information Access workshop*, 4th Eur. Conf. on Principles and Practice of KDD (PKDD-2000).

[2] T. Joachims. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. *In Proceedings of the 14th International Conference on Machine Learning ICML97*. 1997.

[3] P. Soucy & G. Mineau. A Simple Feature Selection Method for Text Classification. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*. 2001.