Support Vector Machines based on a Semantic Kernel for Text Categorization

Georges Siolas, Florence d'Alché-Buc
Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie, boite 169
4, place Jussieu 75252 Paris Cedex 05 France
E-mail : georges.siolas, florence.dalche@lip6.fr

Abstract

We propose to solve a text categorization task using a new metric between documents, based on a priori semantic knowledge about words. This metric can be incorporated into the definition of radial basis kernels of Support Vector Machines or directly used in a K-nearest neighbors algorithm. Both SVM and KNN are tested and compared on the $20 - newsgroups$ database. Support Vector Machines provide the best accuracy on test data.

1 Introduction

Text categorization represents an interesting challenge to statistical inference and machine learning communities due to the growing demand for automatic information retrieval systems. Recently, a large number of learning methods [9] such as decision trees [12], naive Bayes classifier and Support Vector Machines [8],[6],[17],[5] have been successfully applied to this task. However, quite surprisingly even the most efficient learning systems (SVM) are based on rather simple representations of documents such as term frequency−inverse document frequency ($(TF - IDF)$ that only take into account the frequencies of the terms in indexed documents and in the whole collection. In some sense, during the last years, the efficiency of automatic tools was traded of against the lack of linguistics into the representation of documents. Based on the experience that prior knowledge always enhances performance of a system when appropriately introduced, we have investigated ways of introducing semantic knowledge into text classifiers and texts representation. We focused our study on the application of Support Vector Machines [2], [18] that present very good capacity to generalize and were already used with success on text categorization tasks. From our point of view, SVM has especially the advantage as other "relational learning methods" to be well-suited to incorporate a priori knowledge by the choice of an appropriate metric. Different works [14],[15],[3] have pointed out the impact that the choice of a metric has on the generalization capacity of classifiers. A simple way to define a new metric is to build one from the data themselves. For instance if the new metric is built from a linear transformation of the data, one can search for the best transformation matrix from the data themselves. Short and Fukunaga in [14] proposed to optimize the metric used in k-NN by determining it from the training data. A famous example of this approach was provided by Simard et al. who defined the tangent distance to compare digit images using the definition of set of linear small transformations that lead classes invariant. As reported by Vapnik in his book [19], the best performance on the US post office data base of digits were obtained for one-nearest neighbor with the tangent distance which shows the high relevance of the knowledge introduced there.

The method we develop for texts relies on the a priori mapping of the input vectors into a space more semantically relevant. Equivalently, as the transformation of the data is linear, it comes to modify the Euclidean metric that allows to compare to features vectors.

We will present in section 2 our semantic mapping we called semantic smoothing that is equivalent to the definition of a semantic metric. Section 3 will be devoted to the brief presentation of SVM with references

0-7695-0619-4/00 $10.00 © 2000 IEEE

to previous works on text categorization. In section 4, experimental results concerning both SVM and KNN will be discussed. In section 5, a conclusion and different openings to this work will be presented.

2 Incorporation of semantic into texts metric

Texts as features vectors

In text categorization [9], one has to assign categories to documents. A document can belong to several classes. When a corpus of documents and the corresponding desired categories are available, the task consists in solving separate binary supervised classification problems.Most of the learning systems rely on the following preprocessing of the data. First, words of the whole collection are stemmed and are selected to constitute an index. Usually this feature selection step is realized by the optimization of a mutual information criterion to provide the most relevant terms for the classification tasks. Once the index is available, each document is represented as a features vector where each feature is the value of the frequency of the corresponding term in the document multiplied by the inverse frequency of the term in the whole documents collection [13]. This representation mainly exploits and expresses the statistical properties of the document collection in terms of occurencies of symbols (terms) of a predefined alphabet (index). Whereas some methods use syntactical tags to characterize more precisely words [1], [17], very few methods have been proposed to take into account semantic categories of terms or semantic information. One successful attempt can be found in the FAQ finder system of [4] that makes use of a semantic similarity and a term similarity to select relevant answers to the requests.

One possible explanation of the encountered difficulties might be that the nature of semantic information is rather relational in opposition to the unique character of grammatical or syntactical information.

A semantic proximity defined for terms

The method we propose is based on the exploitation of the informations provided by Wordnet [11], a hierarchical semantic database of English words . In Wordnet, the main relations between words are the hypernym links which works as "is-a" relationships for nouns and adverbs, and synonym links defined for adjectives and adverbs. The length of the path between two terms indicates if two terms are semantically close or not. This information has already been used in the FAQ finder system[4] to define a semantic similarity. We also define here a semantic proximity but we will make another use of this function.

The *semantic proximity* between two words is defined in the [0,1] interval. If words are equal the proximity is equal to one.the words are different, the proximity between them is defined as the inverse of the length of the path (in the tree) linking the two words. When a word does not belong to this graphical dictionary (Wordnet), it is considered as separated from the other words by paths of half maximal length. Obviously this proximity does not covers all the aspects of the complex semantic relationships between words and also groups of words but provides a useful indication of semantic position of words among the others.

Semantic smoothing and metric Once the training corpus (documents with stemmed words) has been preprocessed in order to build an index of the most relevant terms, a proximity (symmetric) matrix P that reflects semantic relations between the index terms is built. Its dimension is thus the size of the index. Let us now consider the linear transformation defined by this matrix P.

Applied to $TF - IDF$ features vectors, it performs a kind of **semantic smoothing** of the vectorial data. Indeed terms that are semantically close to many others terms and are a priori strongly related to them, gain in importance (the corresponding feature value increases) while semantically isolated terms loose their importance.

For example, if the word "society" is highly present while the words "people" and "country" are less, the application of semantic smoothing will increase the values of the last two terms because "society", "people" and "country" are strongly related concepts. The new encoding of the documents is richer than standard tf-idf encoding since, in addition to statistical information, it embodies grammatical/syntactical and semantic information. In a sense, this linear transformation smoothes the vectorial representation using a semantic bias. Let us now consider the metric induced by this transformation on the data. For two vectors **x** and **y**, ordinary Euclidean metric is transformed into the following metric where the positive matrix S is defined as $S = P^2$.

$$\| P(x-y) \|^2 \quad = \quad (x-y)^T P.P(x-y) \tag{1}$$

$$\| P(x-y) \|^2 \quad = \quad (x-y)^T S(x-y) \tag{2}$$

This metric can be incorporated to the definition of a kernel in SVM.

3 Support Vector Machines

Support Vector Machines were introduced by Boser, Guyon and Vapnik in the seminal paper [2] and analyzed in [18]. SVM, in their general form, maps the input data into a new space using kernel function centered into support vectors and then makes a linear separation in the new space. Different symmetric functions can be used as kernels K : radial basis kernels and polynomial kernels for instance. Let us define $\{(x_i, y_i), i = 1...l\}$ the training sample for a binary classification problem. If the vector α and the scalar b are the parameters of the output hyperplane, f the SVM function is defined as :

$$f(x) = sgn(\sum_{i=1}^{i=l} \alpha_i.y_i.K(x, x_i) + b) \tag{3}$$

The induction principle derived to determine weights of output hyperplane (and support data) is the maximization of the margin between the output hyperplane and the data encoded in the hidden layer of the network. To deal with non separable data, the margin concept was softenized [18] in order to accept as SV some points that are on the wrong side of the margin frontiers.

One of the most interesting characteristic of SVM is that they belong to the class of learning machines that implement Structural Risk Minimization principle : this brings a justification to their very good performance on all the real-life problems so far tested. However, as all algorithms based on relational coding, the accuracy of the SVM highly depends on the definition of the relation, here the kernel function, that describes the "similarity" of two given examples.

Several authors have claimed the relevance of SVM for text categorization tasks [8], [6],[17], [5]. SVM are indeed well-suited to cope with high-dimensional data. Joachims in [8] showed that they do very well with sparse data and outperform other systems. But none of these works have take advantage of the possibility to add prior linguistic knowledge into kernels.

To implement our approach, we have chosen the radial basis kernel that usually gets very good performance with few tuning and which is still a reproducing kernel when a metric is used as the argument of exponential.

$$K(x, y) = exp(-\gamma \| x - y \|^2) \tag{4}$$

After semantically smoothing the vectors, we get

$$K(x, y) = exp(-\gamma \| (x-y)^T S(x-y) \|^2) \tag{5}$$

Other kernels based on the usual definition of similarity between two documents could be used as well, together with the semantic proximity matrix. Indeed, one great advantage of SVMs is the fact that in order to apply its powerful learning algorithm, it is sufficient that the matrix K of the kernel-transformed data is positive. Hence, it is possible to apply SVMs as soon as a positive matrix K can be defined over the data. This opens large areas of problems where SVMs could be applied as well studied in [16].

4 Numerical results

Data sets and preprocessing We tested semantic smoothing in the 20 newsgroups data set. The 20 newsgroups data set contains 20,000 Usenet articles partitioned in 20 thematic categories. Each category contains 1,000 articles and 4% of the articles are cross-posted. Results of text categorization experiments with this data set can be found in [7] [10]. We divide the initial multi-class categorization problem into 20 separate bi-class problems, each time considering the totality of a category's articles as the positive examples and an equal number of articles randomly chosen among all the other classes as the negative examples. In each case, we use 2/3 of the examples in the train set, while the remaining examples are put in the test set. The complexity of the calculation needed to generate the semantic proximity matrix limits the number of words that could be used as indexing terms. We performed tests with indexes of 150, 200, 250 & 300 words, selected by the highest mutual information criterion. Increasing the number of the indexing words did not led to significantly improved performances, so we settled with a 200-word index. A similar approach for selecting the number of features is used in [6] and a discussion of the feature size problem in text classification can be found in [10]. In general [5], [7], adding words into the index with low predictive power adds noise.

In using Wordnet, we notice that an important ratio of words were not present in the wordnet database (about 30 among 200). This was a source of deterioration of our results using the semantic metric that does not appear in the results.

Numerical results We evaluated the relevance of our semantic metric using two machines : a SVM using an exponential kernel with γ=0.001 and a k-nearest-neighbors.

The results are detailed in table 1. For each category, results are given as averages obtained from 5 random splits of the data set.

newsgroup	KNN	SD	S-KNN	SD	SVM	SD	S-SVM	SD
alt.atheism	73.91	1.76	83.20	2.57	86.64	2.10	87.84	2.84
comp.graphics	71.02	2.17	78.82	1.73	78.36	2.27	84.51	2.03
comp.os.ms-win.misc	71.72	1.97	76.87	2.01	82.86	1.84	89.92	2.97
comp.sys.ibm.pc.hardware	71.62	2.36	77.77	1.89	80.45	2.48	86.92	2.48
comp.sys.mac	75.67	3.12	76.71	2.22	84.96	2.44	90.68	2.62
comp.windows.x	61.61	2.22	76.31	2.72	85.74	2.96	87.84	3.01
misc.for sale	74.85	3.06	79.94	2.25	85.46	1.94	88.16	2.69
rec.autos	79.31	2.73	77.06	2.19	87.54	2.13	89.04	3.06
rec.motorcycles	80.63	2.20	76.42	2.12	93.68	2.92	91.58	2.34
rec.sport.baseball	67.26	2.63	81.31	1.82	90.12	2.09	90.57	2.14
rec.sport.hockey	65.56	1.76	87.87	2.07	93.7	3.12	92.5	2.34
sci.crypt	68.72	2.45	84.96	1.89	90.51	2.49	90.21	2.40
sci.electronics	73.01	2.54	75.11	1.72	79.88	2.48	81.08	2.11
sci.med	68.56	2.47	77.84	2.95	85.01	1.77	88.01	2.91
sci.space	70.97	2.52	80.75	2.61	88.1	1.73	85.99	2.37
soc.religion	68.66	2.94	82.15	3.06	90.39	2.01	93.09	2.12
talk.politics.guns	70.05	2.07	87.12	3.01	87.41	3.18	92.2	2.29
talk.politics.mideast	77.36	2.87	86.05	2.94	90.54	3.15	91.14	2.69
talk.politics.misc	64.96	1.97	80.15	2.30	81.33	1.93	83.28	2.07
talk.politics.religion	72.37	1.77	76.27	2.13	85.71	2.05	86.02	2.55

Table 1: Average accuracy and standard deviation (based on 5 random split of the data) for each newsgroup. S indicates the semantic metric.

method	accuracy	precision	recall	#sv
KNN	71.79	92.86	47.58	n/a
KNN+P	80.13	77.40	85.34	n/a
SVM	86.44	86.35	86.81	388
SVM + P	88.52	91.25	85.34	452

Table 2: Average scores for accuracy, precision and recall, and number of support vectors.

For both machines, the introduction of the new a priori metric increases the classification rate (accuracy). This empirically shows that semantic smoothing is relevant for text categorization tasks. SVM, with any metric, outperforms kNN and this confirms the supremacy of the induction principle used for SVM.
In table 2, average scores are presented for the 20 classes. Performance are not only measured in term of accuracy on test set but in terms of precision and recall, two classical criteria of IR community (information Retrieval) [9] Semantic SVM offers the best trade-off between recall and precision even if the best precision is obtained by the simple k-nearest-neighbors classifier and the recall of the semantic enhanced SVM is lower than the one of the simple SVM classifier. It should be also noticed that compared to previous works on the same database (learnt with slightly different protocols) [10],[7], the Semantic-SVM improves greatly the performance both in term of recall/precision and in term of classification rate (accuracy), while the index

size is amazingly small.

However a remark comes to temper these good results, the introduction of the semantic proximity matrix in the kernel increases the number of support vectors. The average number (for each of the 20 categories) is quite large in both approaches (with or without semantic). This could be explained by the fact that for each category problem, negative examples are taken from the 19 other categories and thus should be very different the ones from the others. Thus the extraction of support data cannot be very selective.

5 Conclusion

We propose to introduce semantic a priori knowledge into text processing in the framework of text categorization. The simple but efficient semantic smoothing we propose can be seen as a priori mapping of feature vectors to a semantically more suitable space or equivalently as a modification of the ordinary Euclidean metric. When incorporated into SVM kernels or k-NN , the semantic a priori improves significantly the performance while the size of the index is kept small. Moreover, in case of SVM, the results in terms of precision, recall, and accuracy appear to be very high. This comforts the idea that SVM provides an attractive framework to deal with real-life problems where domain knowledge is available. Furthermore, the framework of data support machines opens a new interesting perspective, encoding documents as proximity data rather than numerical vectors.

References

[1] M.-R. Amini, H. Zaragoza, and P. Gallinari. Stohastic models for surface information extraction in texts. In *Proc. of 9th International Conference of Artificial Neural Networks*, Edinburgh, UK, 1999.

[2] V. Vapnik B. Boser, I. Guyon. A training algorithm for optimal margin classifier. In *Fifth Annual Workshop on Computational Learning Theory.*

[3] A. Smola V.Vapnik B. Scholkopf, P. Simard. Prior knowledge in support vector kernels. In *Proc. of ICANN'96.*

[4] R. D. Burke, K. J. Hammond, V. A. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently-asked question files: experiments with the faq finder system. Technical Report TR-97-05, University of Chicago, 1997.

[5] H. Drucker, D. Wu, and V. N. Vapnik. Support vector machines for spam categorization. *IEEE transactions on neural networks*, 10(5):1048–1054, 1999.

[6] S. Dumais, J. Platt, J. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of ACM-CIKM98*, 1998.

[7] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In Douglas Fisher, editor, *Proceedings of the fourteenth international conference of machine learning.* Morgan Kaufman, 1997.

[8] T. Joachims. Text categorization with many relevant features. In *Proceedings of the 10th european conference on machine learning.* Springer Verlag, 1998.

[9] D. D. Lewis. *Representation and learning in information retrieval.* PhD thesis, University of Massachussetts, 1992.

[10] A. McCallum, R.Rosenfeld, T.Mitchell, and A.Y.Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the 15th international conference on machine learning*, 1998.

[11] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on wordnet. Technical report, Stanford University, 1993.

[12] I. Moulinier. *Une approche de la catégorisation de textes par l'apprentissage symbolique.* PhD thesis, Université Pierre et Marie Curie - Paris 6, 1997.

[13] G. Salton. Developments in automatic text retrieval. *Science*, 1991.

[14] R. Short and K. Fukunaga. The oprimal distance measure for nearest neighbor classification. *IEEE Trans. on Information Theory*, 1981.

[15] P. Y. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new tranformation distance. *Neural Information Processing Systems*, 5:50–58, 1993.

[16] B. Schlkopf A. Smola P. Bartlett K.-R. Muller K. Obermayer R. Williamson T. Graepel, R. Herbrich. Classification on proximity data with lp-machines. In *Proc. of ICANN'99.*

[17] H. Taira and H. Masahiko. Text categorization using support vector machines. In *IPSJ SIG-NL*, number 98-NL-128-24, pages 173–180. American association for artificial intelligence, 1998.

[18] V. Vapnik. *The nature of statistical learning.* Springer, 1995.

[19] V. Vapnik. *Statistical learning theory*, chapter 12. John Wiley & Sons, 1998.