Kernel PCA Based Clustering for Inducing Features in Text Categorization

Zsolt Minier[1] and Lehel Csató[1] *

1- Babeş-Bolyai University - Department of Mathematics and Computer Science
RO-400084, Cluj-Napoca, Romania

Abstract. We study dimensionality reduction or feature selection in text document categorization problem. We focus on the first step in building text categorization systems, that is the choice of efficiently representing numerically the natural language text. This numerical representation is going to be used by machine learning algorithms. We propose a representation based on word clusters.

We build a kernel matrix from the word distribution over the different categories and apply kernel PCA to extract a low-dimensional representation of words. On this low-dimensional representation we use K-means clustering to group words into clusters and use these clusters subsequently in the document categorization task. We show that kernel PCA based clustering gives better or comparable performance than several advanced clustering methods when applied for the standard Reuters corpus.

1 Introduction

Text categorization is an interesting problem in natural language processing that is solved using machine learning methods [1]. The problem at hand is a standard classification problem: given a set of documents written in natural language (English most of the time) and a small set of category labels, learn the assignment of documents to categories given the examples in the training set. Testing algorithm is done by assigning category labels to previously unseen documents. We know the correct labels from the documents; we use the ModApté split [2] of the Reuters corpus.

The categorization task can be divided into two sub-problems: (a) the representation of text written in natural language as data suitable for machine learning algorithms and (b) categorizing the transformed data. We are more interested in the first issue since we believe that the commonly used bag of words model [1] oversimplifies the document: it reduces them to purely word frequency counts, removing any semantic information present in the text.

An obvious method of capturing more semantics of the text is to cluster the words into semantically related groups. Each word in a group will be accounted as a *feature* identifying the respective group of words. By reducing the number of features to the number of word clusters we improve the frequency measures of related notions: similar words will be grouped into the same cluster.

This grouping benefits in lowering the sensitivity of the document representation to synonymous words that make related documents seem unrelated in

*The research was partly supported by the grant CEEX/1474 of the Romanian Ministry of Education and Research.

the classical bag of words model. Another important benefit of clustering is the reduction of document vectors from the tens of thousands to the hundreds, an important gain in storage space.

We employ the kernel PCA method to find a low-dimensional representation of the words [3]. Since the number of words is too high, we first perform a selection and reduce the kernel matrix to ≈ 3500 rows and columns. We extract eigen-directions as coordinate basis in the nonlinear space corresponding to the kernel we use. The clusters of words are identified over this eigen-space using the K-means algorithm.

The rest of the paper is structured as follows: in the next section we outline the existing approaches to word clustering for text categorization. Section 3 presents our method detailing each phase of the algorithm. Section 4 includes the experimental results of our tests and the last section concludes and enumerates further research directions.

2 Previous work

The earliest methods for reducing the size of the bag of words model were term ranking methods in which the distribution of each word over the document classes is used to decide whether the word is useful or not for text categorization. Yang and Pedersen [4] compare such methods and find that the χ^2 statistic is one of the best predictors of usefulness for words. These methods are very simple and yet very efficient because the needed computation time is linear in the number of words.

Clustering words for text categorization was first used by Baker and McCallum [5] and tested successfully for the Naive Bayes classifier. There are numerous other methods for clustering words used in text categorization, we implemented some based on the information bottleneck principle for clustering dyadic data [6]. The first one is from Dhillon et al. [7] and the second is due to Bekkerman et al. [8].

We present a different approach to clustering; based on the kernel PCA algorithm developed by Schölkopf et al. [3]. With this method, we try to reduce the dimension of word representation space which makes it easier to cluster words based on semantic similarity. This idea is also the starting point of the spectral clustering method of Ng et al. [9].

3 The proposed method

The goal of the method is to group words into hard clusters. This can be formalized as

$$W = \cup_{i=1}^{l} W_i \text{ and } W_i \cap W_j = \emptyset, \quad i \neq j$$

where W is the set of words and there are l word clusters represented by W_i.

A straightforward solution would be clustering the word frequencies over categories using the K-means algorithm. Our approach first transforms the word representations into a space that is smoother than the original document class frequency space. This is done by first applying the kernel PCA algorithm [3]

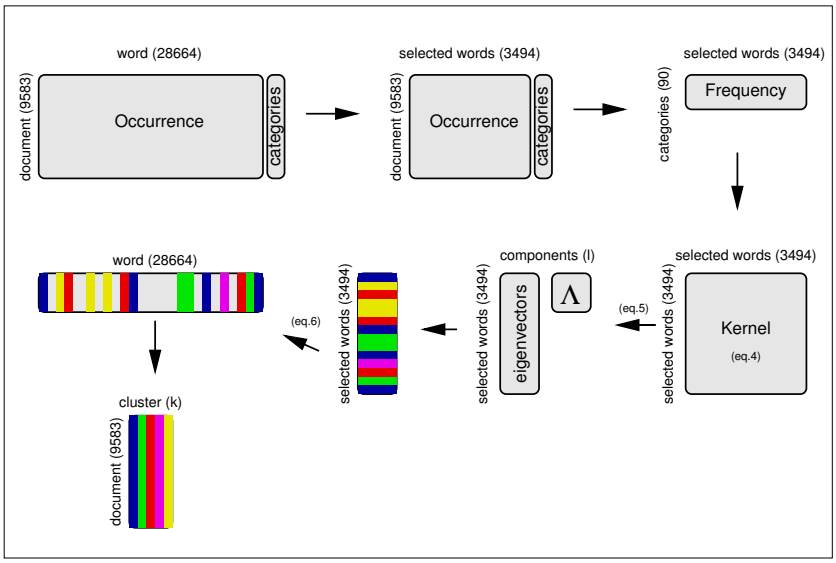

Fig. 1: The scheme of the algorithm.

and thus transforming each word into the rows of the eigenvector matrix (whose columns are the eigenvectors corresponding to the most important eigenvalues). These word representations of low dimensionality are then clustered with the k-means algorithm to find the word clusters. A schematic description of the algorithm can be seen in Fig.1, next we detail the steps of the algorithm.

As we found 28664 (no stemming and only punctuation and number removal) words in the corpus used, this would lead to a far too big kernel matrix and we propose a simplification of the problem. As most of the words in the corpus have very low frequency, we select the ones which occur more than 30 times and are not stopwords. This results in 3494 words. The kernel matrix is built only with these words.

Then, the kernel PCA algorithm is performed over the selected words. Each word is represented as a frequency vector over the categories of the corpus $w_i = (p_{i1}, ..., p_{i|C|})$ where $|C|$ is the number of categories. In the case of the ModApté split of the Reuters corpus this results in vectors of dimension 90. The kernel matrix K is computed by computing the kernel function over each pair of words $K_{ij} = k(w_i, w_j)$. Possible kernel functions are:

$$k(w_i, w_j) = w_i \cdot w_j = \sum_k w_{ik} w_{jk} \quad (1)$$

$$k(w_i, w_j) = (w_i \cdot w_j + 1)^p \quad (2)$$

$$k(w_i, w_j) = \exp\left\{-\frac{\|w_i - w_j\|^2}{2\sigma^2}\right\} \quad (3)$$

For the kernel matrix to be appropriate for principal component analysis, it has to be centered. This can be achieved by the following transformation, as described by Schölkopf et al. [3].

$$\tilde{K} = K - 1_m K - K 1_m + 1_m K 1_m \tag{4}$$

where $(1_m)_{ij} = \frac{1}{m}$ and m is the number of words.

The principal components of the kernel matrix K are calculated using the jdqr package [10] and are returned as the diagonal matrix D containing k of the largest eigenvalues and as the matrix V having as columns the corresponding eigenvectors.

$$K_{(m\times m)} \approx V_{(m\times k)} D_{(k\times k)} V^T_{(k\times m)} \tag{5}$$

The number of eigenvalues (denoted l on Fig.1) is determined by the rank of the kernel matrix. In our case 22 eigenvalues exist for the kernel matrix constructed with the linear kernel, 93 eigenvalues with the RBF kernel and 26 eigenvalues with the polynomial kernel of rank 2.

After performing PCA, the rows of the matrix formed by the eigenvectors represent the words. These word representations are clustered using the K-means algorithm in order to find the semantically related ones. The number of clusters is decided empirically.

As the rest of the words make 69.29% of the training data, they are assigned to clusters. This is not an easy task to solve optimally. We simplify the problem by assigning them to the cluster whose center is closest to the respective word by using Euclidean distance in the original space of word distributions over categories.

$$w_i \rightarrow W_j \text{ where } j = \min_k \left(\left\| w_i - \sum_{w_l \in W_k} \frac{w_l}{|W_k|} \right\| \right) \tag{6}$$

The centers of the clusters are not updated after each word is added to them.

This way every word in the corpus is assigned to a cluster. Transforming the documents into this representation consists of creating vectors for each document representing the number of words found in the document that belong to each cluster. These vectors are then transformed by a tfidf function [1] and then normalized.

4 Results

To compare this method to the methods found in the literature, we implemented the clustering methods of [7, 8], the spectral clustering method of [9] as well as the χ^2 term ranking method of [4]. The testing corpus used was the ModApté split of the Reuters corpus. The LibSVM library [11] was used for classification. The common performance measures of a text categorization system are precision and recall, and their harmonic mean, the F_1 measure (see [4]). The breakeven point also provides a good performance metric and is used throughout the literature (see [8] for detail). It is defined as the value of precision of the system when

method	nr	mP	mR	mBEP	mF1	MP	MR	MBEP	MF1
χ^2	5209	88.46	84.59	86.52	86.48	71.61	61.25	66.43	66.02
frequency	3494	74.19	69.63	71.91	71.83	47.86	40.00	43.93	43.57
linear kernel	50	82.48	78.09	80.29	80.04	41.90	36.65	39.28	38.67
RBF kernel	100	83.55	79.64	81.60	81.54	50.07	44.15	47.11	46.92
poly 2 kernel	50	84.67	79.40	82.04	81.95	45.35	37.68	41.52	41.16
spectral clustering	200	82.22	77.35	79.78	79.71	47.92	37.33	42.63	41.96
Dhillon et al.	175	80.80	77.00	78.90	78.85	51.22	42.80	47.01	46.63
Bekkerman et al.	125	82.22	77.35	79.78	79.71	46.87	38.16	42.51	42.06
K-means	75	79.82	75.16	77.49	77.41	43.95	36.33	40.14	39.77

Table 1: Results obtained for the Reuters corpus given in percentage. Notation: nr=number of words in the first two rows and number of clusters in the rest, mP=micro-precision, mR=micro-recall, mBEP=micro-breakeven, mF1=micro-$F_\mathbf{1}$, MP=macro-precision, MR=macro-recall, MBEP=macro-breakeven, MF1=macro-$F_\mathbf{1}$

precision equals recall or if this value does not exist, then it is defined as the algebraic mean of the two when they are closest to each other.

The results are shown in Table 1. In the first row we show the results of the χ^2 term ranking method (86.52% mBEP) still unmatched by the other methods we implemented. The second row shows the results of the term frequency ranking method (71.91% mBEP) with exactly the 3494 terms that are selected to build the kernel matrix. In the next three rows we give the best results of our method with different kernels in ascending order of performance: linear, RBF and *polynomial with rank 2* that achieves the highest micro averaged BEP among all: 82.04%. The subsequent rows show the results of other clustering methods: the spectral clustering method of Ng et al. [9], the Information Bottleneck implementations of Dhillon et al. [7] and Bekkerman et al. [8]. In the latter case we did not perform any tuning of the ν, β_{min} and β_{max} parameters, meaning that probably better performance could have been achieved in a cross-validation setting. The last row shows the results of the straightforward K-means clustering of words based on the frequency matrix, given as baseline.

5 Conclusions and Future Work

We showed that the presented kernel PCA based clustering method performed comparable with the other clustering methods we implemented. This is probably due to the ability of the kernel methods to effectively reduce the dimensionality of the representation space of words. We comment also on the superior performance of the χ^2 method compared to the other more sophisticated methods: the good performance of the χ^2 term ranking method can be due to the specifics of the Reuters corpus. As Bekkerman et al. [8] also note, most of the big categories of the Reuters corpus can be identified based on only a very few keywords (for example the "earn" category comprising of 1087 test documents can be identified

with 87% precision based on the appearance of the single "vs" token). This property does not favor clustering because the other words assigned to the cluster of "vs" appear as noise and degrade performance.

A direction of further exploration is to test these methods on the 20 Newsgroups corpus to compare the methods on more evenly distributed data. Another direction is the combination of kernels [12] in order to be able to better interpret the dimensionality reduction of word representations.

Analyzing the relationship to spectral clustering could lead to fruitful insight to the working of the algorithm. Another algorithm based on similar ideas is the Latent Semantic Kernel and Gram-Schmidt Kernel method of Cristianini et al. [13] which could be explored within the settings of this method.

References

[1] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[2] Chidanand Apté, Fred J. Damerau, and Sholom M. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

[3] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[4] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pages 412–420, 1997.

[5] L. Douglas Baker and A.K. McCallum. Distributional clustering of words for text classification. In W. Bruce Croft, Alistair Moffat, Cornelis J. Van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998. ACM Press, New York, US.

[6] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In B. Hajek and R. S. Sreenivas, editors, *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377. University of Illinois, 1999.

[7] I. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, March 2003.

[8] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208, 2003.

[9] A.Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[10] D.R. Fokkema, G.L.G. Sleijpen, and H.A. Van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM J. Sci. Comput.*, 20(1):94–125 (electronic), 1999.

[11] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines (version 2.31), September 07 2001.

[12] S. Sonnenburg, G. Rätsch, C. Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[13] N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2/3):127–152, 2002. Special Issue on Automated Text Categorization.