Combinatorial PCA and SVM Methods for Feature Selection in Learning Classifications (Applications to Text Categorization)

Andrei V. Anghelescu
Dept. of Computer Science
Email: angheles@cs.rutgers.edu
Phone: 1-732-445-4578

Ilya B. Muchnik
DIMACS
Email: muchnik@dimacs.rutgers.edu
Phone: 1-732-445-0073

Rutgers, The State University of New Jersey
110 Frelinghuysen Rd., Piscataway, NJ 08854-8019

Abstract— In this paper we describe a purely combinatorial approach of obtaining meaningful representations of text data. More precisely, we describe two different methods that materialize this approach: we call them combinatorial principal component analysis (cPCA) and combinatorial support vector machines (cSVM). These names emphasise mathematical analogies between the well known PCA and SVM, on one hand, and our respective methods.

For evaluating the selected spaces of features, we used the environment set for TREC 2002 and used a very common classifier: 1-nearest neighbour (1-NN). We compared the results obtained on the feature sets calculated by the procedures we described (cPCA and cSVM) with the results obtained on the original feature space. We showed that by selecting a feature space on average 50 times smaller than the original space, the performance of the classifier does not decrease by more than 2%.

1. INTRODUCTION

In text stream analysis one of the main problems is finding an effective method to classify documents fast and correctly. This is the reason why dimensionality reduction and related methods of representation of significant information are critical to developing a good text classifier. In such applications, the dimensionality of the term space may be problematic. The classification accuracy degrades with higher dimensions [1], [2]. Recently, it has been found that sophisticated text classification algorithms, such as SVM, known to generally scale well with the dimension of the representation space, also lose accuracy with high dimensions of the data space [3].

Prior work in the areas of feature selection and dimensionality reduction includes principal component analysis to find orthogonal dimensions in the space of documents [4], [5], as well as document clustering techniques to estimate feature strength [6].

In this paper we describe a purely combinatorial approach of obtaining meaningful representations of text data. More precisely, we describe two different methods that materialise this approach: we call them combinatorial principal component analysis (cPCA) and combinatorial support vector machines (cSVM). These names emphasise mathematical analogies between the well known PCA and SVM, on one hand, and our respective methods.

Using our cPCA method, one can divide features of the data (columns of the data matrix) into groups according to their degree of contrast (feature values are frequences of words in documents in the data set). The ordering of these groups along the scale of contrast constitutes the analogy with the classical PCA. In particular, this ordered scale of degrees of contrast plays the role of spectrum of eigen-values and their associated groups of features represent the eigen-vectors (considered as boolean vectors, which define the groups). Because this analysis does not use label information, there is no need of a validation test for the determined feature sets. These sets are to be used in designing a classifier, which, of course, has to be validated.

The cSVM application is very different in nature from cPCA. Firstly, the method makes use of the labels of documents. Secondly, the two stages that compose the method are also very different: in the first we filter documents (in contrast with cPCA which orders features), determining the most significant ones; the second step consists of feature ranking based on training a linear SVM, followed by feature selection (the SVM is trained using the filtered training data).

It is interesting to note that, in spite of obvious differences, the two methods share the same formal model. For both data is represented as rectangular matrix of non-negative values and all its sub-matrices are analysed using a particular score function. This function is defined over the set of all these

sub-matrices and both procedures find the unique sub-matrix that yields the maximum of the score function. This common mathematical core joins these two methods and is the reason to present them together in our paper.

The paper is structured into 5 sections, of which this introduction is the first. The second section contains the description of the cPCA method. In the third section we describe the cSVM method and emphasise the commonalities of the methods. The environment used for experiments and the obtained experimental results are presented in the fourth and fifth sections, respectively. The conclusions are presented in the last section.

2. COMBINATORIAL PCA

We start from the assumption that, given points from the same class, features that occur frequently in sufficiently many of these points are bound to be significant to the classification. From each class we select these significant features and use them to represent the points in this much smaller space.

The data is represented as a matrix, $A = \|a_{ij}\|$, where rows correspond to points (documents) and columns to features (frequences of words). It is clear that the elements of A are non-negative. In order to determine significant sub-matrices of A, we use as the set W the set of indices of rows and columns in the entire data matrix. Below we use the following notations: $W = (W_r, W_c)$, where W_r is a set of indices for rows of the matrix A, W_c is a set of indices for columns of the same data-matrix ($|W_r| = n$, $|W_c| = f$, $n + f = N$). $H = (H_r, H_c)$ is a subset of W ($H_r \subseteq W_r$, $H_c \subseteq W_c$). We define, additionally, that $\pi(i, H) = 0$ if H doesn't include at least one element from both sets W_r and W_c (in other words, if $H_r \neq \emptyset$ and $H_c \neq \emptyset$):

$$\pi(i, H) = \begin{cases} \sum_{j \in H_c} a_{ij}, & \text{if } i \in H_r \\ \sum_{j \in H_r} a_{ij}, & \text{if } i \in H_c \\ 0, & \text{if } H_r = \emptyset \vee H_c = \emptyset \end{cases} \quad (1)$$

The sub-matrix-cluster that we are looking for is defined by the subset of indices H^* which gives the maximum value for the function

$$F(H) = \min_{i \in H} \pi(i, H). \quad (2)$$

Definition 1
A set $H_0 \subseteq W$ is a strict local maximum of $F(H)$ if

$$\forall H, \ H_0 \subset H \subseteq W \ \Rightarrow \ F(H_0) > F(H)$$

The method for determining all local maxima is described in [7]. The H_c parts of these local maxima form the combinatorial principal components. These parts have the following interesting properties:

1) they form an ordered chain of sets:

$$H_1 \supset H_2 \supset \cdots \supset H_p$$

2) this order is mirrored in the sequence:

$$F(H_1) < F(H_2) < \cdots < F(H_p)$$

with $F(H_p)$ being the global optimum of $F(H)$ over all subsets of W.

Using the sequence of local maxima, $H_1, H_2, ..., H_p$, one can easily test different extreme sub-matrices with different *levels of word frequence* (in our experimental work we used such levels that contained a subset of 100-300 features).
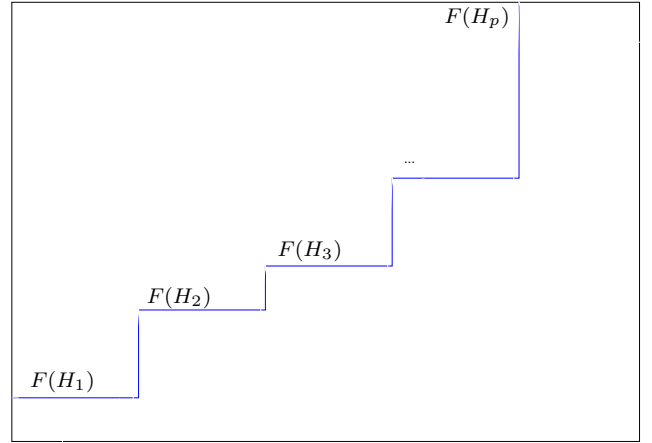

Figure 1 – Effective values of weights

Let us consider the following synthetic example. The clusters are sub-matrices of data which include each other according to the ordering given by the clustering model: the deepest cluster-matrix represents *the largest weighted quasi-clique* if the input data-matrix would be interpreted as a hyper-graph; its *effective weight* is also the largest possible; the second cluster includes the first one and represents the second level of a quasi-clique with less value of the effective weight in it, etc. The effective weight is used as the objective function whose optimisation gives the above clustering structure for the data. Figure 1 shows usual changes of effective values of weights along the above mentioned stratification (for one of the data-matrices used in our analysis. In Figure 2 we show an example of a small matrix and its sub-matrices-clusters found by our method.

When representing documents in a matrix where each row corresponds to a document and each column corresponds to a feature, it is clear that this tower structure gives an ordinal scale for both documents and their features. The scale of documents points to which documents contain the most frequent words (of course, after having filtered stop-words), and, which include really rare words; similarly, a related
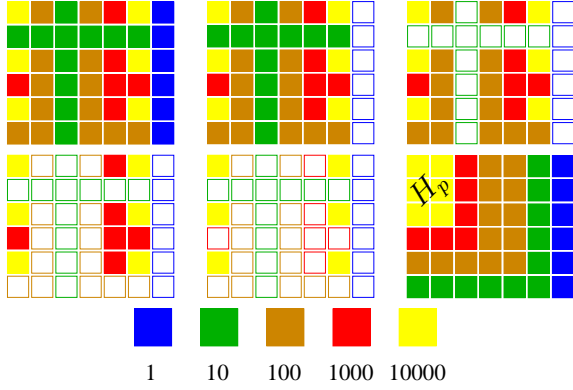
Figure 2 – Structure of documents and features

feature scale shows which words are most frequent, and, what is their "location" in documents. In the present study we used for the purpose of learning, only the ordinal scale for document features. As an improvement, we plan to use a similar scale for documents in the near future.

So, if one gets a chain of nesting set of words (parts of our nesting clusters) presented in the considered data-matrix, one can follow the order of the chain and interpret any position (subset of the corresponding words) as a particular subspace in which any classifier program can work. Of course, the most interesting case for us would be if we could construct a good classifier using a low dimension subspace. The fact that we can search candidates of those subspaces in the constructed chain provides a very efficient way to search the candidate subspaces.

3. COMBINATORIAL SUPPORT VECTORS

The second method, combinatorial support vectors, addresses the problem of feature selection by using a two-step strategy, which involves, at first, selecting a subset of significant documents followed by a feature selection based on information obtained from these documents. In consequence, the working data set will be the set of documents, identified in section 2 by its set of indices, W_r

All notations used in section 2 are carried to this section and, additionally, we introduce: the set of positive-labelled documents

$$X^+ = \{x_1^+, \ldots, x_i^+, \ldots, x_{n+}^+\}, \ |X^+| = n+$$

with its associated set of indices of positive-labelled training points,

$$W_r^+ = \{1, \ldots, i, \ldots, n+\}, \ |W_r^+| = n+$$

and analogously, the sets of negative-labelled training points and their indices, respectively:

$$X^- = \{x_1^-, \ldots, x_j^-, \ldots, x_{n-}^-\}, \ |X^-| = n-$$
$$W_r^- = \{1, \ldots, j, \ldots, n-\}, \ |W_r^-| = n - .$$

We also have that $W_r = W_r^+ \cup W_r^-$, $|W_r| = |X| = n$ and $n = (n+) + (n-)$.

The data is represented in a matrix, denoted by K, defined as:

$$\forall i \in \overline{1, n+}, \ \forall j \in \overline{1, n-} \ \ K_{ij} = e^{-\alpha d^2(x_i, x_j)},$$

where $d(x_i, x_j)$ is the Euclidian distance between x_i and x_j in the original feature space and α is the average similarity between the training points:

$$\alpha = \left(\frac{2}{n(n-1)} \sum_{\substack{k, l = 1 \\ k > l}}^{n} d^2(x_k, x_l) \right)^{-1}$$

In the matrix K we defined the similarity of a point with a set of points of opposite class as being:

$$\pi'(i, H_r) = \begin{cases} K_{i \cdot} = \sum_{j \in H_r^-} K_{ij}, & \text{if } i \in H_r^+ \\ K_{\cdot j} = \sum_{i \in H_r^+} K_{ij}, & \text{if } j \in H_r^- \\ 0 & \text{if } H_r^+ = \emptyset \vee H_r^- = \emptyset \end{cases}$$
(3)

with the remark that $\pi'(i, H_r)$ can be positive only if both the subset of positive and the subset of negative indices of H_r are not empty.

As performance criterion we used

$$F(H_r) = \min_{i \in H_r} \pi'(i, H_r) = \min \left(\min_{i \in H_r^+} K_{i \cdot}, \min_{j \in H_r^-} K_{\cdot j} \right)$$
(4)

which yields as solution of the problem the set:

$$\left(H_r^{+*}, H_r^{-*} \right) = \arg \max_{H_r = (H_r^+, H_r^-)} F(H_r)$$

The solution of the combinatorial problem is a set of indices, (H_r^{+*}, H_r^{-*}), which is uniquely associated with the set of points (X^{+*}, X^{-*}).

Since the elements of K represent point similarities, the geometrical interpretation of the resulting set (X^{+*}, X^{-*}) is that it contains the positive and negative points which are closest to the opposite class. We illustrated this concept in Figure 3. This represents a new model of the margin region of the training set X. We call X^{+*} and X^{-*} the set of positive and negative combinatorial support vectors, respectively.

Subsequently, the sets of combinatorial support vectors are used as training data for building a linear SVM. The support vectors found in the process of training the SVM determine the optimal hyper-plane of separation which reflects the weights of features, creating a scale onto which the features are ranked.

TABLE 1 – Classification results of 1-NN in the original feature space and those generated by cPCA and cSVM. The rows **avg** and σ_{avg} present the micro-averages (respectively standard deviations) of sensitivities and specificities obtained from classifying in the three examined feature spaces. The row $\overline{\sigma}$ presents the macro-average standard deviation obtained from running 5-fold cross-validation experiments multiple times.
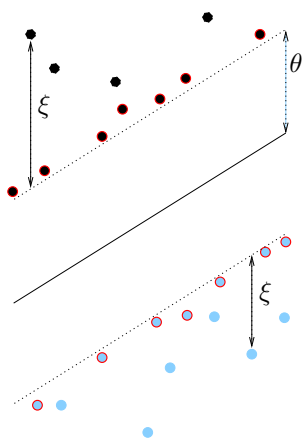
Topic	OrigSpace		cPCA		cSVM	
	sensitivity	specificity	sensitivity	specificity	sensitivity	specificity
avg	0.526	0.942	0.510	0.950	0.629	0.811
σ_{avg}	0.204	0.080	0.199	0.047	0.144	0.094
$\overline{\sigma}$	0.062	0.012	0.065	0.012	0.068	0.068
R101	0.833	0.967	0.839	0.970	0.854	0.861
R102	0.685	0.930	0.602	0.834	0.744	0.788
R103	0.554	0.987	0.406	0.962	0.574	0.788
R104	0.686	0.971	0.623	0.932	0.779	0.830
R105	0.692	0.957	0.692	0.956	0.801	0.787

Assuming the availability of a procedure for training a SVM, called SVM-Train, the process of ranking and selecting features is described in algorithm 1

The result of algorithm 1 is a set of ranked features from which the user selects the top S_0, as dictated by the application.

Until the end of this section, we will comment on conceptual similarities between the two proposed methods, cPCA and cSVM. For both methods, the underlying theoretical work intensively uses the theory of monotonic systems. This implies that objects are not used independently, but strongly as elements of system, interacting with each other and, thus, are important with respect to the other data.

The subset of data yielding the optimal value for the score function, F, can be seen as a central structure of the data, with all local minima as layers wrapped around this structure, with

---

Algorithm 1 Feature ranking and selection

Require:

$$X_0 = X^{+*} \cup X^{-*} = \{x_1, \ldots, x_l\}^T, \ |X_0| = l$$
$$y = \{y_1, \ldots y_l\}^T$$

S_0 - the desired number of features

$\alpha \leftarrow \text{SVM-Train}\,(X_0)$
$w \leftarrow \sum\limits_{k=1}^{l} \alpha_k y_k x_k \qquad // \ w = (w_1, \ldots, w_f)$
$w_s \leftarrow \text{sort}(w_1^2, \ldots, w_f^2)$
$r \leftarrow w_s(1 : S_0)$

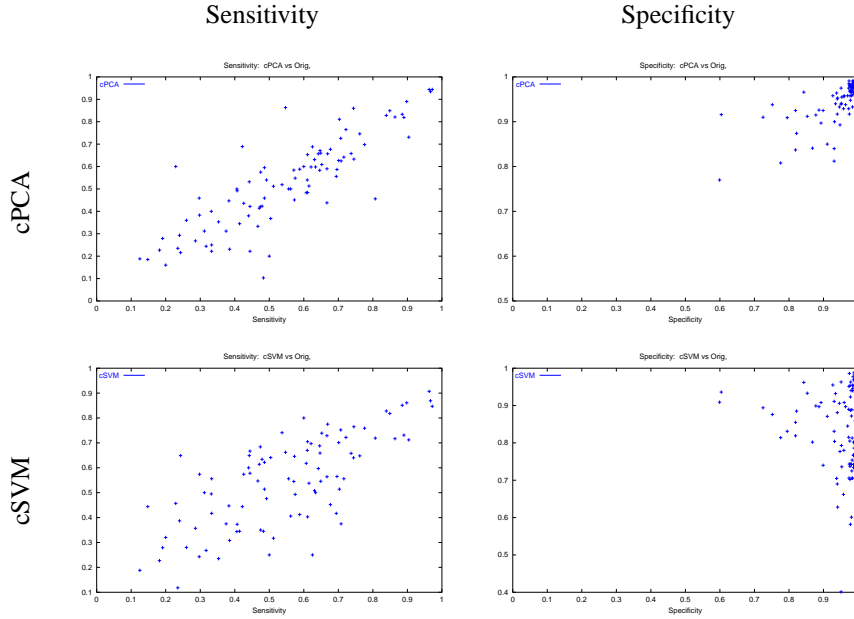**Ensure:** Feature ranked list r

---

the number of centres determined at run-time.

The consequence, also illustrated in this paper, is that the theory of monotone systems provides means of complete and multi-faceted analysis of data.

Although conceptually the two methods work on different representations of the data, there is a strong sense of complementarity: the combinatorial PCA works directly on features, while the combinatorial SVM focuses initially on determining the significant points.

The analogy between cPCA and cSVM can also be observed from the similarity of equations (1),(2) and (3),(4) respectively. Indeed, the only difference between them arises from the data they are defined on. In particular, $\pi(i, H)$ uses the matrix A, containing feature frequences, while $\pi'(i, H_r)$ uses a transformation of this data, by means of a radial basis function. As it can be observed, one method can be transformed into the other simply by substituting W with W_r and a_{ij} with K_{ij}. In this respect, the methods are similar analyses of the data set, from different points of view.

PSfrag replacements

Figure 3 – Example of combinatorial support vectors. The combinatorial support vectors are denoted by the points decorated with a red border. The margin of the presented data set is found between the dotted lines and has a width of $2 \cdot \theta$.

Sensitivity   Specificity

Figure 4 – Scatterplot of sensitivities and specificities of 1-NN in cPCA, cSVM vs. 1-NN in Orig

4. DESCRIPTION OF THE EXPERIMENTAL ENVIRONMENT

The input data consisted of the Reuters Corpus Volume 1 [8], containing the Reuters newswire stories from August 20, 1996 through August 19, 1997.

The system was tested on 94 topics obtained from TREC 2002 [9], selected from the total of 100 TREC topics available, such that each class (relevant, non-relevant) was described by at least 10 examples. On average, each topic was described by 80 positive and 750 negative examples. The union of all words in the documents describing a topic contained on average approximately 20,000 words.

The classification was performed in the feature spaces calculated by cPCA and cSVM, as well as in the original space, for comparison purposes. The classifier used was 1-NN. The classification results were evaluated using 5-fold cross-validation. The data was split using stratified random sampling. Each cross-validation experiment was run 5 times, each with a different random split of the data. We set the number of features selected by cPCA and cSVM to be, on average, 350, thus a reduction in dimension by a factor of 50.

The performance of the system was evaluated based on the confusion matrix calculated on the available labelled testing data. In this matrix, we defined the following quantities:

TP   the number of correctly identified positives
FN   the number of mis-classified positives
TN   the number of correctly identified negatives
FP   the number of mis-classified negatives

As performance measures we used the **sensitivity**(ζ) and **specificity**(η), defined by the formulae:

$$\begin{aligned} \zeta &= \frac{TP}{TP+FN} \\ \eta &= \frac{TN}{TN+FP} \end{aligned}. \quad (5)$$

In table 1 we provide a comparative perspective on the performances of 1-NN in the three spaces used in our experiments. The results are presented by topic for the first 5 topics (rows R101 to R105) and for each feature space the obtained sensitivity and specificity are provided. The feature spaces are designated by Orig, for the original space, cPCA and cSVM for those generated by the two methods presented in this paper. The complete results are available presented in [10]. The first row in table 1 presents the macro-average of the estimators, ζ and η. That is, for each topic we calculated the average sensitivity and specificity (as resulting from the 5 cross-validation runs) and we averaged these topic-specific averages. The second row of the table 1 presents the standard deviation of the average results presented in the first row. The third row of the table 1 presents the macro-average standard deviation: for each topic we calculated the standard deviations of the estimators obtained from the 5 experimental runs, then we averaged them over all topics.

In figure 4 we present a scatterplot of the results obtained in the cPCA and cSVM spaces, versus the results obtained in the original space. It can be observed that 1-NN tends to performs better in the cPCA space than in cSVM and in both it performs a little worse than in the original space.

5. Conclusions

In this document, we presented two combinatorial methods for dimensionality reduction. They offer simple means of determining a scale on which data points and features can be ordered, thus making possible a linear search for the optimal set of features or documents.

For evaluating the selected spaces of features, we used the environment set for TREC 2002 and used a very common classi?er: 1-nearest neighbour (1-NN). We compared the results obtained on the feature sets calculated by the procedures we described (cPCA and cSVM) with the results obtained on the original feature space. We showed that by selecting a feature space on average 50 times smaller than the original space, the performance of the classi?er does not decrease by more than 2%.

We are interested in investigating the application of these methods on different types of data. Given the nature of cPCA, which selects those features with high frequences, the analysed data should allow good discrimination based on these high frequence features. We anticipate that a good application would be in image vision. In contrast, cSVM performs an analysis of the data points and is less dependent on how the features contribute towards discriminating between the classes. Our expectations are that cSVM would perform well with any data representable in an Euclidian space.

Acknowledgements

References

[1] H. Almuallim and T. G. Dietterich, "Learning with many irrelevant features," in *Proceedings of the Ninth National Conference on Arti?cial Intelligence (AAAI-91)*, vol. 2. Anaheim, California: AAAI Press, 1991, pp. 547-552 [Online]. Available: citeseer.nj.nec.com/almuallim91learning.html

[2] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the International Conference on Machine Learning*, D. Sleeman and P. Edwards, Eds., 1992, pp. 249-256.

[3] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs, in *NIPS*, 2000, pp. 668-674 [Online]. Available: citeseer.nj.nec.com/article/weston01feature.html

[4] H. Schutze, D. A. Hull, and J. O. Pedersen, "A comparison of classifiers and document representations for the routing problem," in *Research and Development in Information Retrieval*, 1995, pp.229-237 [Online]. Available: citeseer.nj.nec.com/schutze95comparison.html

[5] E. D. Wiener, J. O. Pedersen, and A. S. Weigend, neural network approach to topic spotting, in *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US, 1995, pp. [Online]. Available: citeseer.nj.nec.com/wiener95neural.html

[6] Y. Yang and J. Wilbur, "Using corpus statistics to remove redundant words in text categorization," *Journal of the American Society of Information Science*, no. 47, 1996.

[7] E. Kuznetsov, I. Muchnik, and L. Shvartser, systems and their properties," *Non-Numerical Information Analysis in Social Studies*, 1985.

http://www.datalaundering.com/download/monsysp.pdf

http://www.datalaundering.com/download/cleaning.pdf

[8] T. Rose, M. Stevenson, and M. Whitehead, Reuters Corpus Volume 1 - from yesterday's news to tomorrow's language resources," *in Proceedings of the Third International Conference on Language Resources and Evaluation*, 2002.

[9] E. M. Voorhees and D. Harman, "Trec 2002 overview," in *Proceedings of the 2002 Text Retrieval Conference*, 2002. [Online]. Available: http://trec.nist.gov/act_part/guidelines/?lter2002_guide.html

[10] A. Anghelescu and I. Muchnik, "Complete set of results of classifications in the spaces generated by cSVM and cPCA, compared with those obtained original space," 2003. [Online]. Available: http://mms-01.rutgers.edu/Documents