International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 2, February 2014

Analysis of Sparsity in a Support Vector Machine Based Feature Selection Method

G. Malik[1], M. Tarique[2]

[1]R. L. A (Eve.) College
University of Delhi, India

[2]Dyal Singh College
University of Delhi, India

Abstract: ***Text classification is an important and well studied area of pattern recognition, with a variety of modern applications in natural language documents; we classify text documents into a set of predefined categories. Under the sparse model documents are represented by sparse vectors, where each word in the vocabulary corresponds to one coordinate axis. In a large collections of documents, both the time and memory required for training classifiers connected with the processing of these vectors may This calls for using a feature selection method, not only to reduce the number of features but also to increase the sparsity of document vectors.***

Keywords: Classification, Data Mining, Pattern Recognition, Support Vector

1. Introduction

Trends towards personalizing information services and client-based applications have increased the importance of effective and efficient document categorization techniques. It is that aspect of text classification that led us to explore methods for training classifiers that optimally use the computing memory and processing cycles for the available training data. In particular, we consider tradeoffs between the quality of document classification, as measured by commonly used performance measures, and reductions of a feature set used to represent the data.

In this study we analyze a method for feature selection based on Support Vector Machines [1] with linear kernels. This paper explore how this and other feature selection methods can be used to make tradeoffs between the amount of training data and the sparsity of the data representation when working with a limited amount of system memory.

The experimental results on a large collection of Reuters documents [2] show that the SVM-based feature selection provides a suitable way of preserving the classification performance while significantly reducing the size of the feature space and increasing the sparsity of data.

2. SVM for Feature Selection

Feature selection is the process of selecting a subset of the terms occurring in the training set and using only this subset as features in text classification. Feature selection serves two main purposes. First, it makes training and applying a classifier more efficient by decreasing the size of the effective vocabulary. When the size of training data is very large, it is conceivable that the training of classifiers cannot be performed over the full set of data due to limited computing resources. For example Support Vector Machines, require the whole training data to be stored in the main memory all the time for practical purpose. Thus it can become necessary to work with smaller subsets of training data instead. Now the question is how we can choose the best possible features. Here we analyzed a simple procedure that has proven quite effective and the experimental results show its credibility.

The idea is first to train linear Support Vector Machines on a subset of training data to create initial classifiers. In this model each classifier is a hyperplane separating 'positive' and 'negative' examples for the class and can be represented by the normal and a constant. The second step involves eliminating features that have weights close to zero in this normal, in order to achieve a specified level of data sparsity. Sparsity is here defined as the average number of non-zero components in the vector representation of data or, in other words, the average number of terms left in documents after some terms have been discarded. Finally, using only features retained after the feature selection step, we create a representation of the full training set of documents.

We retrain the linear SVM classifier in the reduced feature space and use the final model to classify the test data. This method is designed to take advantage of the memory freed as a result of increased data sparsity and allow one to work

with larger training sets while keeping the memory consumption constant.

3. Feature Selection Methods

3.1.1 Pruning

Pruning methods are typically applied prior to feature selection to reduce the number of possible features. They are particularly important because the number of possible features is typically very large in a text document, and it is likely that most of these features are irrelevant. Generally, very rare and very frequent words are commonly eliminated. For example, any word appearing two or fewer times is most likely irrelevant. Similarly, the most frequent words are also unlikely to be relevant - for example, "the," "a," "and," "or," etc... are very common in English, but are not usually the features that are useful for classifying a text document.

3.1.2 Random

[7] Brie y describes a random feature selection metric, which was designed for use as a control in the experiment performed by the authors. They cite a study that claims that it scored very high for precision, but had a very low recall rate. The feature selector randomly ranks all features. This appears to be a good method of establishing a baseline for any experimentation with different feature selection metrics.

3.1.3 Normal-Based Feature Selection

In this study we use the Support Vector Machine with linear kernels. Training examples are described by vectors xi = (xi1,...,xid), where d represents the dimensionality of the feature space, i.e., the number of distinct features in the model. In general, the class predictor trained by SVM has the form

Prediction(x) = sgn[b + ∑i αi K(x, xi)]

but in the case of a linear kernel K(x, z) = xTz

this can be rewritten as

sgn[b + wTx] for w = ∑i αi xi,

where the vector of weights w = (w1,...,wd) can be computed and accessed directly. Geometrically, the predictor uses a hyperplane to separate the positive from the negative instances, and w is the normal to this hyperplane. The linear classifier categorizes new data instances by testing whether the linear combination w1x1 + . . . + wdxd of the components of the vector x = (x1,...,xd) is above or below some threshold –b (possibly 0). In our feature selection approach we use the absolute value |wj| as the weight of a feature j; that is, we consider a feature more likely to be useful for training and classification if its coefficient wj has a large absolute value. This type of feature weighting seems intuitively appealing because features with small values of |wj| do not have a large influence on the predictions of the classifier based on w; this can be seen as meaning that these features are not important for classification purposes, and that consequently they could be dispensed with in the training phase as well. A theoretical justification for retaining the highest weighted features in the normal has been independently derived in a somewhat different context by Sindhwani et al. [8]. In this study we also use the linear SVM classifier as the classification model since it has been shown to outperform most of other classification methods on text data [9, 10].

4. Sparsity and the Number of Features

Most of the existing research has focused on the reduction of the number of features (i.e., reduction of feature space dimensionality) rather than increasing the sparsity of the resulting vectors (i.e., reduction of memory requirements for vector storage). It is interesting to explore the relationship between these two aspects of data representation.

It is expected that by reducing the number of features to a certain percentage of the initial set one will increase the sparsity of vectors. However, for a fixed percentage of features to be retained, various feature scoring methods yield significantly different levels of vector sparsity.

It can be seen that these weightings are less reluctant than information gain to assign high scores to less common features but they are not nearly as favorable to rare features as the odds ratio. In addition, normal's trained on larger subsets of the training set are more favorable to rare features than those trained on smaller subsets. A closer look at the selected feature sets reveals that this is partly because some of the features that are rare in the whole training corpus have not been present in the smaller training subsets at all and partly because some were present there but was too rare to be ranked highly.

5. Conclusion and Future Implications

We have studied different feature selection schemes we emphasized the concept of sparsity as a more appropriate characteristic of the data representation than the number of

features used, particularly when a variety of feature selection procedures are considered.

Furthermore, other linear classifiers could be similarly used to weight and select features, including Perceptron [11], Winnow [12], Bayes Point Machine [13], LLSF [14], Widrow-Hoff [15], exponentiated gradient [16, 15], and so on. (Naive Bayes is also essentially a linear classifier if we work with logarithms of probabilities. This has been exploited by e.g. Gärtner and Flach [17].) While these methods are known to be more or less successful in classifying documents, it would be interesting to see how they compare with the SVM-based feature selection method in reducing the feature space.

References

[1] J. Brank, M. Grobelnik, N. Milić-Frayling & D. Mladenić: Feature selection using support vector machines.
[2] Reuters Corpus, Volume 1, English Language, 1996-08-20 to 1997-08-19. Available through http://about.reuters.com/researchandstandards/corpus/ Released in November 2000
[3] Y. Yang, J. O. Pedersen: A comparative study on feature selection in text categorization. Proc. 14th ICML Conf., pp. 412–420, 1997
[4] D. Mladenić, M. Grobelnik: Feature selection for unbalanced class distribution and Naive Bayes. Proc. 15th ICML Conf., pp. 258–267, 1999
[5] D. Mladenić: Feature subset selection in text-learning. Proc. 10th ECML Conf. LNCS vol. 1398, pp. 95–100, 1998
[6] D. Mladenić: Machine learning on non-homogeneous, distributed text data. Ph. D. thesis, University of Ljubljana, Slovenia, 1998
[7] George Forman. An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn, Res., 3:1289{1305, 2003
[8] V. Sindhwani, P. Bhattacharyya, Subrata Rakshit: Information theoretic feature crediting in multiclass support vector machines. First SIAM Int. Conf. on Data Mining, 2001
[9] S. Dumais, J. Platt, D. Heckerman, M. Sahami: Inductive learning algorithms and representations for text categorization. Proc. 7th Int. Conf. on Information and Knowledge Management, pp. 148–155, 1998
[10] T. Joachims: Text categorization with support vector machines: learning with many relevant features. Proc. 10th ECML, LNCS vol. 1398, pp. 137– 142, 1998
[11] F. Rosenblatt: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65(6), pp. 386-408. Reprinted in: J. A. D. Anderson, E. Rosenfeld (Eds.), Neurocomputing: Foundations of Research, MIT Press, 1998, pp. 89–114
[12] N. Littlestone: Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning, 2(4), pp. 285–318, 1988
[13] R. Herbrich, T. Graepel, C. Campbell: Bayes point machines. Journal of Machine Learning Research, 1(Aug), pp. 245–279, August 2001.
[14] Y. Yang, C. Chute: A linear least squares fit method for terminology mapping. Proceedings of the 15th Int. Conf. on Computational Linguistics (COLING 1992), II: 447–53, 1992.
[15] D. D. Lewis, R. E. Schapire, J. P. Callan, R. Papka: Training algorithms for linear text classifiers. Proc. 19th ACM SIGIR Conf., pp. 298–306, 1996
[16] J. Kivinen, M. K. Warmuth: Exponentiated gradient versus gradient descent for linear predictors. Tech. Report UCSC-CRL-94-16, Baskin Center for Computer Engineering & Information Sciences, University of California, USA, June 21, 1994 (revised December 7, 1995)
[17] T. Gärtner, P. A. Flach: WBCSVM: Weighted Bayesian classification based on support vector machines. Proc. 18th ICML Conf., pp. 154–161, 2001

Author Profile

G. Malik received M.Sc in Mathematics from A.M.U and M.Tech in C.S from I.S.I. Kolkata India currently he is working as an Asst Professor in deptt of Mathematics R.L.A College (E), University of Delhi India.

M Tarique received M.Sc in Mathematics from B.H.U and M.Tech in C.S from I.S.I. Kolkata India; currently he is working as an Asst Professor in department of Mathematics Dyal Singh College, University of Delhi India.