

Multiclass Text Classification

A Decision Tree based SVM Approach

Srinivasan Ramaswamy
School of Information
Univeristy of California, Berkeley
srini@ischool.berkeley.edu
CS294 Practical Machine Learning Project

Abstract

This paper discusses about combining Support Vector Machine and decision trees for multi class text classification. Support Vector Machines are trained on each class at each level of the tree and the SVM which is more successful in predicting a class at that level is selected as the decision in that node. Thus a tree is constructed with different SVM in each node. And the tree constructed is used for classifying the multiclass text. Results had shown that this method works comparably better than the other classifiers like simple SVM and Naïve Bayes.

1. Introduction

In the past decade the amount of digital information has rapidly increased and it is growing at an exponential rate. In this era of digital publication there are copious electronic textual information handed everyday. Hence it leads to the problem of Information organization and Management. In many contexts (Dewey, Yahoo and Mesh) trained professional are employed to categorize new items, but it is a highly time consuming and expensive process. Hence automated text classification gained importance and it proved to be an effective solution for the management of growing electronic documents. Application of statistical learning methods and machine learning techniques has increased in the recent past, which includes Bayes probabilistic approaches [6,7], decision trees [7], Support Vector Machines [3] and many other inductive learning algorithms.

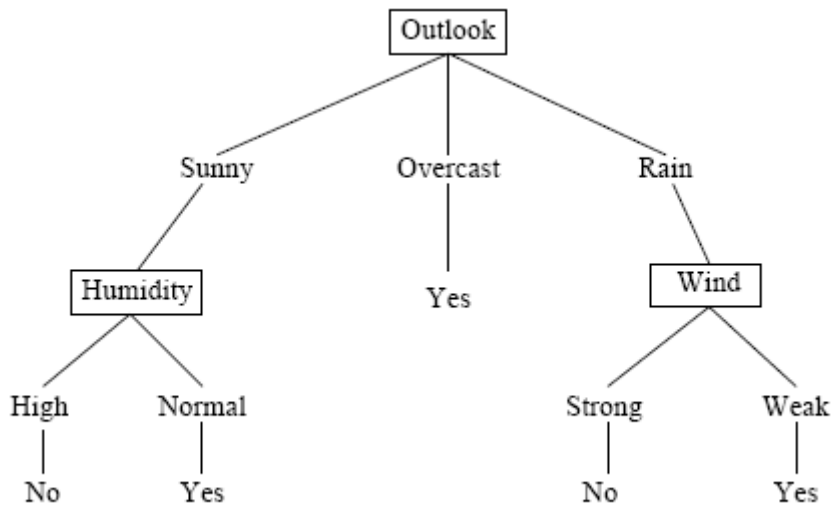
In real world context most of text available cannot be classified into two categories. Hence multiclass text classification would be a major solution to current scenario of information organization and management. This paper aims at experimenting the combination of some of the best classifiers in an efficient way to solve the multi class text classification problem. Support Vector Machines is well known as a powerful binary classifier and it can generalize well when there are many fewer training examples in one class than the other. On the other hand decision trees are well known for their classification of multinomial attributes.

It has been shown in the past that combining classifiers could improve the result

of the classification. In particular Decision trees and SVM combination has lead to good results [10]. This combination would work well for multiclass text classification as SVM is an efficient binary classification and decision trees can be used to arrange different SVM's for different classes in an order which gives maximum information.

2. Decision Trees

A decision tree is a tree whose internal nodes are tests and whose leaf nodes are categories. Each internal node test one attribute and each branch from a node selects one value for the attribute. The attribute used to make the decision is not defined. So we can use the attribute which gives maximum information. And the leaf node predicts a category or class. The decision trees are not limited to boolean functions, but they can be extended for general categorically values functions.

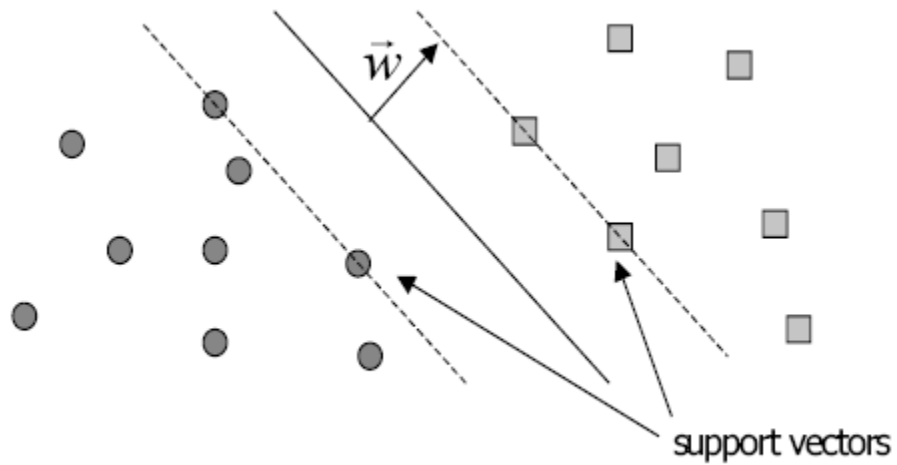


In the above example the given instances can be divided based on the values it take for the attribute “outlook”. The instances are split based on attributes and the one which gives the maximum information is selected as the decision for that node. Hence in the above example we could say that selecting “Outlook” at the root node gives maximum information at that level. And the edges represent the values the attributes can take and the instances are divided accordingly to each child nodes. The tree can be a m-ary tree depending upon the values that the attributes can take. The attribute selection is based on a heuristic approach that the particular attribute will give the best split at a particular level. But this approach has been successful over the past.

3. Support Vector Machines

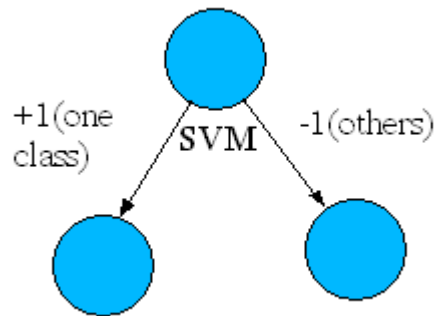
The Support Vector Machines is a classifier, proposed by Vapnik [11], that finds a maximum margin separating hyper plane between two classes of data. Though there are non-linear extensions to SVM, it has been shown that the linear kernels outperform the non-linear kernels in case of text classification [13]. Hence in multiclass text

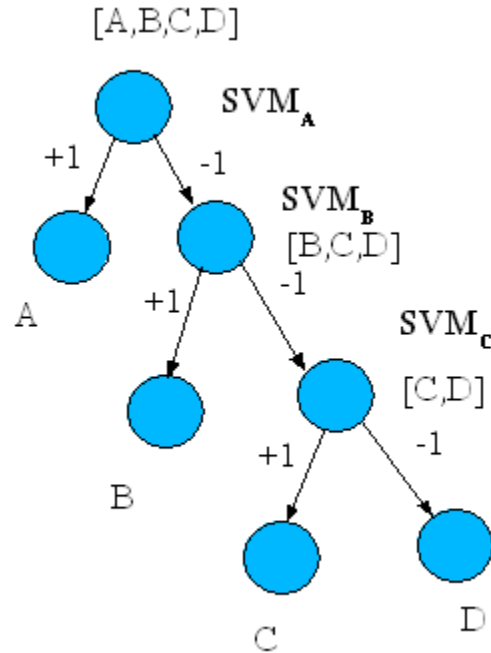
classification, m binary SVM's are trained independently, which acts as a one vs. other classifier.



4. Decision Trees and SVM

We would like to explore both the nature of decision trees and Support vector machines in a way its suitable to the multiclass problem. To achieve better classification for each class, we made the SVM as decisions in the tree. For each level, we train m independent binary SVM's, where m is the number of class unclassified classes at that level. The best SVM is selected at each level from the m SVM's trained, based on the information gain.





To determine the best SVM at a given node, entropy is found for each SVM, based on the number of positive instances classified by each.

$$H(X) = - \sum_{i=1}^n P(X = i) \log_2 P(X = i)$$

It is then used to calculate the information gain based on the entropy of the root node.

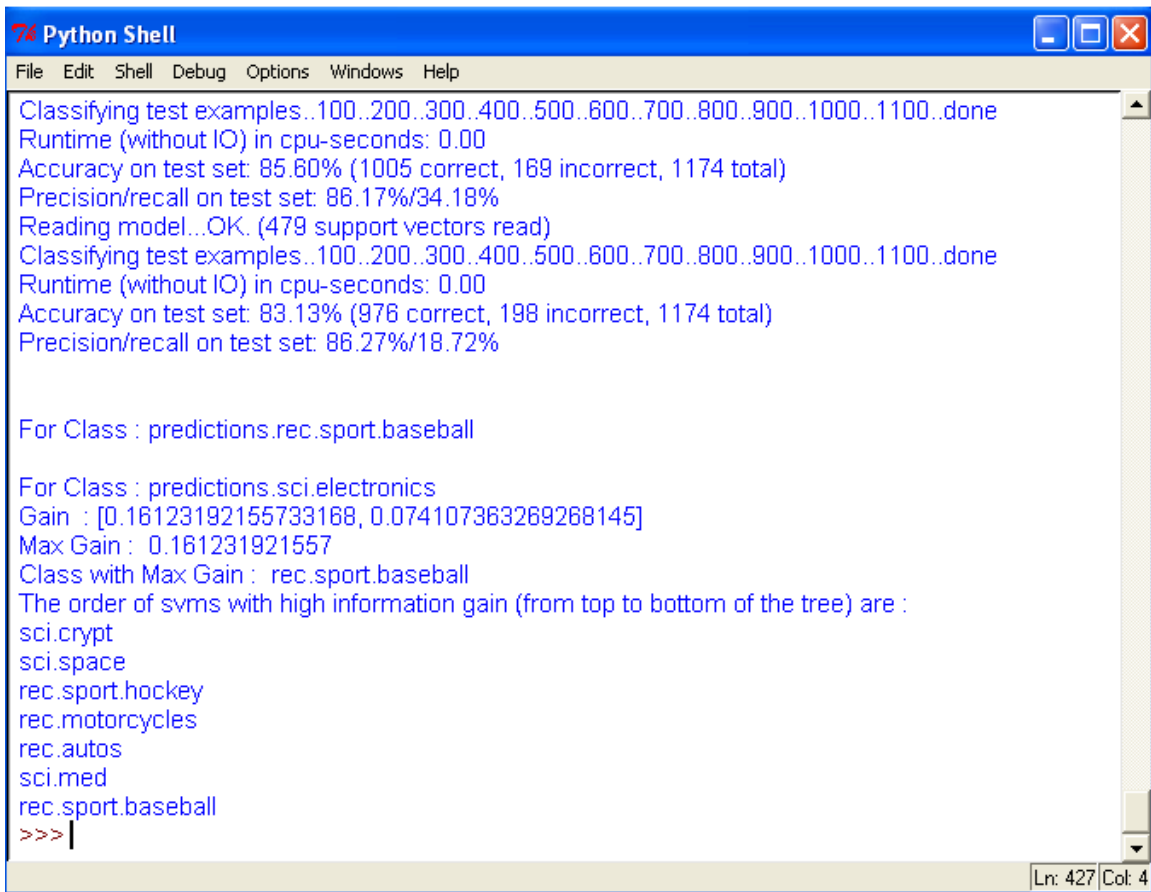
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where A is each decision and S is the set of training instances at a given level. The above mentioned formula for gain gives us the mutual information between the decision A and the target class variable over the set of instances S.

In the example shown above there are four classes A,B,C and D at the first level and we train four different SVM with each of the classes, in a one vs. others model. Among the four SVM we would select the best one, by calculating the information gain for each of them, based on the m classes. And the instances at that level are classified using the best SVM and the left node contains the instances (leaf node in this case) classified as positive by the best SVM. To keep the algorithm simple, we limited ourselves by not processing the left node further. So the left node may contain some false positives, as classified by the best SVM. All the other instances are passed to the right node where another set of SVM's are trained and the same process as mentioned above is repeated till all the classes are classified.

5. Experiments

We selected 20 newsgroups data set and selected 8 class of news groups from the collection for our experiments. The newgroup data is pre-processed and made as simple text files thereby removing all the headers, replies and other non- related information. The preprocessed data from these selected classes are divided as 60% training data and 40% testing data. But we selected only 250 documents for training and around 400 documents for testing. The stop words are eliminated and the feature selection was simple and did not involve any special techniques, to test the classifier with such features.



```
Python Shell
File Edit Shell Debug Options Windows Help
Classifying test examples..100..200..300..400..500..600..700..800..900..1000..1100..done
Runtime (without IO) in cpu-seconds: 0.00
Accuracy on test set: 85.60% (1005 correct, 169 incorrect, 1174 total)
Precision/recall on test set: 86.17%/34.18%
Reading model...OK. (479 support vectors read)
Classifying test examples..100..200..300..400..500..600..700..800..900..1000..1100..done
Runtime (without IO) in cpu-seconds: 0.00
Accuracy on test set: 83.13% (976 correct, 198 incorrect, 1174 total)
Precision/recall on test set: 86.27%/18.72%

For Class : predictions.rec.sport.baseball

For Class : predictions.sci.electronics
Gain : [0.16123192155733168, 0.074107363269268145]
Max Gain : 0.161231921557
Class with Max Gain : rec.sport.baseball
The order of svms with high information gain (from top to bottom of the tree) are :
sci.crypt
sci.space
rec.sport.hockey
rec.motorcycles
rec.autos
sci.med
rec.sport.baseball
>>> |
```

In our implementation of the classifier we used SVMLight [4] to train the SVM. The classes which we selected for our experiments are rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space. The classifier was trained with these classes and then tested with the test data. The accuracy of the individual classes are shown in Table 1.

```

Python Shell
File Edit Shell Debug Options Windows Help
Reading Instance File into instanceDict
Splitting based on Class: rec.sport.baseball .....

Finding Removable Instances for Class : rec.sport.baseball
Testing Completed ...
File Name: C:\mlproject\test.arff
Reading Instance File into instanceDict
Accuracy of Class 0 : 88.6666666667 %
Accuracy of Class 1 : 90.9871244635 %
Accuracy of Class 2 : 63.0769230769 %
Accuracy of Class 3 : 81.6666666667 %
Accuracy of Class 4 : 95.3616352201 %
Accuracy of Class 5 : 17.9652605459 %
Accuracy of Class 6 : 54.5454545455 %
Accuracy of Class 7 : 90.8389585342 %
Total Accuracy : 0.432434120145
Confusion Matrix :
[[ 665.  25.  5.  0.  0.  45.  5.  5.]
 [ 55. 1060.  0.  0.  10.  25.  0.  15.]
 [ 5.  0.  410.  220.  0.  5.  5.  5.]
 [ 0.  0.  165.  735.  0.  0.  0.  0.]
 [ 0.  6.  0.  0.  1213.  35.  0.  18.]
 [1240.  905. 1365. 1015.  845. 1810. 1825. 1070.]
 [ 0.  0.  30.  10.  0.  5.  90.  30.]
 [ 5.  0.  0.  5.  15.  41.  29.  942.]]
>>>|
Ln: 985 Col: 4

```

6. Results

It has been observed that the classes which are predicted at the top level of the tree has good accuracy and the ones predicted at the lower levels have poor accuracy. In our experiment “sci.crypt” which was classified at the root node has 95.83% accuracy.

| Classes | Accuracy |
|--------------------|-----------------|
| rec.autos | 88.6666666667 % |
| rec.motorcycles | 90.9871244635 % |
| rec.sport.baseball | 63.0769230769 % |
| rec.sport.hockey | 81.6666666667 % |
| sci.crypt | 95.3616352201 % |
| sci.electronics | 17.9652605459 % |
| sci.med | 54.5454545455 % |
| sci.space | 90.8389585342 % |

Table 1. Individual Class Accuracy

But on the contrary the last leaf node “sci.electronics” has 17.9652605459 % accuracy,

because all the instances which were not classified correctly (false negatives) are classified as the last leaf node.

| Classes | rec. autos | rec. motorcycle | rec.sport baseball | rec.sport hockey | sci. crypt | sci. electronics | sci.med | sci. space |
|--------------------|------------|-----------------|--------------------|------------------|------------|------------------|---------|------------|
| rec.autos | 133 | 5 | 1 | 0 | 0 | 9 | 1 | 1 |
| rec.motorcycles | 11 | 212 | 0 | 0 | 2 | 5 | 0 | 3 |
| rec.sport.baseball | 1 | 0 | 82 | 44 | 0 | 1 | 1 | 1 |
| rec.sport.hockey | 0 | 0 | 33 | 147 | 0 | 0 | 0 | 0 |
| sci.crypt | 0 | 1 | 0 | 0 | 225 | 6 | 6 | 3 |
| sci.electronics | 248 | 181 | 273 | 203 | 169 | 362 | 365 | 214 |
| sci.med | 0 | 0 | 6 | 2 | 0 | 1 | 18 | 6 |
| sci.space | 1 | 0 | 0 | 1 | 3 | 7 | 5 | 170 |

Table 2 Confusion Matrix

The same dataset and the same feature selection technique is used to test different other classifiers. These experiment were performed with Weka [5]. The results obtained are shown along with the results of our classifier (Table 3). From the results we could infer that the decision tree based SVM perform well when compared to all other ordinary algorithms. Though there are some algorithms which achieve higher accuracy than this decision tree based SVM, our implementation is in its over simplified form. Improving the algorithm in different ways could improve the results further.

| Description | DT+SVM | Ibk | SMO | Naïve |
|----------------------------------|--------|---------|---------|---------|
| Number of Correctly Classified | 1349 | 701 | 965 | 864 |
| Incorrectly Classified instances | 1820 | 2468 | 2204 | 2305 |
| Total Accuracy | 42.64% | 22.1205 | 30.4512 | 27.2641 |

Table 3. Accuracy Statistics

7. Future Work

Currently the tree is restricted to one side and hence the misclassified instances i.e. false positives are not considered for reclassification into the correct class. The accuracy of the classifier could be improved further by considering those false positives for reclassification. Moreover decision tree allows us to select different classifiers at each level of the tree and hence we could experiment by selecting different classifiers like Naïve Bayes at the root and then SVM on the other nodes. And it can be experimented further to find which combination improves the efficiency of the classifier to the maximum.

8. References

- [1] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [2] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM'98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, US, 1998. ACM Press, New York, US.
- [3] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings 10th European Conference on Machine Learning (ECML)*, Springer Verlag, 1998.
- [4] T. Joachims, Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.
- [5] Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
- [6] Kostas Tzeras, Stephan Hartmann. Automatic Indexing Based on Bayesian Inference Networks (1993) (Make Corrections). *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*
- [7] Lewis, D.D. and Ringuette, M.. A comparison of two learning algorithms for text categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, 81-93, 1994.
- [8] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
- [9] T. Mitchell. *Machine learning*. McGraw Hill, New York, US, 1996.
- [10] Thao Nguyen, Iris Bass, Mingkun Li, Ishwar K. Sethi, Investigation of Combining SVM and Decision Tree for Emotion Classification
- [11] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer- Verlag, 1995.

- [12] Yang, Y. and Pedersen, J.O. A comparative study on feature selection in text categorization. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, 412-420, 1997.
- [13] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.