Multiclass Classification on Web Data using Support Vector Machines

Kinshuk Jerath, Sarvjeet Singh, Nikhil Jain, Himanshu Agarwal

May 3, 2002

1 Introduction and Motivation

Support Vector Machines (SVMs) have been found to excel at two-class discriminative learning problems on standard datasets. There have been fairly successful efforts to extend them to the multiclass problem. None of the experiments done so far, though, have been done on web data which is inherently more "difficult" because of high overlap between classes. We make an effort to test SVMs on web data, namely the DMoz and Yahoo! hierarchies, using the ECOC technique. We find that SVMs do not perform well on this data.

2 Algorithm and Implementation

2.1 Crawl

The first step was to crawl the Yahoo! tree to a level such that we get enough data for each class and which is, at the same time, a fair representation of the actual Yahoo! tree i.e. it is not a skewed crawl. To that end we implemented the following crawl algorithm:

1. Deciding upon the classes to be crawled
    - Start from the root
    - If one class is needed, then pick the current class
    - If $n > 1$ classes are needed then get classes from each subtree rooted at this node in the ratio of the number of docs inside each of these classes
2. Retrieving documents from these classes
    - Start at the class node and consider the subtree rooted at this class. The aim is to pick a fixed number of documents from each class

- First take all the documents available at this level i.e. in this class
- The remaining documents are picked from the subclasses rooted at this tree in the ratio of the documents available in each class

2.2 Classification

As mentioned above, SVMs extended to multiclass classification were used, using the ECOC technique. This involves learning a number of different binary classifiers (in this case SVMs) and using their outputs to determine the label for a new test example. In previous experiments using this approach, the lowest known error rates on two standard datasets viz. 20-newsgroups and Industry sector datasets have been reported.

ECOC reduces the multiclass categorization problem to a group of binary classification tasks and combines the binary classification results to predict multiclass labels. R is an $m \times l$ *code matrix*, where m is the number of classes while l is the number of binary classifiers being used. Each entry $R_{ij} \in \{-1, +1\}$. $R_{i\cdot}$ is the i^{th} row of the matrix and defines the code for class i. $R_{\cdot j}$ is the j^{th} column of the matrix and defines a split for the j^{th} classifier to learn. Let $(f_1, \cdots, f_l)$ be the classifiers trained on the partitions indicated in the code matrix. Let $g : \Re \to \Re$ be a chosen loss function. The multiclass label c of a new examples x is

$$H(x) = argmin_{c \in \{1,\cdots,m\}} \sum_{i=1}^{l} g(f_i(x) R_{ci})$$

. The loss function $g(z) = (-z)_+$ where $(z)_+ = max(z, 0)$, is used with SVMs. The code matrix used can be 1) the one-vs-all (OVA) matrix, where the diagonal of the $m \times m$ code matrix is filled with $+1$s while all other entries are -1, and 2) BCH codes, a matrix construction technique that yields high column- and row-separation[1]. High row-separation is desirable since this ensures good error-correcting properties for the codes of the different classes. High column separation ensures that the different classifiers learn different functions and hence do not make the same mistakes in multiple bits i.e. if one classifier makes a mistake, many others do not repeat it. Put in other words, high column separation strives to make the various classifiers independent.

Document vectors were normalized to unit length.

3 Datasets

Experiments were done on the following datasets :

[1]Row-separation is defined to be the average over all rows of the smallest Hamming distance between the row and all other rows: $\frac{1}{n} \sum_{i=1}^{n} min_{j \neq i} Hamming(r_i, r_j)$. Column-separation is defined analogously.

- Yahoo! : This consisted of 83 classes with an average of 1594 training documents and 680 test documents for every class. The quadratic optimizations on this dataset were either taking a lot of time (running into tens of hours) or were failing. We therefore tried classification on a smaller part of the data, namely only the classes which contained "Arts" in their name and had atleast 50 documents (test+train). Note that besides containing all the classes in the "Arts" subtree, it also included classes from other subtrees e.g. the classes Regions.Asia.Arts_and_Crafts and Business_and_Economy.Shopping_and_Services.Arts_and_Crafts were also included. There were 14 classes in all. It may be noted that this approach has the essence of the GraphSVM approach (except the fact that the similar classes were not chosen using any statistical techniques like a cnfusion matrix).

- DMoz : This consisted of 140 classes with an average of 1012 documents per class for training and 483 for testing. The actual DMoz tree available had 481 classes but the ECOC+BCH technique can handle a maximum of 160 classes due to theoretical limitations. Thus, the classes under a node having all children as leaves were combined into one class. e.g. Arts.Animation.Anime, Arts.Animation.Cartoons, Arts.Animation.Others, Animation.Voice_Actors were combined into Arts.Animation. The following sets of classes were mixed.

- Modified 20-newsgroups : To introduce overlap in the classes, sets of similar classes were determined and 30% documents from each of these were moved into a *.mixed class. The above was done for the following class sets:

  1. (a) comp.graphics
     (b) comp.os.ms-windows.misc
     (c) comp.sys.ibm.pc.hardware
  2. (a) alt.atheism
     (b) soc.religion.christian
     (c) talk.religion.misc
  3. (a) talk.politics.guns
     (b) talk.politics.mideast
     (c) talk.politics.misc

An average of 568 training and 187 test documents were there for every class.

4 Results

- Yahoo Arts : The table attached gives the results. There are large variations in the the recall and precision figures. The values for both of these are high

for classes with large number of documents and low for those with small number of documents. Classes with a high number of docs pull docs from classes with a small number of docs. Equivalently, documents originally belonging to classes with a small number of training examples are usually mis-classified.

- Unmodified 20-newsgroups : The table attached gives the results. The recall is far better than with the one-vs-all case but the precision goes down slightly.

- Modified 20-newsgroups : The tables give the results averaged over five runs of the experiment. Again, the results are much better than the one-vs-all case. Specifically, there is a marked increase in recall with that slight trade-off in precision.

5 Conclusion

SVM was not found to perform well on the above data. In general, the optimizations were too slow.

- Yahoo! and DMoz : One major problem here was that the optimizations were taking a lot of time. in the case of Yahoo!, for the classifier learning on the split determined by the first column of the code matrix, the optimization *failed*! The above may have been due to an implementation problem though we have not found one yet. The optimizations on DMoz are also taking a lot of time and were running at the time of preparation of this report.

- 20-newsgroups modified : Recall was found to increase greatly when compared with recall figures obtained by using the one-vs-all approach. Precision values were slightly lesser.

Class name	Recall	Precision	Train docs
ArtsArt_History	34.2657	84.4828	283
ArtsArtists	11.254	57.377	735
ArtsDesign_Arts	59.4883	72.6562	2165
ArtsHumanitiesHistoryBy_RegionU_S_States	60.4381	52.7559	1917
ArtsHumanitiesHistoryBy_Subject	16.763	53.7037	433
ArtsHumanitiesHistoryBy_Time_Period	54.0166	70.9091	857
ArtsHumanitiesHistoryU_S_History	38.2166	55.9701	1853
ArtsHumanitiesLiteratureAuthorsAuthor_Societies	11.1111	60	75
ArtsPerforming_Arts	85.8401	69.3107	3627
ArtsVisual_ArtsBody_Art	9.67742	27.2727	89
ArtsVisual_ArtsPainting	84.0708	53.2348	3177
Buss_and_Eco_Shop_and_ServicesArts_and_Crafts	72.1456	76.0465	4453
RegionalRegionsAsiaArts_and_Humanities	0	0	37
RegionalRegionsEuropeArts_and_HumanitiesHumanities	26.7717	69.3878	261

Table 1: Yahoo! Arts

Class name	Recall	Precision
alt.atheism	83.871	93.4132
comp.graphics	79.4521	78.733
comp.os.ms-windows.misc	81.25	78.7879
comp.sys.ibm.pc.hardware	68.8596	81.7708
comp.sys.mac.hardware	81.6514	86.4078
comp.windows.x	79.5455	88.3838
misc.forsale	86.4865	86.4865
rec.autos	87.6652	89.6396
rec.motorcycles	95.1542	96.4286
rec.sport.baseball	96.5217	93.2773
rec.sport.hockey	97.4026	97.4026
sci.crypt	94.7826	91.9831
sci.electronics	83.1818	74.6939
sci.med	92.511	88.6076
sci.space	94.7368	88.1633
soc.religion.christian	96.9828	83.3333
talk.politics.guns	93.7198	84.7162
talk.politics.mideast	97.2603	90.2542
talk.politics.misc	80.4469	87.8049
talk.religion.misc	59.7222	83.4951

Table 2: Unmodified 20-newsgroups : The recall is much better than the all-vs-one case

Class name	Recall	Precision
alt.atheism	61.3803	70.267
soc.religion.christian	86.3371	64.4639
talk.religion.misc	36.3357	57.3718
religion.mixed	16.8688	26.8527

Table 3: For the case of religion.mixed

Class name	Recall	Precision
comp.graphics	61.9969	66.5286
comp.os.ms-windows.misc	69.5573	57.9256
comp.sys.ibm.pc.hardware	53.4357	55.8353
comp.mixed	20.125	32.2047

Table 4: For the case of comp.mixed

Class name	Recall	Precision
talk.politics.guns	58.7082	61.5
talk.politics.mideast	77.2893	64.2614
talk.politics.misc	47.5054	69.1181
politics.mixed	29.202	28.4289

Table 5: For the case of politics.mixed

Class name	Recall	Precision
comp.sys.mac.hardware	88.3279	79.769
comp.windows.x	88.8637	85.5747
misc.forsale	87.1915	85.4692
rec.autos	87.6652	91.2863
rec.motorcycles	95.3851	96.2311
rec.sport.baseball	96.7467	93.5048
rec.sport.hockey	97.6191	96.162
sci.crypt	95	92.0024
sci.electronics	83.1818	78.9028
sci.med	94.7137	76.6686
sci.space	95.614	87.5502

Table 6: For the other(untouched) classes