Discriminative methods for multi-labeled classification

Shantanu Godbole Sunita Sarawagi

KReSIT, IIT Bombay
Powai, Mumbai, 400076, India
shantanu,sunita@it.iitb.ac.in

Abstract. In this paper we present methods of enhancing existing discriminative classifiers for multi-labeled predictions. Discriminative methods like support vector machines perform very well for uni-labeled text classification tasks. Multi-labeled classification is a harder task subject to relatively less attention. In the multi-labeled setting, classes are often related to each other or part of a is-a hierarchy. We present a new technique for combining text features and features indicating relationships between classes, which can be used with any discriminative algorithm. We also present two enhancements to the margin of SVMs for building better models in the presence of overlapping classes. We present results of experiments on real world text benchmark datasets. Our new methods beat accuracy of existing methods with statistically significant improvements.

1 Introduction

Text classification is the task of assigning documents to a pre-specified set of classes. Real world applications including spam filtering, e-mail routing, organizing web content into topical hierarchies, and news filtering rely on automatic means of classification. Text classification can be broadly categorized into discriminative techniques, typified by support vector machines [1] (SVMs), decision trees [2] and neural networks; and generative techniques, like Naïve Bayes (NB) and Expectation Maximization (EM) based methods. From a performance point of view, NB classifiers are known to be the fastest, learning a probabilistic generative model in just one pass of the training data. Their accuracy is however relatively modest. At the other end of the spectrum lie SVMs based on elegant foundations of statistical learning theory [3]. Their training time is quadratic to the number of training examples, but they are known to be the most accurate.

The simplest task in text classification is to determine whether a document belongs to a class of interest or not. Most applications require the ability to classify documents into one out of many (> 2) classes. Often it is not sufficient to talk about a document belonging to a single class. Based on the granularity and coverage of the set of classes, a document is often about more than one topic. A document describing the politics involved in the sport of cricket, could

be classified as `Sports/Cricket`, as well as `Society/Politics`. When a document can belong to more than one class, it is called multi-labeled. Multi-labeled classification is a harder problem than just choosing one out of many classes.

In this paper, we present algorithms which use existing discriminative classification techniques as building blocks to perform better multi-labeled classification. We propose two enhancements to existing discriminative methods. First, we present a new algorithm which exploits correlation between related classes in the label-sets of documents, by combining text features and information about relationships between classes by constructing a new kernel for SVMs with heterogeneous features. Next, we present two methods of improving the margin of SVMs for better multi-labeled classification. We present experiments comparing various multi-labeled classification methods. Following this, we review related work and conclude with future research directions.

2 Multi-labeled classification using discriminative classifiers

Suppose we are given a vector space representation of n documents. In the bag-of-words model, each document vector d_i has a component for each term feature which is proportional to it's importance (term frequency or TFIDF are commonly used). Each document vector is normalized to unit L_2 norm and is associated with one of two labels, $+1$ or -1. The training data is thus $\{(d_j, c_i), j = 1, \ldots, n\}, c_i \in \{-1, +1\}$.

A linear SVM finds a vector $\mathbf{w}$ and a scalar constant b, such that for all i, $c_i(\mathbf{w}_{c_i} \cdot d_j + b) \geq 1$, and $\|\mathbf{w}\|$ is minimized. This optimization corresponds to fitting the thickest possible slab between the positive ($c = +1$) and negative ($c = -1$) documents.

Most discriminative classifiers, including SVMs, are essentially two-class classifiers. A standard methods of dealing with multi-class problems is to create an *ensemble* of yes/no binary classifiers, one for each label. This method is called *one-vs-others*. For each label l_i, the positive class includes all documents which have l_i as one of their labels and the negative side includes all other documents. During application, the set of labels associated with a document d_j is $\{k\}$, such that $\mathbf{w}_k \cdot d_j + b_k > 0$. This is the basic SVM method (denoted SVM) that serves as a baseline against which we compare other methods.

2.1 Limitations of the basic SVM method

Text classification with SVMs is faced with one issue; that of all classifiers in an ensemble rejecting instances. In one-vs-others, all constituents of the ensemble emit a $(\mathbf{w}_c \cdot d + b_c)$ score; for multi-labeled classification we admit all classes in the predicted set, whose score $\mathbf{w}_c \cdot d + b_c > 0$. However, in practice, we find that a significant fraction of documents get negative scores by all the classifiers in the ensemble.

Discriminative multi-class classification techniques, including SVMs, have historically been developed to assign an instance to exactly *one* of a set of classes that are assumed to be disjoint. In contrast, multi-labeled data, by its very nature, consists of highly correlated and overlapping classes. For instance, in the

Reuters-21578 dataset, there are classes like *wheat–grain*, *crude–fuel*, where one class is almost a parent of the other class although this knowledge is not explicitly available to the classifier. Such overlap among classes hurts the ability of discriminative methods to identify good boundaries for a class. We devise two techniques to handle this problem in Section 4. Correlation between classes can be a boon as well. We can exploit strong mutual information among subsets of classes to "pull up" some classes when the term information is insufficient. In the next section, we present a new method to directly exploit such correlation among classes to improve multi-label prediction.

3 Combining text and class membership features

The first opportunity for improving multi-labeled classification is provided by the co-occurrence relationships of classes in label sets of documents. We propose a new method for exploiting these relationships.

If classification as class C_i is a good indicator of classification as class C_j, one way to enhance a purely text-based SVM learner is to augment the feature set with $|C|$ extra features, one for each label in the dataset. The cyclic dependency between features and labels is resolved iteratively.

Training: We first train a normal text-based SVM ensemble $S(0)$. Next, we use $S(0)$ to augment each document $d \in D$ with a set of $|C|$ new columns corresponding to scores $\mathbf{w}_{c_i} \cdot d + b_{c_i}$ for each class $c_i \in C$. All positive scores are transformed to $+1$ and all negative scores are transformed to -1. In case all scores output by $S(0)$ are negative, the least negative score is transformed to $+1$. The text features in the original document vector are scaled to $f (0 \leq f \leq 1)$, and the new "label dimensions" are scaled to $(1-f)$. Documents in D thus get a new vector representation with $|T|+|C|$ columns where $|T|$ is the number of term features. They also have a supervised set of labels. These are now used to train a new SVM ensemble $S(1)$. We call this method *SVMs with heterogeneous feature kernels* (denoted SVM-HF). The complete pseudo-code is shown in Figure 1. This approach is directly related to our previous work on Cross-Training [4] where label mappings between two different taxonomies help in building better classification models for each of the taxonomies.

1: Represent each document as a vector d in term space and $\|d\| = 1$
2: Build one-vs-rest SVM classifier $S(0)$ using text tokens only
3: **for** each document $d \in D$ **do**
4: Apply $S(0)$ to d, getting a vector $\gamma_C(d)$ of $|C|$ scores (see text)
5: Concatenate vectors d and $\gamma_C(d)$ into a single training vector with label carried over from $S(0)$, with relative term-label weight determined by f, maintaining $\|d\| = 1$
6: Add this vector into the training set of $S(1)$
7: **end for**
8: Induce a new one-vs-rest SVM classifier $S(1)$ for all $d \in D$

Figure 1. SVMs with heterogeneous feature kernels

Testing: During application, all test documents are classified using $S(0)$. For each document, the transformed scores are appended in the $|C|$ new columns with appropriate scaling. These document are then submitted to $S(1)$ to obtain the final predicted set of labels.

The scaling factor: The differential scaling of term and feature dimensions has special reasons. This applies a special kernel function to documents during training $S(1)$. The kernel function in linear SVMs gives the similarity between two document vectors, $K_T(d_i, d_j) = \frac{\langle d_i \cdot d_j \rangle}{|d_i||d_j|}$. When document vectors are scaled to unit L_2 norm, this becomes simply the $cos\theta$ of the angle between the two document vectors, a standard IR similarity measure. Scaling the term and label dimensions sets up a new kernel function given by $K(d_i, d_j) = f \cdot K_T(d_i, d_j) + (1 - f) \cdot K_L(d_i, d_j)$, where K_T is the usual dot product kernel between terms and K_L is the kernel between the label dimensions. The tunable parameter f is chosen through cross-validation on a held out validation set. The label dimensions interact with each other independent of the text dimensions in the way we set up the modified kernel. Just scaling the document vector suitably is sufficient to use this kernel and no change in code is needed.

4 Improving the margin of SVMs

In multi-labeled classification tasks, the second opportunity for improvement is provided by tuning the margins of SVMs to account for overlapping classes. It is also likely that the label set attached with individual instances is incomplete. Discriminative methods work best when classes are disjoint. In our experience with the Reuters-21578 dataset, multi-labeled instances often seem to have incomplete label sets. Thus multi-labeled data are best treated as 'partially labeled'. Therefore, it is likely that the 'others' set includes instances that truly belong to the positive class also. We propose two mechanisms of removing examples from the large negative set which are very similar to the positive set. The first method does this at the document level, the second at the class level.

4.1 Removing a band of points around the hyperplane:

The presence of very similar negative training instances on the others side for each classifier in an SVM ensemble hampers the margin, and re-orients the separating hyperplanes slightly differently than if these points were absent. If we remove these points which are very close to the resultant hyperplane, we can train a better hyperplane with a wider margin. The algorithm to do this consists of two iterations:

1. In the first iteration, train the basic SVM ensemble.
2. For each SVM trained, remove those negative training instances which are within a threshold distance (band) from the learnt hyperplane. Re-train the ensemble.

We call this method the *band-removal* method (denoted BandSVM). When selecting this band, we have to be careful not to remove instances that are crucial in defining the boundary of the others class. We use a held-out validation

dataset to choose the band size. An appropriate band-size tries to achieve the fine balance between large-margin separation, achieved by removing highly related points, and over-generalization, achieved by removing points truly belonging to the negative class.

4.2 Confusion matrix based "others" pruning:

Another way of countering very similar positive and negative instances, is to completely remove all training instances of 'confusing' classes. Confusing classes are detected using a confusion matrix quickly learnt over held out validation data using any moderately accurate yet fast classifier like naïve Bayes [5]. The confusion matrix for a n-class problem is nXn matrix M, where the ij^{th} entry, M_{ij}, gives the percentage of documents of class i which were misclassified as class j. If M_{ij} is above a threshold β, we prune away all confusing classes (like j) from the 'others' side of i when constructing a i-vs-others classifier. This method is called the *confusion-matrix based pruning* method (denoted ConfMat). This two-step method is specified as:

1. Obtain a confusion matrix M over the original learning problem using any fast, moderately accurate classifier. Select a threshold β.
2. Construct a one-vs-others SVM ensemble. For each class i, leave out the entire class j from the 'others' set if $M_{ij} > \beta$.

If the parameter β is very small a lot of classes will be excluded from the others set. If it is too small, none of the classes may be excluded resulting in the original ensemble. β is chosen by cross-validation.

ConfMat is faster to train than BandSVM, relying on a confusion matrix given by a fast NB classifier, and requires only one SVM ensemble to be trained. The user's domain knowledge about relationships between classes (e.g. hierarchies of classes) can be easily incorporated in ConfMat.

5 Experiments

We describe experiments with text classification benchmark datasets and report the results of a comparison between the various multi-labeled classification methods. We compare the baseline SVM method with ConfMat, BandSVM, and SVM-HF.

All experiments were performed on a 2-processor 1.3GHz P3 machine with 2GB RAM, running Debian Linux. *Rainbow*[1] was used for feature and text processing and SVMLight[2] was used for all SVM experiments.

5.1 Datasets

Reuters-21578: The Reuters-21578 Text Categorization Test Collection is a standard text categorization benchmark. We use the Mod-Apte split and evaluate all methods on the given train/test split with 135 classes. We also separately

[1] `http://www.cs.cmu.edu/~mccallum/bow/`
[2] `http://svmlight.joachims.org`

use random 70–30 train/test splits (averaged over 10 random splits), to test statistical significance, for a subset of 30 classes. We did feature selection by using stemming, stopword removal and only considered tokens which occurred in more than one document at least once, and selected the top 1000 features by mutual information.

Patents: The Patents dataset is another text classification benchmark. We used the *wipo–alpha* collection which is an English language collection of patent applications classified into a hierarchy of classes with subclasses and groups. We take all 114 sub-classes of the top level (A to H) using the given train/test split. We also report average over 10 random 70–30 train/test splits for the F sub-hierarchy. We consider only the text in the abstract of the patent for classification and feature selection is the same as that for the Reuters dataset.

5.2 Evaluation measures

All evaluation measures discussed are on a per instance basis and the aggregate value is an average over all instances. For each document d_j, let T be the true set of labels, S be the predicted set of labels. Accuracy is measured by the Hamming score which symmetrically measures how close T is to S. Thus, $\text{Accuracy}(d_j) = |T \cap S|/|T \cup S|$. The standard IR measures of Precision (P), Recall (R) and F_1 are defined in the multi-labeled classification setting as $P(d_j) = |T \cap S|/|S|$, $R(d_j) = |T \cap S|/|T|$, and $F_1(d_j) = 2P(d_j)R(d_j)/(P(d_j) + R(d_j))$.

5.3 Overall comparison

Figures 2 and 3 shows the overall comparison of the various methods on the Reuters and Patents datasets. Figure 2 shows comparison on all 135 classes of Reuters as well as results of averaging over 10 random train/test splits on a subset of 30 classes. Figure 3 shows the comparison for all 114 subclasses of Patents and average over 10 random train/test splits on the F class sub-hierarchy. For both datasets we see that SVM-HF has the best overall accuracy. SVM has the best precision and ConfMat has the best recall. We also observe that BandSVM and SVM-HF are very comparable for all measures.

	30 class subset				All 135 classes			
Method	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
SVM	82.02	**92.65**	82.47	87.26	81.26	**92.41**	82.45	87.15
ConfMat	76.16	81.64	**88.00**	84.7	80.92	87	**88.37**	87.68
BandSVM	83.18	89.87	87.41	88.63	81.73	88.44	87.54	**87.99**
SVM-HF	**84.25**	91.56	86.94	**89.19**	**82**	88.66	87.27	**87.96**

Figure 2. The Reuters-21578 dataset.

We did a directional *t–test* of statistical significance between the SVM and SVM-HF methods for the 30 class subset and the F sub-hierarchy. The accuracy and F_1 scores of SVM-HF were 2% better than SVM, being a small but significant difference at 95% level of significance. The t values were 2.07 and 2.02 respectively over the minimum required value of 1.73 for $df = 18$.

	F class sub-hierarchy				All 114 subclasses			
Method	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
SVM	66.65	**73.65**	67.57	70.48	42.47	**56.76**	43.37	49.16
ConfMat	66.62	69.70	**70.63**	70.16	41.67	53.40	**51.65**	**52.51**
BandSVM	67.30	72.09	68.90	70.45	43.30	55.24	48.61	51.70
SVM-HF	**68.86**	72.06	69.78	**70.90**	**44.41**	55.35	49.84	**52.45**

Figure 3. The Patents dataset

5.4 Interpreting co-efficients

With all documents scaled to unit L_2 norm, inspecting the components of $\mathbf{w}$ along the label dimensions derived by SVM-HF gives us some interesting insights into various kinds of mappings between the labels. The signed components of $\mathbf{w}$ along the label dimensions represent the amount of positive or negative influence the dimension has in classifying documents. As an example for the Reuters dataset, the label dimension for *grain* (+8.13) is highly indicative of the class *grain*. *Wheat* (+1.08) also has a high positive component for *grain*, while *money-fx* (−0.98) and *sugar* (−1.51) have relatively high negative components. This indicates that a document getting classified as *wheat* is a positive indicator of the class *grain*; and a document classified as *sugar* or *money-fx* is a negative indicator of the class *grain*.

5.5 Comparing number of labels

Figure 4 shows the size of the true set of labels T, and the predicted set S. We fix $|S|$ to be $1, 2, 3$ for each $|T| = 1, 2, 3$. For instance, for $|T| = 1$, $|S| = 1$ for 99% of the instances for the SVM method, and only 1% of the instances are assigned $|S| = 2$. For singleton labels, SVM is precise and admits only one label whereas other methods admit a few extra labels.

	T=1			T=2			T=3		
Corresponding S=	1	2	3	1	2	3	1	2	3
SVM	0.99	0.01	0	0.5	0.5	0	0.52	0.35	0.13
ConfMat	0.83	0.14	0.03	0.27	0.63	0.1	0.17	0.3	0.48
BandSVM	0.89	0.09	0.01	0.32	0.64	0.03	0.22	0.3	0.43
SVM-HF	0.94	0.06	0.01	0.34	0.63	0.02	0.3	0.26	0.39

Figure 4. Percentage of instances with various sizes of S for T=1,2,3 with 30 classes of Reuters. Here, 68% of all test instances in the dataset had T=1; 22% had T=2; 8% had T=3; others had T greater than 3.

When $|T| = 2, 3$, we see that SVM still tends to give lesser number of predictions, often just one, compared to the other methods which have a high percentage of instances in the $|T| = |S|$ column. One reason for this is the way one-vs-others is resolved. All negative scores in one-vs-others are resolved by choosing the least negative score and treating this as positive. This forces the prediction set size to be 1 and the semantics of least negative is unclear. The

percentages of documents assigned all negative scores by SVM is 18% for 30 classes of Reuters, while ConfMat, BandSVM, and SVM-HF assign all negative scores to only 4.94%, 6.24%, and 10% of documents respectively.

6 Related work

Limited work has been done in the area of multi-labeled classification. Crammer *et al.* [6] propose a one-vs-others like family on online topic ranking algorithms. Ranking is given by $\mathbf{w}_{c_i} \cdot x$ where the model for each class $\mathbf{w}_{c_i}$ is learnt similar to perceptrons, with an update of $\mathbf{w}_{c_i}$ in each iteration, depending on how imperfect ranking is compared to the true set of labels. Another kernel method for multi-labeled classification tested on a gene dataset is given by Elisseeff *et al.* [7]. They propose a SVM like formulation giving a ranking function along with a set size predictor. Both these methods are topic ranking methods, trying to improve the ranking of all topics. We ignore ranking of irrelevant labels and try to improve the quality of SVM models for automatically predicting labels. The ideas of exploiting correlation between related classes and improving the margin for multi-label classification are unique to our paper.

Postive Example Based Learning–PEBL [8] is a semi-supervise learning method similar to BandSVM. It also uses the idea of removing selected negative instances. A disjunctive rule is learned on features of strongly positive instances. SVMs are iteratively trained to refine the positive class by selectively removing negative instances. The goal in PEBL is to learn from a small positive and a large unlabeled pool of examples which is different from multi-labeled classification.

Multi-labeled classification has also been attempted using generative models, although discriminative methods are known to be more accurate. McCallum [9] gives a generative model where each document is probabilistically generated by all topics represented as a mixture model trained using EM. The class sets which can generate each document are exponential in number and a few heuristics are required to efficiently search only a subset of the class space. The Aspect model [10] is another generative model which can be naturally employed for multi-labeled classification, though no current work exists. Documents are probabilistically generated by a set of topics and words in each document are generated by members of this topic set. This model is however used for unsupervised clustering and not for supervised classification.

7 Conclusions

We have presented methods for discriminative multi-labeled classification. We have presented a new method (SVM-HF) for exploiting co-occurrence of classes in label sets of documents using iterative SVMs and a general kernel function for heterogeneous features. We have also presented methods for improving the margin quality of SVMs (BandSVM and ConfMat). We see that SVM-HF performs 2% better in terms of accuracy and F_1 than the basic SVM method; a small but statistically significant difference. We also note that SVM-HF and BandSVM are very comparable in their results, being better than ConfMat and SVM. ConfMat has the best recall, giving the largest size of the predicted set;

this could help a human labeler in the data creation process by suggesting a set of closely related labels.

In future work, we would like to explore using SVMs with the positive set containing more than one class. The composition of this positive set of related candidate classes is as yet unexplored. Secondly, we would like to theoretically understand the reasons for accuracy improvement in SVM-HF given that there is no extra information beyond terms and linear combinations of terms. Why should the learner pay attention to these features if all the information is already present in the pure text features? We would also like to explore using these methods in other application domains.

Acknowledgments: The first author is supported by the Infosys Fellowship Award from Infosys Technologies Limited, Bangalore, India. We are grateful to Soumen Chakrabarti for many helpful discussions, insights and comments.

References

1. T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML*-1998.
2. R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.
3. V. Vapnik. *Statistical Learning Theory*, John Wiley, 1998.
4. S. Sarawagi, S. Chakrabarti, and S. Godbole. Cross training: learning probabilistic mappings between topics. In *Proceedings of the ACM SIGKDD*-2003.
5. S. Godbole, S. Sarawagi, and S. Chakrabarti. Scaling multi-class support vector machines using inter-class confusion. In *Proceedings of ACM SIGKDD*-2002.
6. K. Crammer and Y. Singer. A family of additive online algorithms for category ranking. *Journal of Machine Learning Research*, 1025–1058, 2003.
7. A. Elisseeff and J. Weston. Kernel methods for multi-labelled classification and categorical regression problems. Technical Report, BioWulf Technologies, 2001.
8. H. Yu, J. Han, and K. C-C. Pebl: Positive example-based learning for web page classification using SVM. In *Proceedings of ACM SIGKDD*-2002.
9. A. McCallum. Multi-label text classification with a mixture model trained by EM. *AAAI Workshop on Text Learning*-1999.
10. T. Hofmann and J. Puzicha. Unsupervised learning from dyadic data. *Technical Report TR-98-042*, Berkeley, 1998.