Using Undirected Graphs to Guide Mutli-label Text Classification

John Cardente[*]

12/13/2013

1 Introduction

The importance of multi-label classification has increased with the growth of online collaboration forums containing large amounts of text on diverse topics. User generated annotations, called tags [1], are commonly used to help discover relevant information. These tags are often inconsistently applied which reduces their effectiveness.

Supervised machine learning techniques can be used to identify the topics within a body of text and predict the appropriate tags. Using these techniques may substantially improve the consistency of text annotation and make finding relevant information easier. This requires machine learning algorithms capable of accurately modeling many tags and processing large data sets in reasonable amounts of time.

This paper presents and evaluates a new Network Guided Naive Bayes (NGNB) classifier that uses undirected graphs to accurately and quickly predict labels for multi-topic text. The NGNB classifier is compared to Binary Relevance Multinomial and Parametric Mixture Naive Bayes models to determine its relative effectiveness.

2 Related Work

Multi-label text classification is a well established field. Tsoumakas et al [4] and Puurala [5] provide a good overview. Madjarov et al [6] compare various multi-label prediction algorithms using a variety of data sets. McCallum [7] and Ueda et al [8] present mixture models based on Naive Bayes classifiers that attempt to learn the correlations between tags. Wang et al [9] and Zhang et al [10] respectively describe the use of directed acyclic and edge-based graphs to perform multi-label classification.

3 NGNB

Many multi-label prediction algorithms, in particular mixture models, explicitly consider tag co-occurrence relationships during training but not prediction. The conditional probabilities generated by these algorithms inherently confound the relationships between tags. While these techniques are effective, improved accuracy and efficiency may be obtained through explicitly considering tag relationships during prediction. The Network Guided Naive Bayes (NGNB) model presented in this paper explores this potential.

During NGNB training, a binary Multinomial Naive Bayes classifier is learned for each tag. An undirected graph is also created from the tag co-occurrences in the training examples. Each graph node represents a single tag. An edge in the graph indicates that the two associated tags appeared together in a training sample label. Self-loops are included in the graph to represent the case when a tag appears alone. Counts for each edge are recorded to represent the strength of the relationships.

During prediction, NGNB first identifies the most likely tag using the per-tag Multinomial Naive Bayes models. This step ignores any tag relationships and evaluates each in isolation. For very large datasets, a graph search starting from highly central nodes can be used to find the most likely tag without having to evaluate all the tags. Once identified, the log-odds of the most likely node is scaled by the ratio of the self-loop edge count to the number of times the tag appeared in the training set. The resulting weighted log-odds is used to estimate the likelihood that the most likely tag should be predicted alone.

Next, the undirected graph is used to determine the set of potentially relevant additional tags. Unlikely

[*] john.cardente@emc.com

tags are removed from this set using the per-tag Multinomial Naive Bayes predictions computed in the first step. The power-set of the remaining nodes, up to a configurable size limit, is computed. Each power-set combination is evaluated by inducing a sub-graph for the associated tags and propagating the per-tag log-odds values from the edge nodes to the node representing the most likely tag. A scaling factor, based on the smallest edge count in the graph, is used to reduce the contribution of the propagated log-odd values as they flow through the network. Using the smallest edge count introduces a penalty for large tag sets and prevents their over prediction. After the propagation phase is completed, the accumulated log-odds value at the node for the most likely tag is used to estimate the likelihood of the combination. After all the power-set combinations are evaluated, the set with the highest accumulated log-odds value is chosen as the final prediction.

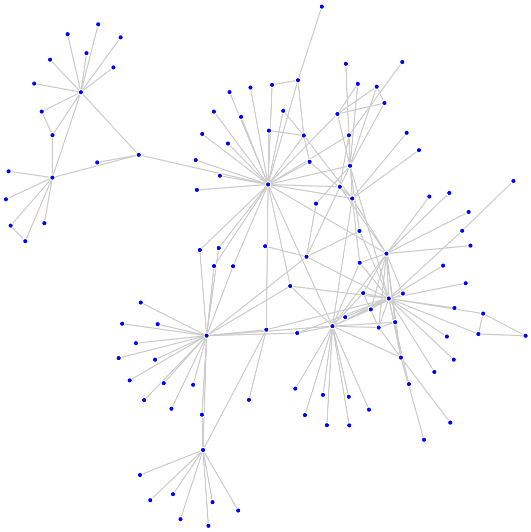

Figure 1: *Tag co-occurrence undirected graph for training set.*

4 Evaluation

4.1 Data

To evaluate the NGNB approach, a corpus of posts to the Stack Exchange website across a wide variety of topics [2] was utilized. The data set, provided by StackExchange, consisted of 6 million posts each containing a title, body, and one or more human generated labels representing the associated topics. The full data set contained 41928 unique tags with an average of 1.6 tags per post.

To reduce the time and resources required to evaluate the NGNB and other models, a smaller training data set was created from 200000 posts containing 116 commonly occurring tags. Figure 1 illustrates the tag co-occurrence undirected graph for this training set. The network consists of a single connected component containing 176 edges between the 116 tag nodes. The NGNB approach can also be applied to networks containing multiple connected components.

To further reduce the evaluation time and resources, uninformative words were removed from the training set using within-class popularity and Gini Coefficient metrics based on [3].

An independent set of 15000 posts were selected to create a test data set. The uninformative words identified in the training were not removed from the test set to evaluate the effect of noisy data.

4.2 Models

To evaluate NGNB, two baseline models were implemented for comparison. A Binary Relevance Multinomial Naive Bayes (BRNB) model was used to establish the effectiveness of predicting each tag in isolation. In this model, a separate Multinomial Naive Bayes binary classifier was built for each tag. During prediction, a tag was selected if the positive outcome probability was greater than that of the negative outcome. No size limit was placed on the final predicted set of tags.

A Parametric Mixture Model (PMM1) was used to represent the effectiveness of considering tag occurrence relationships while calculating the word-tag conditional probabilities. During training, the PMM1 algorithm uses the tags associated with each sample to adjust the conditional word probabilities. After applying an Expectation Maximization process, the resulting conditional word-tag probabilities implicitly reflect the influence of the tag co-occurrence relationships. During prediction, a greedy approach is used to iteratively select the set of tags that most increase the likelihood until it cannot be increased further. See [8] for further details of the PMM1 algorithm.

The NGNB implementation follows the description provided in Section 3. All of the models were implemented from scratch in the Python programming language.

In all three cases, separate classifiers were built for the title and body features of the sampled posts. The union of the tags predicted by the title and body classifiers were used for the overall prediction.

4.3 Cross Validation

Figure 2 provides the recall, precision, and F1 score results from using 10-fold cross validation to evaluated the three models. The BRNB model achieved the highest recall but lowest precision yielding the overall lowest F1 score. Both PMM1 and NGNB performed better with NGNB achieving the best results across all three measures. These results indicate that the naive approach of ignoring tag dependencies substantially reduces performance. They also show that considering these dependencies during the testing phase (NGNB) can be as effective as in the training phase (PMM1).

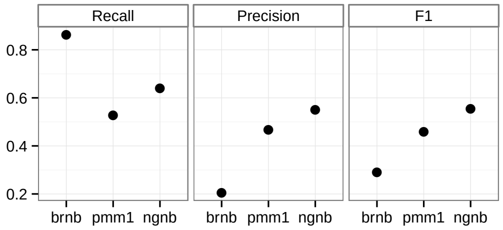

Figure 2: *Recall, Precision, and F1 score results from 10-fold cross validation testing.*

Figure 3 illustrates the average per-fold train and test execution times for the models. The data shows that the PMM1 model required substantially more time to train than the simpler BRNB and NGNB models. PMM1 also took longer to classify the test folds. In the both the training and testing phases, NGNB performed similarly to BRNB.

4.4 Tag Scaling

Figure 4 provides the F1 score results for the three models while increasing the number of possible tags.

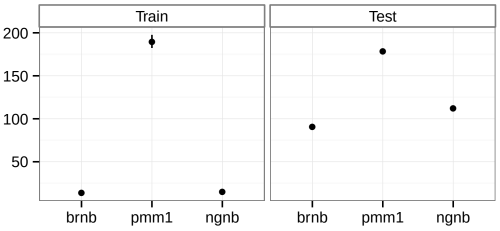

Figure 3: *Cross validation per-fold train and test execution times. Values are in seconds, dots represent the mean, and error bars reflect the 95% confidence interval.*

Although the performance of all three models decrease as the number of tags increase, NGNB exhibits the slowest rate of decay.

Figure 5 illustrates the average per-fold training times for the models. The data shows that the PMM1 model's training time dramatically increases with the number of tags. BRNB and NGNB similarly exhibited a substantially smaller increase in training time as the number of tags increased suggesting that these algorithms are more scalable.

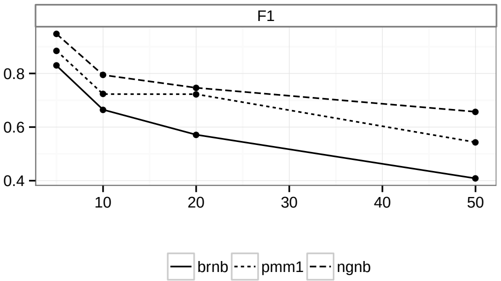

Figure 4: *F1 score 10-fold cross validation results for increasing number of tags.*

4.5 Test Set

Figure 6 provides the recall, predicision, and F1 results from classifying the test data set using the three models. As in the case of cross validation, NGNB provided the best results indicating that the algorithm generalizes well even in the presence of noisy samples.

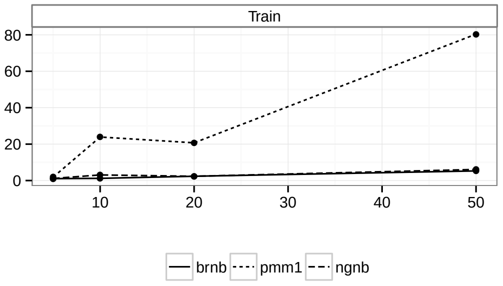

Figure 5: *Average cross validation per-fold train execution times. Y axis is time in seconds. X axis is the number of tags.*

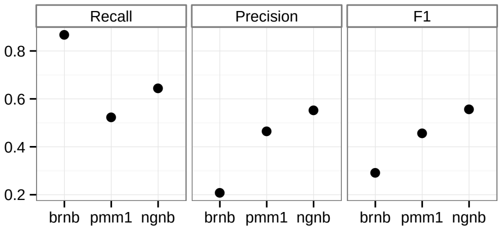

Figure 6: *Recall, Precision, and F1 score results from classifying the test data set.*

5 Conclusions

The growth of online collaborative forums has made multi-label classification a common activity. The size and diversity of these data sets create the need for scalable machine learning algorithms capable of modeling many labels with varying degrees of dependency. Mixture models are a common method used in multi-label classification. This paper presented data indicating that such models may require substantial training time to calculate the conditional probabilities representing the tag relationships. To overcome this challenge, this paper presented a new algorithm call Network Guided Naive Bayes (NGNB) that deferred consideration of the tag dependencies to the prediction phase. Data from cross validation, tag scaling, and test data-set testing provided evidence that NGNB can be as effective as complex mixture models such as PMM1 while only requiring training and test times comparable to simpler models like BRNB. NGNB also degraded the least as the tag population size increased suggesting better scalability. These findings suggest that the NGNB method may be worthy of further investigation and development to enable multi-label classification of large online datasets.

References

[1] http://en.wikipedia.org/wiki/Tag_(metadata)

[2] http://blog.stackoverflow.com/2009/06/stack-overflow-creative-commons-data-dump/

[3] S. Singh, H. Murthy, and T. Gonsalves. Feature selection for text classification based on gini coefficient of inequality. *Journal of Machine Learning Research-Proceedings Track* 10, 76-85.

[4] G. Tsoumakas, and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining* (IJDWM), 2007, 3(3), 1-13.

[5] A. Puurula. Mixture Models for Mutli-label Text Classification University of Waikato.

[6] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Dzeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition* 45.9 2012, 3084-3104.

[7] A. McCallum. Multi-label text classification with a mixture model trained by EM. *AAAI'99 Workshop on Text Learning.* 1999,1-7.

[8] N. Ueda, K. Saito. Parametric mixture models for multi-labeled text. *Advances in neural information processing systems.* 2002, 721-728.

[9] X. Wang, and G. Sukthankar. Multi-label relational neighbor classification using social context features. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* ACM, 2013, 464-472.

[10] M. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* ACM, New York, NY, USA, 999-1008.

[11] C. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval. *Cambridge University Press* 2008