10-701 Project Final Report
Hierarchical Bayesian Models for Text Classification

Yangbo Zhu
yangboz@cs.cmu.edu

1 Introduction

Hierarchical structure is a natural and effective way of organizing information. Well known examples include the Dewey Decimal System, Yahoo Directory and computer file systems. We view such a hierarchy as a tree, which is consisted of a root node, certain levels of intermediate nodes and leaf nodes. Suppose the documents to be classified could be fit into a topic tree. Intuitively, if we know the parents of a leaf node, we can describe the leaf more accurately. Therefore, we can use hierarchical information to improve the performance of text classification. In this project, we propose a new method called *General Hierarchical Shrinkage* (GHS), and compare it with the original *Hierarchical Shrinkage* (HS) method and *Naive Bayes*.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the GHS algorithm and some technical details. Section 4 presents experimental results, and compares GHS with HS and Naive Bayes, along with some preliminary discussion. Section 5 concludes this paper.

2 Related Work

This project is inspired by the *Hierarchical Shrinkage* method in [McCallum 1998]. In HS, we first train a Naive Bayes model for each class of documents. Each class is represented with a leaf node in the topic tree. Given a non-leaf node A, the model of A is the mean of all leaf nodes in the subtree with A as its root. Therefore, the model of root node is the mean of all classes. Furthermore, we add a "super root" on top of the original root node, with a uniform conditional distribution. After building the tree, we assume the model of each class is a linear combination of all the nodes along the path from leaf to the super root. The weights for linear combination can be optimized using a simple *Expectation Maximization* (EM) method.

The HS method arises from a general parameter estimator called *Shrinkage estimator* or *James-Stein estimator*, which was discovered by [Stein 1956], and later extended by [James & Stein 1961]. The basic idea of Shrinkage Estimator is as follows: when estimating a group of parameters $(\theta_1, ..., \theta_n)$, we can reduce the *mean square error* (MSE) by shrinking $\{\theta_i\}$ towards their mean $\bar{\theta} = \sum_i \theta_i$, even if $\{\theta_i\}$ are completely independent. Since this statement contradicts to people's common sense, it's called Stein's Paradox. We will briefly review it in Section 3.2.

[Koller & Sahami 1997] proposes another hierarchical method for text classification, which is called *Pachinko Machine* (PM). PM also group classes into a topic tree, and compute the model of each node based on all documents belonging to it. However, PM does not combine different nodes together to produce mixture models. Instead, it takes a greedy top-down search strategy to locate the "best" leaf-node for the document. The search start at root. At each node A, PM picks a sub-branch of A according to certain criteria. PM repeats this action until it reaches a leaf. Therefore, the accuracy of the entire process is the product of accuracy on all levels. For example, if the accuracy on each level is 0.9, and there are three levels, then the final accuracy is $0.9^3 = 0.73$.

3 Methods

3.1 Naive Bayes

We assume a document is generated by two steps: first choose a class c_j with probability $P(c_j)$, then generate its bag of words according to the conditional distribution $P(w|c_j)$. Based on this assumption, we use the algorithm in Table 6.2 of [Mitchell 1997] to train Naive Bayes classifiers.

Given a labeled document d_i, the probability that it belongs to c_j is $P(c_j|d_i) \in \{0, 1\}$. We estimate the prior distribution of class c_j:

$$P(c_j) = \sum_{i=1}^{|D|} \frac{P(c_j|d_i)}{|D|} \tag{1}$$

where $|D|$ is the number of documents.

The conditional distribution is estimated by:

$$P(w_t|c_j) = \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i)P(c_j|d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N(w_s, d_i)P(c_j|d_i)} \tag{2}$$

where $|V|$ is the vocabulary size, $N(w_t, d_i)$ is the term-frequency (TF) os w_t in d_i.

After the classifier is built, we classify future documents as:

$$c(d_i) = \arg\max_{c_j} P(c_j|d_i) = \arg\max_{c_j} \prod_{w_t \in d_i} \frac{P(w_t|c_j)P(c_j)}{P(w_t)} \tag{3}$$

3.2 James-Stein Estimator

The *James-Stein estimator* is simple to state, but hard to believe at first glance. Assume there are a group of variables $\{x_i\}, i = 1, \ldots, n$, which follow Gaussian distribution $N(\mu, \sigma^2 I)$, where I is the identity matrix. We are interested in estimating the set of parameters μ based on observation $x = X$. A natural and intuitive estimate is the *maximum likelihood estimation* (MLE) $\hat{\mu} = X$. [Stein 1956] demonstrated that, in terms of minimizing *mean square error* (MSE) $E(\|\hat{\mu} - \mu\|^2)$, the *James-Stein estimator* is better than MLE.

The original *James-Stein estimator* shrinks μ towards a prior $\mu = 0$, when $n > 2$:

$$\hat{\mu} = (1 - \frac{(n-2)\sigma^2}{\|X\|^2})X \tag{4}$$

Notice that when $n \leq 2$, MLE is the best.

A generalized *James-Stein estimator* can shrink μ towards non-zero prior, like the mean $\bar{X} = \sum_i x_i$, when $n > 3$:

$$\hat{\mu} = X + (1 - \frac{(n-3)\sigma^2}{\|X - \bar{X}\|^2})(X - \bar{X}) \tag{5}$$

The reason why people are shocked by Stein's claim is that each $\hat{\mu_i}$ is affected by all variables in x, even if they are completely independent. For example, let μ_1 be the weight of cookies in a given box, μ_2 be the height of Mount Everest, and μ_3 be the speed of light, assume the results of our measurement x follow Gaussian distribution described above. The *James-Stein estimator* can get better MSE than maximum likelihood estimator. It means that the expectation of *total* MSE is reduced, while the MSE of each individual μ_i could be better or worse.

3.3 General Hierarchical Shrinkage Model

Recall that in HS method, the final model θ_j of class c_j is a *linear combination* of all the nodes on the path from leaf to root. The model of each intermediate node in the tree is again a *linear combination* of its children. Therefore, θ_j is actually a linear combination of all classes:

$$\theta_j = \sum_{k=1}^{|C|} \lambda_j^k \theta_k, \qquad \sum_{k=1}^{|C|} \lambda_j^k = 1 \tag{6}$$

where $|C|$ is the number of classes.

The weights $\{\lambda_j^k\}$ is constrained by the hierarchical structure. For example, if classes c_{k1} and c_{k2} are siblings of c_j (i.e. They share the same parent node with c_j), then $\lambda_j^{k1} \equiv \lambda_j^{k2}$.

Based on above observation, a straightforward generalization of HS method is to give $\{\lambda_j^k\}$ more freedom. The maximum freedom for $\{\lambda_j^k\}$ is that they can take any non-negative value, as long as $\sum_{k=1}^{|C|} \lambda_j^k = 1$. Like in HS method, we can still train the weights using EM algorithm, although the number of weights increases from $|C||L|$ in HS ($|L|$ is the depth of tree) to $|C|^2$ in GHS. The training algorithm is very similar to that for HS:

In practice, in order to use all training data, we pick a single document as H for each folder, and update λ_j^k according to the average of all folders. Although this strategy increases computational complexity, it's crucial to the success of GHS algorithm.

4 Experiments

4.1 Data Set

We use the Twenty Newsgroups data set collected by Ken Lang. It contains articles from 20 discussion groups of UseNet, with 1000 articles in each group. In order to compare results with [McCallum 1998] and [Toutanova 2001], we pick 15 groups in our experiments, as

Step (1) Initialization: Set $\lambda_j^k = 1/|C|$. Split training data into two parts T and H.

Step (2) Naive Bayes: Use T to estimate $\theta_{jt} = P(w_t|c_j)$.

Step (3) EM Iteration: Use held-out set H to optimize weights $\{\lambda_j^k\}$.

E Step: Compute the probability that words in H_j is generated by θ_k.

$$\beta_j^k = \sum_{w_t \in H_j} \frac{\lambda_j^k \theta_t^k}{\sum_m^{|C|} \lambda_j^m \theta_t^m} \tag{7}$$

M Step: Update the weights by maximizing the expected likelihood of H_j generated by the mixture distribution.

$$\lambda_j^k = \frac{\beta_j^k}{\sum_m^{|C|} \beta_j^m} \tag{8}$$

Step (4) Convergence Test: If $\sum_m |\lambda_j^m - \lambda_j^{m(old)}| < \epsilon$, exit, otherwise continue with Step (3).

Figure 1: General Hierarchical Shrinkage Algorithm

shown in Table 1. After removing stopwords according to SMART system's list of 524 common words, there are 1.7 million words in total, and the size of vocabulary is about 100,000.

Table 1: Topic hierarchy of 15 groups

Vehicles	rec.autos	rec.motorcycles	
Sports	rec.sport.baseball	rec.sport.hockey	
Politics	talk.politics.guns	talk.politics.mideast	talk.politics.misc
Religion	alt.atheism	soc.religion.christian	talk.religion.misc
Computers	comp.graphics	comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware
	comp.windows.x	comp.sys.mac.hardware	

We use the Bow toolkit [1] for indexing and printing out the word/document matrix, then use Perl scripts to convert the matrix into data file that can be loaded by Matlab. We implement Naive Bayes, HS and GHS method in Matlab. Due to the high dimension of features, we heavily rely on sparse matrix representation to speed up computation.

[1] http://www.cs.cmu.edu/ mccallum/bow/

4.2 Field Selection

Each document in the newsgroup data set contains several fields, like "Subject", "Author", "From", "To" etc. However, some fields are too informative for classifiers. For example, Naive Bayes classifier achieve nearly perfect accuracy ($> 95\%$) using a single field "Newsgroups", with 10% training data. Since there are overlap between classes, 95% is almost the upper bound that any reasonable classifier can achieve in this corpus. The "Newsgroups" field list the names of newsgroups each document belongs to, which is the ground truth for classification. Therefore, people have to pretend that they don't know anything about these fields, and try to come up with new algorithms to improve performance. Without the background knowledge about different fields, we can use simple feature selection methods (e.g. mutual information) to locate these informative fields automatically.

To make the results of this paper meaningful to general text classification problems, most experiments in this paper only take Subject and Text for classification. Since most previous work like [McCallum 1998] and [Toutanova 2001] use all fields, this paper also show results on all fields for comparison.

4.3 Results and Discussion

4.3.1 Accuracy

Table 2 and Figure 2 show classification accuracy of Naive Bayes, HS and GHS on fields Subject and Text. GHS outperforms both Naive Bayes and HS when the size of training data is small. The three algorithms achieve similar performance when the training set is large. All results are the average of a 10-fold cross-validation. In each fold, training samples are randomly picked from the original set, and all the rest documents are used as testing set.

Table 2: Classification accuracy using subject and text (The top row is the number of training documents per group, the bottom row is the improvement of GHS over NB)

Method	5	10	20	40	80	100	200	400	600
NB	**0.310**	0.383	0.489	0.583	0.680	0.705	0.758	0.801	0.816
HS	0.259	0.364	0.492	0.593	0.689	0.712	0.763	0.801	0.816
GHS	0.303	**0.430**	**0.556**	**0.641**	**0.711**	**0.730**	**0.772**	**0.805**	**0.817**
Improve	-2.2%	12.2%	13.8%	10.0%	4.6%	3.6%	1.8%	0.5%	0.1%

Table 3 compares the performance of Naive Bayes, HS and GHS on all fields. No significant difference between the three algorithms is observed.

Table 3: Classification accuracy using all fields (The first row is the number of training documents per group)

Method	5	10	20	40	80	100	200	400	600
NB	0.528	0.583	0.671	0.765	0.819	0.836	0.858	0.880	0.882
HS	0.403	0.589	0.681	0.771	0.823	0.836	0.854	0.877	0.880
GHS	0.424	0.624	0.732	0.790	0.821	0.841	0.855	0.877	0.881

Results in Table 3 are not very consistent with that report by [McCallum 1998] and [Toutanova 2001]. In our results, the accuracy of Naive Bayes is much higher than that

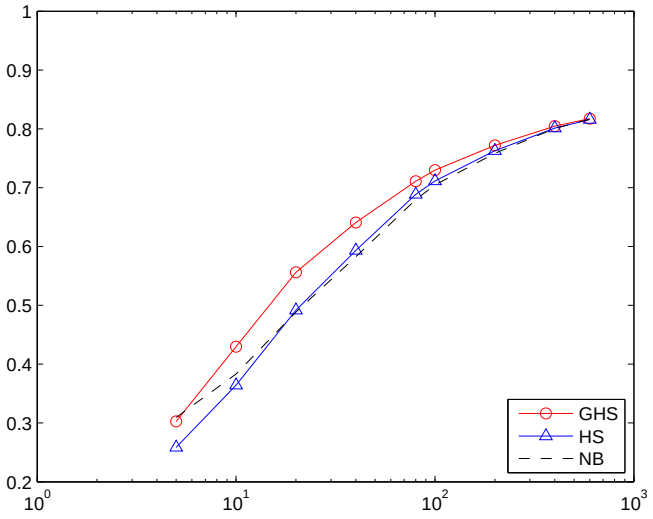

Figure 2: Accuracy of classification with varying size of training set

in those two papers, and the accuracy of HS is a little higher. As a result, the advantage of HS over Naive Bayes is not very significant, at least in this particular corpus.

4.3.2 Mixture weights

Intuitively, when the size of training set increases, the accuracy of Naive Bayes estimation increases. As we are more confident with θ^k, GHS should increase the "self weight" λ_j^j in linear combination $\theta_j = \sum_k \lambda_j^k \theta^k$. Experimental results demonstrate this intuition, which is shown in Figure 3 and Figure 4. However, due to the nature of *James-Stein estimator*, the "self weights" will never equal to one.

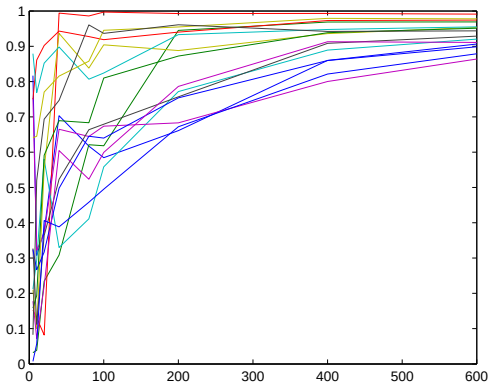

Figure 3: Self weights learned by GHS on varying size of training set

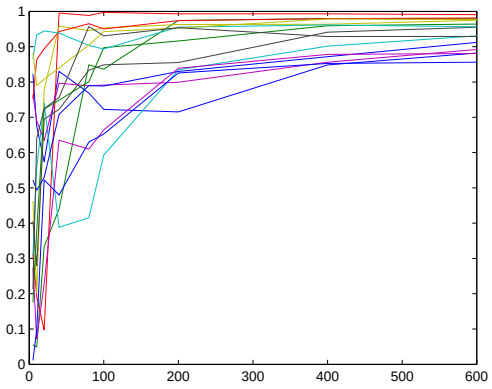

Figure 4: Self weights learned by HS on varying size of training set

5 Conclusion

This paper proposes a new *General Hierarchical Shrinkage* model for text classification. It releases the constraints on mixture weights in original Hierarchical Shrinkage model. Empirical experiments on newsgroup data shows that GHS outperforms Naive Bayes and HS when the training set is small, and achieves the same accuracy with Naive Bayes when the training set is sufficiently large. We can compare these three methods with bias-variance decomposition. Naive Bayes do not require estimating any extra weights, therefore it has the highest bias, and lowest variance. On the other extreme, GHS contains a rich set of tunable weights, which brings low bias as well as high variance. Finally, HS, as a special case of GHS, stands between Naive Bayes and GHS.

References

[Baker 1999] Douglas Baker, Thomas Hofmann, Andrew McCallum and Yiming Yang, A Hierarchical Probabilistic Model for Novelty Detection in Text, *in Proc. of NIPS 1999*.

[Carlin & Louis 2000] Bradley Carlin and Thomas Louis, Bayes and Empirical Bayes Methods for Data Analysis, Second Edition, Chapman and Hall, 2000.

[James & Stein 1961] W. James, Charles Stein, Estimation with Quadratic Loss, *in Proc. of the Fourth Berkeley Symposium on Mathematical Statistics and Probability 1, 361-379, University of California Press*.

[Koller & Sahami 1997] Daphne Koller and Mehran Sahami, Hierarchically Classifying Documents Using Very Few Words, *in Proc. of ICML 1997*.

[McCallum 1998]Andrew McCallum, Ronald Rosenfeld, Tom Mitchell and Andrew Ng, Improving Text Classification by Shrinkage in a Hierarchy of Classes, *in Proc. of ICML 1998*.

[Mitchell 1997] Tom Mitchell, Machine Learning, McGraw Hill, 1997.

[Stein 1956] Charles Stein, Inadmissibility of the Usual Estimator for the Mean of a Multivariate Normal Distribution. *in Proc. of the Third Berkeley Symposium on Mathematical Statistics and Probability 1, 197-206, University of California Press*.

[Toutanova 2001] Kristina Toutanova, Francine Chen, Kris Popat and Thomas Hofmann, Text Classification in a Hierarchical Mixture Model for Small Training Sets, *in Proc. of CIKM 2001*.