Large Margin Multinomial Mixture Model for Text Categorization

Zhen-Yu Pan, Hui Jiang

Department of Computer Science and Engineering, York University,
4700 Keele Street, Toronto, Ontario M3J 1P3, CANADA
Email: {zypan, hj}@cse.yorku.ca

ABSTRACT

In this paper, we present a novel discriminative training method for multinomial mixture models (MMM) in text categorization based on the principle of large margin. Under some approximation and relaxation conditions, large margin estimation (LME) of MMs can be formulated as linear programming (LP) problems, which can be efficiently and reliably solved by many general optimization tools even for very large models. The text categorization experiments on the standard RCV1 text corpus show that the LME method of MMs can largely improve classification accuracy over the traditional training method based on the EM algorithm. Comparing with the EM method, the proposed LME method can achieve over 20% relative error reduction on three independent test sets of RCV1.

Index Terms: Large Margin Estimation (LME), Multinomial Mixture Model (MMM), Linear Programming, Text Categorization

1. INTRODUCTION

The goal of text categorization is the classification of documents into a fixed number of predefined categories. With the rapid growth of online information, text categorization has become one of the key techniques for handling and organizing text data. Text categorization techniques are used to classify news stories, to find interesting information on the WWW, and to guide a user’s search through hypertext. In the literature, many machine learning methods have been used for text categorization, including vector-based methods, support vector machines, Naive Bayes networks, weighted k-Nearest Neighbor (k-NN), multinomial mixture models (MMs), and so on. Among them, MMM has been shown to be quite effective in modeling text documents. Conventionally, MMs are estimated based on the well-known EM algorithm [1]. In this paper, we study a novel discriminative training method based on the principle of large margin to estimate MMs for text categorization. In [2], the so-called large margin estimation (LME) has been successfully applied to Gaussian mixture hidden Markov models (HMMs) for speech recognition. In this work, the idea of large margin estimation will be extended to multinomial mixture models. As we will show, LME of HMMs can be formulated as linear programming (LP) problems under some approximation and relaxation conditions so that the problems can be solved in an extremely efficient way even for very large models. The proposed LME methods have been evaluated on a standard text categorization database, namely the RCV1 corpus [4]. Experimental results clearly show the LME method of MMs can largely improve classification accuracy over the traditional training method based on the EM algorithm. The proposed LME method can achieve more than 20% relative error reduction over the EM method on three RCV1 test sets.

2. MULTINOMIAL MIXTURE MODEL FOR TEXT

In text processing, we normally select a set of features to describe a text document. Some widely used features include how many times a particular word or an n-gram or a phrase or even a given syntax structure appears in a text document. In this way, each text document can be represented as a feature vector, $X = (x_1, \cdots, x_D)$, where $D$ stands for the total number of pre-selected features and each $x_d$ represents the frequency of $d$-th feature in the document.

2.1. Multinomial Model

In text categorization, each class is normally represented by a model. For the $i$-th class, represented by model $\lambda_i$, assume we represent conditional probability $\Pr(x_d = 1 | \lambda_i)$ as $\mu_{id}$, i.e., $\mu_{id} \equiv \Pr(x_d = 1 | \lambda_i), \forall d$. Therefore, given any document, presented by a feature vector $X_t$, if we assume all features are independent given the class identification, the probability to observe this document can be calculated as follows: $\Pr(X_t | \lambda_i) = C_t \prod_{d=1}^{D} \mu_{id}^{x_{td}}$, where $C_t = \left( \sum_{d=1}^{D} x_{td} \right)!$ is a normalization factor and all parameters $\mu_{id}$ satisfy the sum-to-one constraint $\sum_{d=1}^{D} \mu_{id} = 1$, and $x_{td}$ denotes the frequency of $d$-th feature in $X_t$. Obviously, this is a D-dimensional multinomial model. Each conditional probability $\mu_{id}$ for every class, $\lambda_i$, can be easily estimated from training data based on the maximum likelihood (ML) method or some discriminative training criterion, such as minimum classification error (MCE) in [5]. In the text stage, given a new unknown document with its feature vector, $Y = (y_1, \cdots, y_D)$, if all classes are equiprobable, this document should be classified according to the Maximum A Posterior (MAP) decision rule as follows:

$$i^* = \arg \max_i \Pr(Y | \lambda_i) = \arg \max_i \left[ C_Y \prod_{d=1}^{D} \mu_{id}^{y_{td}} \right].$$

(1)

2.2. Multinomial Mixture Model (MMM)

In multinomial mixture models (MMM), each class is modeled by several multinomial models. The contributions from these multinomial models are linearly combined as in other mixture models. In an MMM, denoted as $\lambda_i$, given a document with its feature vector $X_t = (x_{t1}, \cdots, x_{tD})$, the probability to observe the document from this class is computed as:

$$\Pr(X_t | \lambda_i) = \sum_{k=1}^{K} w_{ik} \cdot C_t \cdot \prod_{d=1}^{D} \mu_{ikd}^{x_{td}}$$

(2)
where $\mu_{ikd}$ denotes the conditional probability of the $d$-th feature in $k$-th mixture of model $\lambda_i$ and they all satisfy the sum-to-one constraint, i.e., $\sum_d \mu_{ikd} = 1$, $i, k$, and mixture weights also satisfy the sum-to-one constraint as $\sum_k w_{ik} = 1$ for each model $\lambda_i$. For simplicity, we use $\Lambda$ to denote all MMMs representing all classes.

2.3. MMM Estimation Based on EM

The MMM parameters can also be estimated from training data according to various estimation criteria. However, MMM estimation is not as trivial as multinomial models. In this part, we will first review the traditional estimation methods based on the EM algorithm [1]. Then, we will study some novel discriminative training methods for MMM based on the large margin criterion.

Assume the training set is given and each document is represented by its feature vector $X_t$ ($t = 1, \ldots, T$), in maximum likelihood estimation (MLE), we aim to estimate MMM $\lambda_i$ to maximize the log likelihood function as follows:

$$
\lambda_i^* = \arg \max_{\lambda_i} \sum_{t=1}^T \ln \sum_{k=1}^K \left[ w_{ik} \cdot C_t \prod_{d=1}^D \mu_{ikd} \right] \tag{3}
$$

subject to

$$
\sum_{d=1}^D \mu_{ikd} = 1 \quad \forall i, k, \tag{4}
$$

$$
\sum_k w_{ik} = 1 \quad \forall i. \tag{5}
$$

Obviously, maximization in eq.(3) is not tractable. We need to use the EM algorithm [1] to maximize it iteratively.

In practice, we may need to add smoothing to avoid the data sparseness problem, which results in the following re-estimation formula for one iteration:

$$
\mu_{ikd}^{(n+1)} = \frac{1 + \sum_{t=1}^T X_{td} \cdot \gamma_{ik}}{D + \sum_{t=1}^T \sum_{d=1}^D X_{td} \cdot \gamma_{ik}} \tag{6}
$$

$$
w_{ik}^{(n+1)} = \frac{1 + \sum_{t=1}^T \gamma_{ik}}{K + \sum_{t=1}^T \sum_k \gamma_{ik}}, \tag{7}
$$

where $\gamma_{ik}$ is the so-called responsibility of each mixture, calculated based on the initial models $\lambda_i^{(n)}$ as follows:

$$
\gamma_{ik} = \frac{w_{ik}^{(n)} \cdot \prod_{d=1}^D \mu_{ikd}^{(n)}}{\sum_{k=1}^K \prod_{d=1}^D (\mu_{ikd}^{(n)})^{x_{td}}}, \tag{8}
$$

The above formula actually corresponds to maximum a posteriori (MAP) estimation of MMM under a flat Dirichlet prior distribution.

3. LARGE MARGIN MULTINOMIAL MIXTURE MODEL

In this section, we consider a novel discriminative training method to estimate MMMs for text categorization. In particular, we study the extension of a large margin estimation (LME) method in [2] to estimate MMMs based on the principle of maximizing the minimum separation margin.

Assume training documents are given and each document is represented by its feature vector $X_t$ and its class label is known as $l_t$, let us first define a multi-class separation margin for each training sample, $X_t$, based on log likelihood functions used in the MAP decision rule in eq.(2):

$$
d(X_t | \Lambda) = \ln \Pr(X_t | \lambda_i) - \max_{\lambda_i \neq l_t} \Pr(X_t | \lambda_i) \tag{9}
$$

Obviously, the above margin, $d(X_t | \Lambda)$, roughly measures how far the training document $X_t$ is located from the current decision boundary. If a document is correctly classified, its margin is positive. Otherwise, it is negative.

Based on the large margin principle, even for those training documents with positive margin, we may still want to maximize the minimum margin to build a large margin classifier to achieve more robust and better generalization capability. We first identify a set of the so-called boundary tokens, which all have small positive margins, as:

$$
S = \{ X_t | 0 < d(X_t | \lambda_i) < \varepsilon \} \tag{10}
$$

where $\varepsilon > 0$ is a positive constant. Obviously, all documents in the set $S$ are located relatively close to the decision boundary even though they are all correctly classified.

The large margin principle leads to estimating MMM parameters based on the criterion of maximizing the minimum margin of all boundary tokens in $S$:

$$
\Lambda^* = \arg \max_{\Lambda} \min_{X_t \in S} d(X_t | \Lambda) \tag{11}
$$

where each item $d_t(X_t | \Lambda) = \ln \Pr(X_t | \lambda_i) - \ln \Pr(X_t | \lambda_{l_t})$ is called decision margin. Of course, this minmax optimization should be conducted under sum-to-one constraints in eqs.(4) and (5). In this paper, this method is named as large margin estimation (LME) of MMM. The MMM models estimated in this way are called large margin multinomial mixture models (LM-MMM).

4. ESTIMATING LM-MMM VIA LINEAR PROGRAMMING

However, solving the minmax optimization in the above LME problem is not trivial since it requires optimizing a log likelihood function of MMMs:

$$
l(\lambda_i | X_t) = \ln \Pr(X_t | \lambda_i) = \ln \sum_{k=1}^K \left[ w_{ik} \cdot C_t \prod_{d=1}^D \mu_{ikd} \right]. \tag{12}
$$

Obviously, it is not computationally tractable to directly optimize log-sum in the above equation. We need some approximation methods here to simplify the above log-sum calculation. In this paper, we consider two different approximation methods: One is to use max to approximate the above summation, named as $M$-approx; the other one is to use an expectation-based auxiliary function to approximate the log-sum, named as $E$-approx.

4.1. M-approx: Max-based Approximation

In $M$-approx, the summation over all mixtures in eq.(12) is simply approximated by $\max$ based on the assumption that one mixture dominates the summation. For any document $X_t$, the dominant mixture in model $\lambda_i$, denoted as $s_{t,i}$, can be easily identified based
on the initial models according to: 
\[ \arg \max_{k} \prod_{d=1}^{D} w_{ikd} \cdot \mu_{xtd}^{ikd} \cdot \ln \psi_{ikd} \approx \ln w_{ikd} + \ln \mu_{xtd} \] . Thus, the log likelihood function \( l(\lambda_{i}|X_{i}) \) can be approximated as: 
\[ l(\lambda_{i}|X_{i}) \approx \ln \left[ \psi_{iakd} \cdot \mu_{xtd}^{iakd} \cdot \prod_{d=1}^{D} \mu_{xtd} \right] . \]

For notation convenience, we represent all MMM parameters, \( \Lambda = \{ \mu_{xtd}, w_{ikd} \}, \) in the logarithmic scale: \( \psi_{ikd} = \ln w_{ikd} \) and \( \phi_{ikd} = \ln \mu_{xtd} \). Then, the above approximate log likelihood function can be represented as a linear function of \( \phi_{ikd} \) and \( \psi_{ikd} \). That is,

\[ l(\lambda_{i}|X_{i}) \approx \psi_{iakd} + \sum_{d} x_{td} \cdot \phi_{iakd} + c_{ti} \]  

(13)

where \( c_{ti} \) is a constant independent of MMM parameters. Furthermore, we can approximate the decision margin, \( d_{j1j2}(X_{i}|\Lambda) \) in eq.(11), based on the \( M\text{-approx} \) scheme as the following linear function of \( \phi_{ikd} \) and \( \psi_{ikd} \):

\[ d_{j1j2}(X_{i}|\Lambda) = l(\lambda_{1}|X_{i}) - l(\lambda_{2}|X_{i}) \]  

\[ \approx \sum_{i}^{K} a_{j1j2}^{ik} \cdot \phi_{ikd} + \sum_{i}^{K} \sum_{k=1}^{D} \sum_{d=1}^{D} b_{j1j2}^{ikd} \cdot \phi_{ikd} + c_{j1j2} \]  

(14)

where all coefficients are computed as follows:

\[ a_{j1j2}^{ik} = \delta(i-j_{1}) \delta(k-s_{j1}) - \delta(i-j_{2}) \delta(k-s_{j2}) \]  

(15)

\[ b_{j1j2}^{ikd} = x_{td} \cdot \delta(i-j_{1}) \delta(k-s_{j1}) - x_{td} \cdot \delta(i-j_{2}) \delta(k-s_{j2}) \]  

(16)

\[ c_{j1j2} = c_{j1j2} - c_{j2j2} \]  

(17)

with \( \delta(\cdot) \) denoting the Kronecker delta function.

This approximation is based on one dominant mixture \( s \), which is identified using the initial models, \( \Lambda^{(0)} \). Therefore, this approximation is accurate only within a close neighborhood of \( \Lambda^{(0)} \). Because of this, we need to impose a locality constraint for the LME optimization. One simple choice for the locality constraint is to use the following box constraints:

\[ \phi_{ikd} - \tau_{1} \leq \psi_{ikd} \leq \phi_{ikd} + \tau_{1} \quad \forall i, k, d, \]  

(18)

\[ \psi_{ikd} - \tau_{2} \leq \psi_{ikd} \leq \psi_{ikd} + \tau_{2} \quad \forall i, k. \]  

(19)

where \( \tau_{1} \) and \( \tau_{2} \) are two constants to control box sizes.

Moreover, under the above locality constraints, if we relax the sum-to-one constraints in eqs.(4) and (5), the LME problem of MMM under the \( M\text{-approx} \) can be formulated as the following linear programming (LP) problem:

**Problem 1**

\[ \max_{\Lambda, \rho} \rho \]  

subject to

\[ \sum_{i}^{K} \sum_{k=1}^{D} a_{j1j2}^{ik} \cdot \psi_{ik} + \sum_{i}^{K} \sum_{k=1}^{D} b_{j1j2}^{ikd} \cdot \phi_{ikd} + c_{j1j2} \geq \rho \]  

(21)

for all \( X_{i} \in S \) (with correct model \( \lambda_{i} \)) and other models \( \lambda_{j} \) (\( j \neq i \)).

Along with all boxes constraints in eqs. (18) and (19).

\[ ^{1} \] A linear program is an optimization problem where its objective function and constraints are all linear. Linear programming is a special case of convex optimization and it can be reliably solved with great efficiency.

The above **Problem 1** is a standard LP which can be solved by any general-purpose optimization package. The MMM parameters are then updated based on the found solution. However, since we have relaxed the sum-to-one constraints, the updated model parameters do not satisfy the constraints in eqs.(4) and (5). Strictly speaking, these models are not MMM anymore but we can still use these models to perform classification based on the decision rule in eq.(1). Of course, we can re-normalize the updated model parameters to ensure the sum-to-one constraint but this is not really necessary for text categorization.

### 4.2. E-approx: Expectation-based Approximation

In this section, we consider a different approach to approximate log-sum in the log likelihood function of MMM in eq.(12). For \( l(\lambda_{i}|X_{i}) \), based on the Jensen inequality, we consider the following expectation-based auxiliary function as in EM algorithms:

\[ Q(\lambda_{i}|\lambda^{(n)}_{i}, X_{i}) = \sum_{k=1}^{K} \gamma_{ik} \cdot \ln \left[ w_{ikd} \cdot \mu_{xtd}^{ikd} \right] \]  

(22)

where \( \gamma_{ik} \) is the responsibility of each mixture for \( X_{i} \) as in eq.(8).

According to [6], we have:

\[ Q(\lambda_{i}|\lambda^{(n)}_{i}, X_{i}) \mid_{\lambda_{i} = \lambda^{(n)}_{i}} = l(\lambda_{i}|X_{i}) \]  

(23)

\[ \frac{\partial Q(\lambda_{i}|\lambda^{(n)}_{i}, X_{i})}{\partial \lambda_{i}} \mid_{\lambda_{i} = \lambda^{(n)}_{i}} = \frac{\partial l(\lambda_{i}|X_{i})}{\partial \lambda_{i}} \mid_{\lambda_{i} = \lambda^{(n)}_{i}} \]  

(24)

Therefore, the auxiliary \( Q(\cdot) \) function can be viewed as an approximation to the log likelihood function of MMM within a close neighborhood centered at \( \lambda^{(n)}_{i} \) with accuracy up to first order. Furthermore, the \( Q(\cdot) \) function can also be represented as a linear function w.r.t. all \( \psi_{ikd} \) and \( \phi_{ikd} \) as follows:

\[ Q(\lambda_{i}|\lambda^{(n)}_{i}, X_{i}) = \sum_{k=1}^{K} \sum_{d=1}^{D} (\gamma_{id} x_{td} \cdot \phi_{ikd} + \sum_{k=1}^{K} \sum_{d=1}^{D} (\tilde{f}_{j1j2}^{ikd} \cdot \phi_{ikd} + g_{j1j2} \cdot \psi_{ikd} \cdot \psi_{ikd}) \]  

(25)

where all coefficients are computed as follows:

\[ e_{j1j2}^{ik} = \gamma_{j1j2} \cdot \delta(i-j_{1}) - \gamma_{j1j2} \cdot \delta(i-j_{2}) \]  

(26)

\[ f_{j1j2}^{ikd} = \gamma_{j1j2} x_{td} \cdot \delta(i-j_{1}) - \gamma_{j1j2} x_{td} \cdot \delta(i-j_{2}) \]  

(27)

\[ g_{j1j2} = h_{j1j2} - h_{j2j2} \]  

(28)

After relaxing the sum-to-one constraints, under \( E\text{-approx} \) LME of MMM can be formulated as the following linear programming problem:
Problem 2
\[
\max_{\lambda, \rho} \rho \quad \text{(29)}
\]
subject to
\[
\sum_{i=1}^{N} \sum_{k=1}^{K_i} i_{ik} \cdot \psi_{ik} + \sum_{i=1}^{N} \sum_{d=1}^{D} f_{ikd} \cdot \phi_{ikd} + \frac{1}{2} \sum_{j,j'} \lambda_{jj'} I_{jj'} \geq \rho \quad \text{(30)}
\]
for all \( X_i \in S \) (with correct model \( \lambda_{ij} \)), and other models, \( \lambda_{ij} (j \neq i) \), along with all boxes constraints in eqs. (18) and (19).

Like Problem 1, Problem 2 is also a linear programming (LP) problem and it can be similarly solved with many general optimization tools. Then, all MMM parameters can be updated with the found solution with the sum-to-one constraints relaxed.

5. EXPERIMENTS

The proposed LME methods for MMM have been examined on a standard text categorization task using the Reuters Corpus Volume 1 (RCV1). RCV1 is a large text corpus widely used for evaluating text classification. RCV1 is selected from the Reuters online text database, which consists of English articles between August 20, 1996 and August 19, 1997.[4] There are more than 800,000 newswire articles in the corpus. The articles vary from a few hundred to several thousand words in length. The documents in RCV1 are pre-processed, normalized and manually categorized with respect to topics, industries and regions. According to topic codes, all documents in RCV1 are organized into four hierarchical groups: CCAT (corporate/industrial), ECAT (economics), GCAT (government/social) and MCAT (markets). In our experiments, we perform text categorization among the four general topic categories. To avoid overlapping of categories, those documents belonging to more than one of the four general topic categories are removed in our experiment. Thus, the training set contains 19,806 documents. The development set includes totally 169,417 docs.

We use three different test sets to evaluate the final performance: test1 with 168,823 docs, test2 with 171,141 docs and test3 with 155,884 docs.

Given the amount of training data, we have found that a good choice of mixture number in each MMM is from 4 to 12. In our experiment, we set the mixture number of each MMM to 6, which seems a good tradeoff between accuracy and model complexity. Next, we use the training set and development set to select feature words according to mutual information. We have investigated classification accuracy by using from 200 up to 10,000 features, which are automatically selected based on mutual information. Based on the results (see [7] for details), we have chosen to use 6,000 features in the experiments.

In the experiments, we first build the multinomial baseline (mix=1) using MLE. Then we use the MAP-EM method, in eqs. (6) and (7), to build the MMM baseline with 6 mixtures. Next, we use this MMM baseline (mix=6) as the seed model and apply the large margin estimation, both M-approx (i.e., Problem 1) and E-approx (i.e., Problem 2), to improve classification accuracy (see [7] for details). Here, both Problems 1 and 2 are numerically solved by using the optimization toolbox in Matlab. All of these models are evaluated on the three test sets. The results are given in Table 1. From the results, we can clearly see that the LME methods (both M-approx and E-approx) can significantly reduce classification errors on top of the EM algorithm. Over 20% relative error reduction has been consistently observed on the three independent test sets. If we compare M-approx with E-approx, we can see that E-approx achieves slightly better performance than M-approx on the test sets.

<table>
<thead>
<tr>
<th>Problem 2</th>
<th>( \text{mix}=1 )</th>
<th>( \text{mix}=6 )</th>
<th>( \text{mix}=6 )</th>
<th>( \text{mix}=6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LME</td>
<td>5.78</td>
<td>3.47</td>
<td>2.02</td>
<td>0.82</td>
</tr>
<tr>
<td>MAP-EM</td>
<td>7.09</td>
<td>5.27</td>
<td>3.88</td>
<td>3.71</td>
</tr>
<tr>
<td>LME-M</td>
<td>7.42</td>
<td>5.51</td>
<td>4.16</td>
<td>4.09</td>
</tr>
<tr>
<td>LME-E</td>
<td>6.63</td>
<td>5.04</td>
<td>3.92</td>
<td>3.80</td>
</tr>
<tr>
<td>LME-F</td>
<td>7.33</td>
<td>5.74</td>
<td>4.39</td>
<td>4.22</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS

In this paper, we have extended the idea of large margin estimation (LME) to train multinomial mixture models (MMM) for text categorization. As we have shown, LME of MMMs can be formulated as some linear programming (LP) problems under some approximation and relaxation conditions. Our text categorization experiments on the RCV1 corpus have shown that the LME method of MMMs can largely improve classification accuracy over the traditional training method based on the EM algorithm. Comparing with the EM method, the proposed LME method can achieve over 20% relative error reduction on three independent test sets of RCV1.

7. REFERENCES