Optimal Mixture Models in IR

Victor Lavrenko

Center for Intelligent Information Retrieval
Department of Comupter Science
University of Massachusetts, Amherst, MA 01003,
lavrenko@cs.umass.edu

Abstract. We explore the use of Optimal Mixture Models to represent topics. We analyze two broad classes of mixture models: set-based and weighted. We provide an original proof that estimation of set-based models is NP-hard, and therefore not feasible. We argue that weighted models are superior to set-based models, and the solution can be estimated by a simple gradient descent technique. We demonstrate that Optimal Mixture Models can be successfully applied to the task of document retrieval. Our experiments show that weighted mixtures outperform a simple language modeling baseline. We also observe that weighted mixtures are more robust than other approaches of estimating topical models.

1 Introduction

Statistical Language Modeling approaches have been steadily gaining popularity in the field of Information Retrieval. They were first introduced by Ponte and Croft [18], and were expanded upon in a number of following publications [4, 15, 24, 8, 9, 11, 14]. These approaches have proven to be very effective in a number of applications, including ad-hoc retrieval [18, 4, 15], topic detection and tracking [26, 10], summarization [5], question answering [3], text segmentation [2], and other tasks. The main strength of Language Modeling techniques lies in very careful estimation of word probabilities, something that has been done in a heuristic fashion in prior research on Information Retrieval [21, 19, 20, 25].

A common theme in Language Modeling approaches is that natural language is viewed as a result of repeated sampling from some underlying probability distribution over the vocabulary. If one accepts that model of text generation, many Information Retrieval problems can be re-cast in terms of estimating the probability of observing a given sample of text from a particular distribution. For example, if we knew a distribution of words in a certain topic of interest, we could estimate the probability that a given document is relevant to that topic, as was done in [26, 21, 14]. Alternatively, we could associate a probability distribution with every document in a large collection, and calculate the probability that a question or a query was a sample from that document [18, 4, 15].

1.1 Mixture Models

Mixture models represent a very popular estimation technique in the field of Language Modeling. A mixture model is simply a linear combination of several different distributions. Mixture models, in one shape or another, have been employed in every major

Language Modeling publication to date. For example, *smoothing* [6, 12, 17], a critical component of any language model, can be interpreted as a *mixture* of a topic model with a background model, as highlighted in [15, 13, 16].

This paper will be primarily concerned with the use of mixtures to represent semantic topic models. For the scope of this paper, a topic model will be defined as a distribution, which gives the probability of observing any given word in documents that discuss some particular topic. A popular way to estimate the topic model is by mixing word probabilities from the documents that are believed to be related to that topic. In the next section we will briefly survey a number of publications exploring the use of mixture models to represent topical content.

1.2 Related Work on Mixture Models

Hoffman [9] described the use of latent semantic variables to represent different topical aspects of documents. Hoffman assumed that there exist a fixed number of latent topical distributions and represented documents as weighted mixtures of those distributions. Hoffman used an expectation-maximization algorithm to automatically induce topical distributions by maximizing the likelihood of the entire training set. It is worthwhile to point out that the nature of the estimation algorithm used by Hoffman also allows one to re-express these latent aspect distributions as mixtures of individual document models.

Berger and Lafferty [4] introduced an approach to Information Retrieval that was based on ideas from Statistical Machine Translation. The authors estimated a semantic model of the document as a weighted mixture of translation vectors. While this model does not involve mixing document models, it is still an example of a mixture model.

In the context of Topic Detection and Tracking [1], several researchers used unweighted mixtures of training documents to represent event-based topics. Specifically, Jin et.al. [10] trained a Markov model from positive examples, and Yamron et.al. [26] used clustering techniques to represent background topics in the dataset (a topic was represented as a mixture of the documents in the cluster).

Lavrenko [13] considered topical mixture models as a way to improve the effectiveness of smoothing. Recall that smoothing is usually done by combining the sparse topic model (obtained by counting words in some sample of text) with the background model. Lavrenko hypothesized that by using a *zone* of closely related text samples he could achieve semantic smoothing, where words that are closely related to the original topic would get higher probabilities. Lavrenko used an unweighted mixture model, similar to the one we will describe in section 3.1. The main drawback of the approach was that performance was extremely sensitive to the size of the subset he called the *zone*. A similar problem was encountered by Ogilvie [16] when he attempted to smooth document models with models of their nearest neighbors.

In two very recent publications, both Lafferty and Zhai [11], and Lavrenko and Croft [14] proposed using a weighted mixture of top-ranked documents from the query to represent a topic model. The process of assigning the weights to the documents is quite different in the two publications. Lafferty and Zhai describe an iterative procedure, formalized as a Markov chain on the inverted indexes. Lavrenko and Croft estimate a joint probability of observing the query words together with any possible word in the vocabulary. Both approaches can be expressed as mixtures of document models, and in both

cases the authors pointed out that performance of their methods was strongly dependent on the number of top-ranked documents over which they estimated the probabilities.

1.3 Overview

The remainder of this paper is structured as follows. In section 2 we formally define the problem of finding an *Optimal Mixture Model* (OMM) for a given observation. We also describe a lower bound on solutions to any OMM problem. Section 3 describes unweighted optimal mixture models and proves that finding such models is computationally infeasible. Section 4.1 defines weighted mixture models, and discusses a gradient descent technique for approximating them. Section 5 describes a set of retrieval experiments we carried out to test the empirical performance of Optimal Mixture Models.

2 Optimal Mixture Models

As we pointed out in section 1.2, a number of researchers [13, 16, 11, 14] who employed mixture models observed that the quality of the model is strongly dependent on the subset of documents that are used to estimate the model. In most cases the researchers used a fixed number of top-ranked documents, retrieved in response to the query. The number of documents turns out to be an important parameter that has a strong effect on performance and varies from query to query and from dataset to dataset. The desire to select this parameter automatically is the primary motivation behind the present paper. We would like to find the optimal subset of documents and form an *Optimal Mixture Model*. Optimality can be defined in a number of different ways, for instance it could mean best retrieval performance with respect to some particular metric, like precision or recall. However, optimizing to such metrics requires the knowledge of relevance judgments, which are not always available at the time when we want to form our mixture model. In this paper we take a very simple criterion for optimality. Suppose we have a sample observation: $W_1 \dots W_k$, which could be a user's query, or an example document. The optimal mixture model M_{opt} is a model which assigns the highest probability to our observation.

2.1 Formal Problem Statement

Suppose $W = \{1 \dots n\}$ is our vocabulary, and $W_1 \dots W_k$ is a string over that vocabulary. Let $\mathbb{P}$ be the simplex of all probability distributions over W, that is $\mathbb{P} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \geq 0, |\boldsymbol{x}| = 1\}$. In most cases we will not be interested in the whole simplex $\mathbb{P}$, but only in a small subset $\mathbb{P}' \subset \mathbb{P}$, which corresponds to the set of all possible mixture models. Exact construction of $\mathbb{P}'$ is different for different types of mixture models, and will be detailed later. The optimal model M_{opt} is the element of $\mathbb{P}'$ that gives the maximum likelihood to the observation $W_1 \dots W_k$:

$$M_{opt} = \arg \max_{M \in \mathbb{P}'} P(W_1 \dots W_k | M) \tag{1}$$

In Information Retrieval research, it is common to assume that words $W_1 \dots W_k$ are mutually independent of each other, once we fix a model M. Equivalently, we can

say that each model M is a *unigram* model, and W_i represent repeated random samples from M. This allows us to compute the joint probability $P(W_1 \dots W_k|M)$ as the product of the marginals:

$$M_{opt} = \arg\max_{M \in \mathbb{P}'} \prod_{i=1}^{k} P(W_i|M) \tag{2}$$

Now we can make another assumption common in Information Retrieval: we declare that $W_1 \dots W_k$ are identically distributed according to M, that is for every word w we have $P(W_1 = w|M) = \dots = P(W_k = w|M)$. Assuming W_i are identically distributed allows us to re-arrange the terms in the product above, and group together all terms that share the same w:

$$M_{opt} = \arg\max_{M \in \mathbb{P}'} \prod_{w \in W} P(w|M)^{\#w(W_1 \dots W_k)} \tag{3}$$

Here the product goes over all the words w in our vocabulary, and $\#w(W_1 \dots W_k)$ is just the number of times w was observed in our sample $W_1 \dots W_k$. If we let $T_w = \frac{\#w(W_1 \dots W_k)}{k}$, use a shorthand M_w for $P(w|M)$, and take a logarithm of the objective (which does not affect maximization), we can re-express M_{opt} as follows:

$$M_{opt} = \arg\max_{M \in \mathbb{P}'} \sum_{w \in W} T_w \log M_w \tag{4}$$

Note that by definition, T is also a distribution over the vocabulary, i.e. T is a member of $\mathbb{P}$, although it may not be a member of our subset $\mathbb{P}'$. We can think of T as an *empirical* distribution of the observation $W_1 \dots W_k$. Now, since both M and T are distributions, maximization of the above summation is equivalent to minimizing the cross-entropy of distributions T and M:

$$M_{opt} = \arg\min_{M \in \mathbb{P}'} H(T|M) \tag{5}$$

Equation (5) will be used as our objective for forming optimal mixture models in all the remaining sections of this paper. The main differences will be in the composition of the subset $\mathbb{P}'$, but the objective will remain unchanged.

2.2 Lower Bound on OMM solutions

Suppose we allowed $\mathbb{P}'$ to include all possible distributions over our vocabulary, that is we make $\mathbb{P}' = \mathbb{P}$. Then we can prove that T itself is the unique optimal solution of equation (5). The proof is detailed in section A.1 of the Appendix.

This observation serves as a very important step in analyzing the computational complexity of finding the optimal model M out of the set $\mathbb{P}'$. We proved that any solution M will be no better than T itself. This implies that for every set $\mathbb{P}'$, determining whether $T \in \mathbb{P}'$ is no more difficult than finding an optimal mixture model from that same set $\mathbb{P}'$. The reduction is very simple: given $\mathbb{P}'$ and T, let M be the solution of equation (5). Then, according to section A.1, $T \in \mathbb{P}'$ if and only if $M = T$. Testing

whether $M = T$ can be done in linear time (with respect to the size of our vocabulary), so we have a polynomial-time reduction from testing whether T is a member of $I\!P'$ to solving equation (5) and finding an optimal mixture model.

This result will be used in the remainder of this paper to prove that for certain sets $I\!P'$, solving equation (5) is NP-hard. In all cases we will show that testing whether $T \in I\!P'$ is NP-hard, and use the polynomial-time reduction from this section to assert that solving equation (5) for that particular $I\!P'$ is NP-hard as well.

3 Set-based Mixture Models

The most simple and intuitive type of mixture models is a set-based mixture. In this section we describe two simple ways of constructing a mixture model if we are given a set of documents. One is based on concatenating the documents in the set, the other – on averaging the document models. Very similar models were considered by Lavrenko [13] and Ogilvie [16] in their attempts to create unweighted mixture models. Estimating either of these models from a given set of documents is trivial. However, if we try to look for the *optimal* set of documents, the problem becomes infeasible, as we show in section 3.3.

3.1 Pooled Optimal Mixture Models

First we define a restricted class of mixture models that can be formed by "concatenating" several pieces of text and taking the empirical distribution of the result. To make this more formal, suppose we are given a large collection $\mathcal{C}$ of text samples of varying length. In this paper we will only consider finite sets $\mathcal{C}$. For Information Retrieval applications $\mathcal{C}$ will be a collection of documents. For every text sample $\mathcal{T} \in \mathcal{C}$ we can construct its empirical distribution by setting $\mathcal{T}_w = \frac{\#w(\mathcal{T})}{|\mathcal{T}|}$, just as we did in section 2.1. Here, $|\mathcal{T}|$ denotes the total number of words in $\mathcal{T}$. Similarly, for every subset $S \subset \mathcal{C}$, we can construct its empirical distribution by concatenating together all elements $\mathcal{T} \in S$, and constructing the distribution of the resulting text. In that case, the probability mass on the word w would be:

$$S_w = \frac{\sum_{\mathcal{T} \in S} \#w(\mathcal{T})}{\sum_{\mathcal{T} \in S} |\mathcal{T}|} \tag{6}$$

Now, for a given collection of samples $\mathcal{C}$, we define the pooled mixture set $I\!P_{\mathcal{C},pool}$ to be the set of empirical distributions of all the subsets S of $\mathcal{C}$, where probabilities are computed according to equation (6). We define the *Pooled Optimal Mixture Model* (POMM) problem to be the task of solving equation (5) over the set $I\!P_{\mathcal{C},pool}$, i.e. finding the element $M \in I\!P_{\mathcal{C},pool}$, which minimizes the cross-entropy $H(T|M)$ with a given target distribution T.

3.2 Averaged Optimal Mixture Models

Next we consider another class of mixture models, similar to pooled models described in the last section. These models are also based on a collection $\mathcal{C}$ of text samples, and

can be formed by "averaging" word frequencies across several pieces of text. To make this formal, let $\mathcal{C}$ be a finite collection of text samples. Let $\mathcal{M}$ be the corresponding collection of empirical distributions, that is for each observation $\mathcal{T}_j \in \mathcal{C}$, there exists a corresponding distribution $M_j \in \mathcal{M}$, such that $M_{j,w} = \frac{\#w(\mathcal{T})}{|\mathcal{T}|}$. For a subset $S' \subset \mathcal{C}$, we can construct its distribution by averaging together the empirical distributions of elements in S'. Let S' be a set of text samples, let S be the set of corresponding empirical models, and let $\#(S)$ denote the number of elements in S. The probability mass on the word w is:

$$S_w = \frac{1}{\#(S)} \sum_{M_j \in S} M_{j,w} \tag{7}$$

For a given collection of samples $\mathcal{C}$, we define the averaged mixture model set $I\!P_{\mathcal{C},avg}$ to be the set of averaged distributions of all subsets S' of $\mathcal{C}$, with probabilities computed according to equation (7). We define the *Averaged Optimal Mixture Model* (AOMM) problem to be the task of solving equation (5) over the set $I\!P_{\mathcal{C},avg}$.

3.3 Finding the Optimal Subset is Infeasible

We outlined two possible ways for estimating a mixture model if we are given a set of documents. Now suppose we were given a target distribution T and a collection $\mathcal{C}$, and wanted to find a subset $S \subset \mathcal{C}$ which produces an optimal mixture model with respect to T. It turns out that this problem is computationally infeasible. Intuitively, this problem involves searching over an exponential number of possible subsets of $\mathcal{C}$. In section A.3 of the Appendix we prove that finding an optimal subset for pooled models is NP-hard. In section A.4 we show the same for averaged models. In both proofs we start by using the result of section 2.2 and converting the optimization problem to a decision problem over the same space of distributions. Then we describe a polynomial-time reduction from 3SAT to the corresponding decision problem. 3SAT (described in A.2) is a well-known NP-hard problem, and reducing it to finding an optimal subset of documents proves our searching problem to be NP-hard as well.

It is interesting to point out that we were not able to demonstrate that finding an optimal subset can actually be solved by a nondeterministic machine in polynomial time. It is easy to show that the *decision* problems corresponding to POMM and AOMM are in the NP class, but the original *optimization* problems appear to be more difficult.

4 Weighted Mixture Models

Now we turn our attention to another, more complex class of Optimal Mixture Models. For set-based models of section 3, the probabilities were completely determined by which documents belonged to the set, and no weighting on documents was allowed. Now we consider the kinds of models where in addition to selecting the subset, we also allow putting different weights on the documents in that subset. This flavor of mixture models was used by Hoffman [9], Lafferty and Zhai [11], and Lavrenko and Croft [14] in their research.

4.1 Weighted Optimal Mixture Models

Now if we want to find an optimal mixture model for some observation we not only need to find the subset of documents to use, but also need to estimate the optimal weights to place on those documents. At first glance it appears that allowing weights on documents will only aggravate the fact that finding optimal models is infeasible (section 3.3), since we just added more degrees of freedom to the problem. In reality, allowing weights to be placed on documents actually makes the problem solvable, as it paves the way for numerical approximations. Recall that both POMM and AOMM are essentially combinatorial problems, in both cases we attempt to reduce cross-entropy (equation (5)) over a finite set of distributions: $I\!P_{\mathcal{C},pool}$ for POMM, and $I\!P_{\mathcal{C},avg}$ for AOMM. Both sets are exponentially large with respect to $\mathcal{C}$, but are finite and therefore full of discontinuities. In order to use numerical techniques we must have a continuous space $I\!P'$. In this section we describe how we can extend $I\!P_{\mathcal{C},pool}$ or equivalently $I\!P_{\mathcal{C},avg}$ to a continuous simplex $I\!P_{\mathcal{C},\lambda}$. We define the *Weighted Optimal Mixture Model* problem (WOMM) to be the optimization of equation (5) over the simplex $I\!P_{\mathcal{C},\lambda}$. We argue that a WOMM solution will always be no worse than the solution of a POMM or AOMM for a given $\mathcal{C}$, although that solution may not necessarily lie in $I\!P_{\mathcal{C},pool}$ or $I\!P_{\mathcal{C},avg}$. We look at a simple gradient descent technique for solving WOMM. The technique is not guaranteed to find a globally optimal solution, but in practice converges quite rapidly and exhibits good performance.

WOMM Definition Let $\mathcal{C}$ be our set of text samples and let $\mathcal{M}_{\mathcal{C}}$ be the corresponding set of of empirical models $M_{\mathcal{T}}$ for each sample $\mathcal{T} \in \mathcal{C}$. For an arbitrary set of weights $\lambda \in I\!R^{\#(\mathcal{C})}$ we can define the corresponding model M_λ to be the average of all the models in $\mathcal{M}_{\mathcal{C}}$, weighted by λ:

$$M_{\lambda,w} = \sum_{\mathcal{T} \in \mathcal{C}} \lambda_{\mathcal{T}} M_{\mathcal{T},w} \tag{8}$$

It is easy to verify that equation (8) defines a valid distribution, as long as $|\lambda| = 1$. Now we can define $I\!P_{\mathcal{C},\lambda}$ to be the set of all possible linear combinations of models in $\mathcal{M}_{\mathcal{C}}$, i.e. $I\!P_{\mathcal{C},\lambda} = \{M_\lambda : \lambda \geq 0, |\lambda| = 1\}$. WOMM is defined as solving equation (5) over $I\!P_{\mathcal{C},\lambda}$.

Relationship to Set-based Models It is important to realize that there is a strong connection between WOMM and set-based models from section 3. The simplex $I\!P_{\mathcal{C},\lambda}$ includes both sets $I\!P_{\mathcal{C},pool}$ and $I\!P_{\mathcal{C},avg}$, since:

- **(i)** equations (7) and (8) imply that an AOMM model of a set S is the same thing as a WOMM model M_λ where $\lambda_{\mathcal{T}} = \frac{1}{\#(S)}$ when $\mathcal{T} \in S$, and $\lambda_{\mathcal{T}} = 0$ for $T \notin S$
- **(ii)** equations (6) and (8) imply that a POMM model of a set S is equivalent to a WOMM model M_λ where $\lambda_{\mathcal{T}} = \frac{|\mathcal{T}|}{\Sigma_{\mathcal{T} \in S}|\mathcal{T}|}$ when $\mathcal{T} \in S$, and $\lambda_{\mathcal{T}} = 0$ for $T \notin S$

This implies that every element of either $I\!P_{\mathcal{C},avg}$ or $I\!P_{\mathcal{C},pool}$ is also an element of $I\!P_{\mathcal{C},\lambda}$. Therefore, a weighted optimal mixture model will be as good, or better than any set-based mixture model, as long as we are dealing with the same collection $\mathcal{C}$.

4.2 Iterative Gradient Solution

Since $I\!P_{\mathcal{C},\lambda}$ is a continuous simplex, we can employ numerical techniques, to iteratively approach a solution. We describe a gradient descent approach, similar to the one advocated by Yamron et.al. [26]. Recall that our objective is to minimize the cross-entropy (equation (5) of the target distribution T over the simplex $I\!P_{\mathcal{C},\lambda}$. For a given collection $\mathcal{C}$, every element of $I\!P_{\mathcal{C},\lambda}$ can be expressed in terms of λ, the vector of mixing weights, according to equation (8). We rewrite the objective function in terms of the mixing vector λ:

$$\begin{aligned} & H_\lambda(T|\{M_\mathcal{T}\}) \\ & = -\sum_w T_w \log \sum_\mathcal{T} \left(\lambda_\mathcal{T} / \Sigma_\mathcal{T} \lambda_\mathcal{T}\right) M_{\mathcal{T},w} \\ & = -\sum_w T_w \log \sum_\mathcal{T} \lambda_\mathcal{T} M_{\mathcal{T},w} + \log \sum_\mathcal{T} \lambda_\mathcal{T} \end{aligned} \tag{9}$$

Note that in equation (10), we used the expression $\frac{\lambda_\mathcal{T}}{\Sigma_\mathcal{T} \lambda_\mathcal{T}}$ instead of $\lambda_\mathcal{T}$. Doing this allows us to enforce the constraint that the mixing weights should sum to one without using Lagrange multipliers or other machinery of constrained optimization. In other words, once we made this change to the objective function, we can perform unconstrained minimization over λ. In order to find the maximum of equation (10) we take the derivative with respect to the mixing weight of each element k:

$$\frac{\partial H_\lambda}{\partial \lambda_k} = -\sum_w \frac{T_w M_{k,w}}{\sum_\mathcal{T} \lambda_\mathcal{T} M_{\mathcal{T},w}} + \frac{1}{\sum_\mathcal{T} \lambda_\mathcal{T}} \tag{10}$$

After setting the derivative equal to zero, and re-arranging the terms, we see that the extremum is achieved when for every $k \in \mathcal{C}$ we have:

$$1 = \sum_w \frac{T_w M_{k,w}}{\sum_\mathcal{T} \left(\lambda_\mathcal{T} / \Sigma_\mathcal{T} \lambda_\mathcal{T}\right) M_{\mathcal{T},w}} \tag{11}$$

We can take this equation and turn it into an incremental update rule. Suppose λ_k^n is the mixing weight of element k after n iterations of the algorithm. Then at the next iteration the weight should become:

$$\lambda_k^{n+1} \longleftarrow \sum_w \frac{T_w M_{k,w} \lambda_k^n}{\sum_\mathcal{T} \left(\lambda_\mathcal{T} / \Sigma_\mathcal{T} \lambda_\mathcal{T}\right) M_{\mathcal{T},w}} \tag{12}$$

It is easy to see that when the extremum is achieved, equation (11) holds, and the value λ_k will not change from one iteration to another, so the procedure is convergent. In practice, it is sufficient to run the procedure for just a few iterations, as it converges rapidly. Every iteration of update (12) requires on the order of $(\#(\mathcal{C}) \times \#(W))$, and the number of iterations can be held at a constant.

Local Minima It is important to realize that the iterative update in equation (12) is not guaranteed to converge to the global minimum of equation (10). The reason for that is that the objective function is not convex everywhere. We can see that clearly when we take the second derivative of the objective with respect to the mixture weights:

$$\frac{\partial^2 H_\lambda}{\partial \lambda_k \partial \lambda_l} = \sum_w \frac{T_w M_{k,w} M_{l,w}}{\left(\sum_j \lambda_j M_{j,w}\right)^2} - \frac{1}{\left(\sum_j \lambda_j\right)^2} \tag{13}$$

It is not obvious whether left-hand side of the equation above is positive or negative, so we cannot conclude whether the function is globally convex, or whether it has local minima. In practice we found that the incremental algorithm converges quite rapidly.

5 Experimental Results

In this section we discuss an application of Optimal Mixture Models to the problem of estimating a topic model from a small sample. The experiments were carried out in the following setting. Our collection $\mathcal{C}$ is a collection of approximately 60,000 newswire and broadcast news stories from the TDT2 corpus [7]. For this dataset, we have a collection of 96 event-centered topics. Every topic is defined by the set $R \subset \mathcal{C}$ of stories that are relevant to it. The relevance assessments were carried out by LDC [7] and are exhaustive.

For every topic, our goal is to estimate M_R, the distribution of words in the documents relevant to that topic. We assume that the relevant set R is unknown, but we have a single example document $\mathcal{T} \in R$. The goal is to approximate the topic model M_R as closely as possible using only $\mathcal{T}$ and $\mathcal{C}$. We formulate the problem as finding the optimal weighted mixture model, and use the iterative update detailed in section 4.2 to estimate the optimal mixture. Since we do not know R, we cannot optimize equation (5) for M_R directly. We hypothesize that optimizing for $T = M_\mathcal{T}$ is a good alternative. This hypothesis is similar to the assumptions made in [14]. Note that this assumption may be problematic, since effectively $\mathcal{T}$ is an element of $\mathcal{C}$. This means that our gradient solution will eventually converge to $M_\mathcal{T}$, which is not what we want, since we want to converge to M_R. However, we hope that running the gradient method for just a few iterations will result in a reasonable mixture model.

We carry out three types of experiments. First we demonstrate that the gradient procedure described in section 4.2 indeed converges to the target. Second we look at how well the resulting mixture model approximates the real topic model M_R. Finally, we perform a set of *ad-hoc* retrieval experiments to demonstrate that our mixture model can be used to produce effective document rankings.

5.1 Convergence to Target

Figure 1 shows how quickly the weighted mixture model converges to the target distribution $M_\mathcal{T}$. On the y-axis we plotted the relative entropy (equation (14) in the Appendix) between the mixture model and the target model, as a function of a number of gradient updates. Relative entropy is averaged over all 96 topics. The solid line shows

the mean value, while the bars indicate one standard deviation around the mean. We observe that relative entropy rapidly converges to zero, which is understandable, since $\mathcal{T} \in \mathcal{C}$. Eventually, the iterative procedure will force the mixing vector λ to have zeros in all places except $\lambda_{\mathcal{T}}$, which would be driven to one.

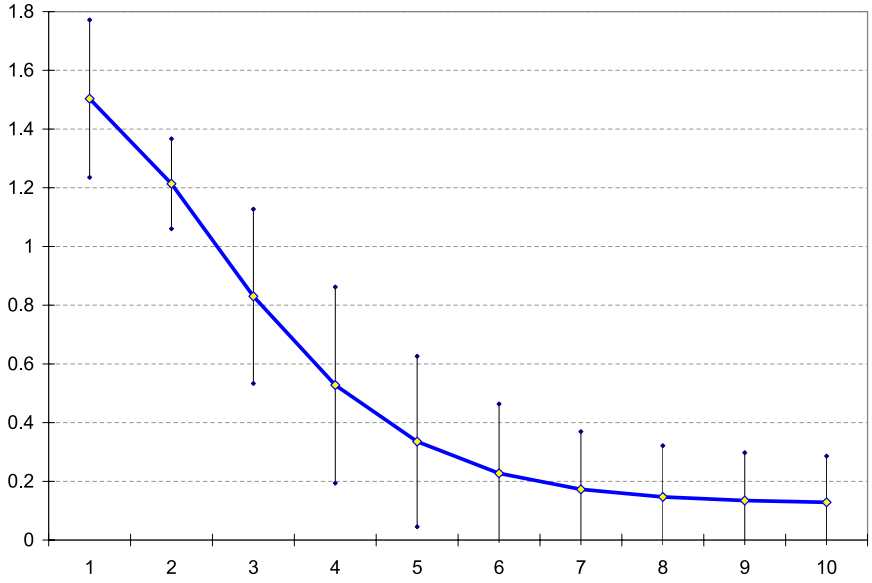

Fig. 1. Convergence of the Weighted Optimal Mixture Model to the target distribution. Target is a single document discussing the topic of interest. Mixture Model estimated using iterative gradient updates.

5.2 Convergence to True Topic Model

In the next set of experiments, we measure relative entropy of our mixture model with respect to M_R, which is the true topic model. M_R is estimated as an averaged mixture model from R, the set of documents known to be relevant to the topic. Note that since the solutions were optimized towards $M_{\mathcal{T}}$ and not towards M_R, we cannot expect relative entropy to continue decreasing with more and more iterations. In fact, once the solution is very close to $M_{\mathcal{T}}$, we expect it to be "far" from M_R. What we hope for is that after a few (but not too many) iterations, the solution will be sufficiently close to M_R. Figure 2 shows the performance with respect to M_R for 0, 1, 2 and 5 iterations. We show relative entropy as a function of how many documents we include in our collection $\mathcal{C}$. The documents are ranked using $\mathcal{T}$ as query, and then some number n of top-ranked documents is used. This is done to highlight the fact that even with gradient optimization, the model shows strong dependency on how many top-ranked documents are used to estimate the probabilities. For comparison, we show the lower bound, which is what we would obtain if we used n documents that are known to be in R.

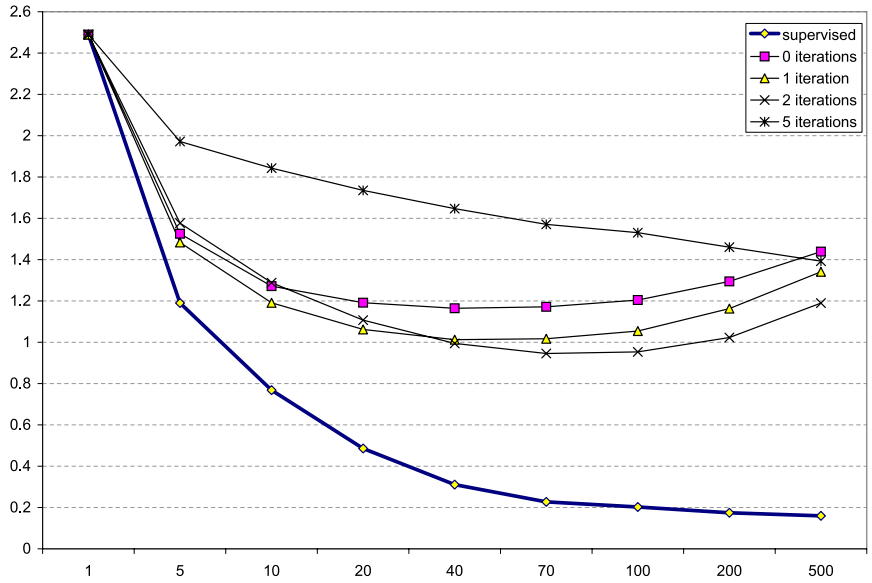

Fig. 2. Convergence of the mixture model to the true topic model: running the gradient update for two iterations is promising.

We observe that running the gradient algorithm for two iterations moves the solution closer to M_R, but doing more iterations actually hurts the performance. Running it for five iterations results in higher relative entropy than not running it at all. We also note that with more iterations, performance becomes less sensitive to the number of documents in $\mathcal{C}$. Overall, we can conclude that running the gradient algorithm for very few iterations is promising.

5.3 Retrieval Experiments

Finally, we consider an application of Optimal Mixture Models to the problem of document retrieval. We start with a single relevant example and estimate a mixture model as was described above. Then, for every document in our dataset we compute the relative entropy (equation (14) in the Appendix) between the model of that document and the estimated mixture model. The documents are then ranked in increasing order of relative entropy. This type of ranking was first proposed by Lafferty and Zhai [11] and was found to be quite successful in conjunction with *Query Models* [11] and *Relevance Models* [14].

Figure 3 shows the retrieval performance when a mixture model is estimated over a set of 10 top-ranked documents using two iterations of the gradient update. To provide a baseline, we replace the estimated mixture model by the model of the single training example, and perform retrieval using the same ranking function. For comparison we also show the performance unweighted mixture model formed from the same set of 10 top-ranked documents.

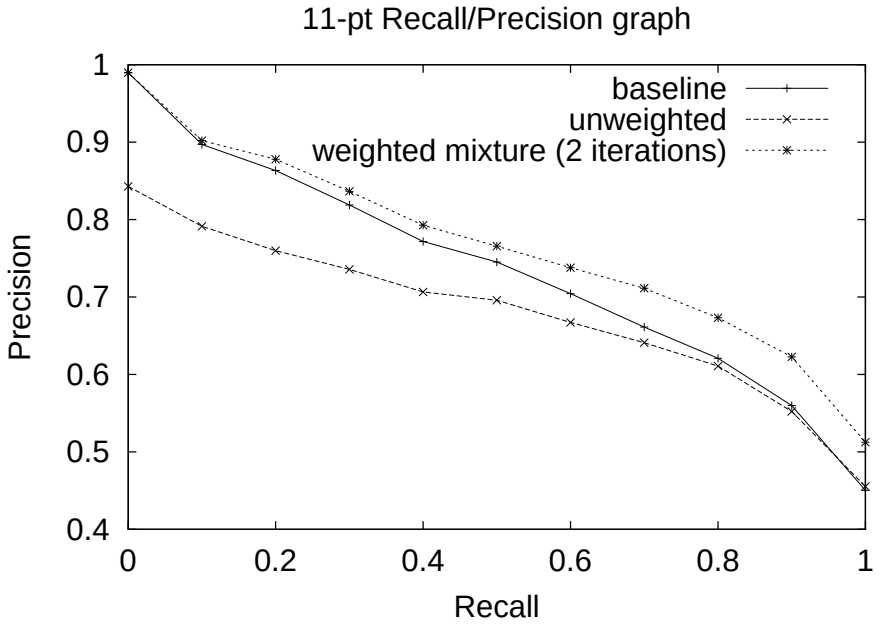

Fig. 3. Document retrieval: mixture model (top) outperforms a baseline of single training example (middle). Mixture model used 10 documents for two iterations.

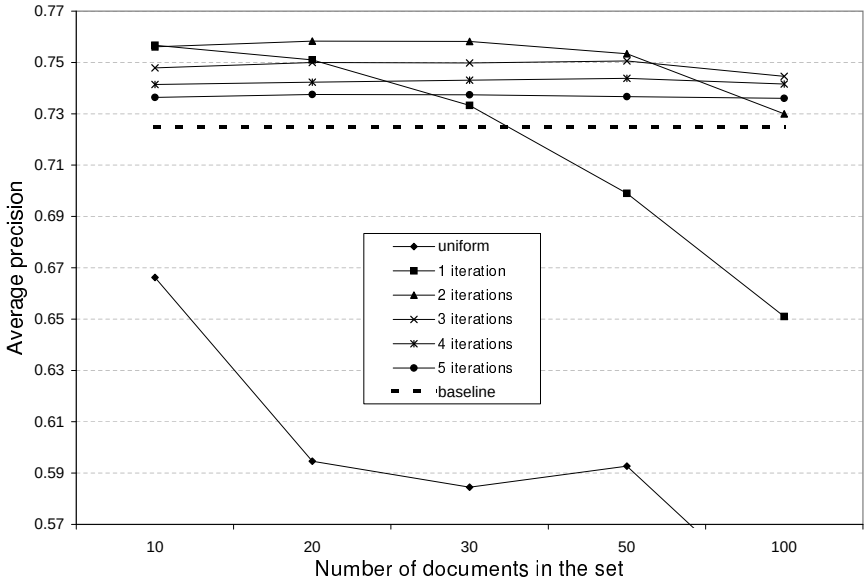

Fig. 4. Optimal Mixture Models exhibit less dependency on the number of top-ranked documents used in the estimation. Two or more iterations with any number of documents outperforms the baseline.

We observe that our weighted mixture noticeably outperforms the baseline at all levels of recall. The improvement is statistically significant at all levels of recall, except at zero. Significance was determined by performing the sign test with a p value of 0.05. The magnitude of the improvement is not very large, partly due to the fact that the baseline performance is already very high. Baseline has non-interpolated average precision of 0.7250, which is very high by TREC standards. Such high performance is common in TDT, where topic definitions are much more precise than in TREC. Weighted mixture model yields average precision of 0.7583, an improvement of 5%. Note that unweighted mixture model constructed from the same set of documents as the weighted model performs significantly worse, resulting in a 9% drop from the baseline. This means that document weighting is a very important aspect of estimating a good topic model.

Finally, we re-visit the issue that motivated this work, the issue of sensitivity to the number of top-ranked documents used in the estimation. We repeated the retrieval experiment shown in Figure 3, but with varying numbers of top-ranked documents. The results are summarized in Figure 4. We observe that Weighted Optimal Mixture Model with two or more iterations of training is fairly insensitive to the number of top-ranked documents that are used in the estimation. The performance is always above the single-document baseline. In contrast to that, we see that unweighted models (uniform weights) perform significantly worse than the baseline, and furthermore their performance varies widely with the number of top-ranked documents used. We believe these results to be extremely encouraging, since they show that weighted mixture models are considerably more stable than unweighted models.

6 Conclusions and Future Work

In this paper we explored using Optimal Mixture Models to estimate topical models. We defined optimality in terms of assigning maximum likelihood to a given sample of text, and looked at two types of mixture models: set-based and weighted. We presented an original proof that it is not feasible to compute set-based optimal mixture models. We then showed that weighted mixture models are superior to set-based models, and suggested a gradient descent procedure for estimating weighted mixture models. Our experiments show weighted mixtures outperforming the baseline on a simple retrieval task, and, perhaps more importantly, demonstrate that weighted mixtures are relatively insensitive to the number of top-ranked documents used in the estimation.

In the course of this work we encountered a number of new questions that warrant further exploration. We would like to analyze in detail the relationship between Optimal Mixture Models proposed in this paper and other methods of estimating a topic model from a small sample. The two obvious candidates for this comparison are query models [11] and relevance models [14]. Both are very effective at estimating accurate topic models starting with a very short (2-3 word) sample. Optimal Mixture Models appear to be more effective with a slightly longer sample (200-300 words). Another question we would like to explore is the use of tempered or annealed gradient descent to prevent over-fitting to the target sample. Finally, we would like to explore in detail the impact of the length of the training sample on the quality of the resulting model.

Acknowledgments

The author would like to thank Micah Adler and W. Bruce Croft for numerous discussions which led to this work. This work was supported in part by the Center for Intelligent Information Retrieval, in part by the National Science Foundation under grant number EIA-9820309, and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsors.

References

1. J. Allan, J.Carbonell, G.Doddington, J.Yamron, and Y.Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop, pp194-218*, 1998.
2. D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. In *Machine Learning, vol.34*, pages 1–34, 1999.
3. A. Berger, R. Caruana, D.Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of SIGIR*, pages 192–199, 2000.
4. A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings on the 22nd annual international ACM SIGIR conference*, pages 222–229, 1999.
5. A. Berger and V. Mittal. OCELOT: a system for summarizing web pages. In *Proceedings of SIGIR*, pages 144–151, 2000.
6. S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the ACL*, 1996.
7. C. Cieri, D.Graff, M.Liberman, N.Martey, and S.Strassel. The TDT-2 text and speech corpus. In *Proceedings of the DARPA Broadcast News Workshop, pp 57-60*, 1999.
8. D. Hiemstra. Using language models for information retrieval. In *PhD Thesis, University of Twente*, 2001.
9. T. Hoffmann. Probabilistic latent semantic indexing. In *Proceedings on the 22nd annual international ACM SIGIR conference*, pages 50–57, 1999.
10. H. Jin, R. Schwartz, S. Sista, and F. Walls. Topic tracking for radio, TV broadcast and newswire. In *Proceedings of DARPA Broadcast News Workshop, pp 199-204*, 1999.
11. J. Lafferty and C. Zhai. Document language models, query models and risk minimization for information retrieval. In *Proceedings on the 24th annual international ACM SIGIR conference*, pages 111–119, 2001.
12. J. Lafferty and C. Zhai. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings on the 24th annual international ACM SIGIR conference*, pages 111–119, 2001.
13. V. Lavrenko. Localized smoothing of multinomial language models. In *CIIR Technical Report IR-222*, 2000.
14. V. Lavrenko and W.B.Croft. Relevance-based language models. In *Proceedings on the 24th annual international ACM SIGIR conference*, pages 120–127, 2001.
15. D. Miller, T. Leek, and R. Schwartz. A hidden markov model information retrieval system. In *Proceedings on the 22nd annual international ACM SIGIR conference*, pages 214–221, 1999.
16. P. Ogilvie. Nearest neighbor smoothing of language models in ir. In *unpublished*, 2000.
17. J. Ponte. Is information retrieval anything more than smoothing? In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, pages 37–41, 2001.

18. J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings on the 21st annual international ACM SIGIR conference*, pages 275–281, 1998.
19. S. Robertson and K. S. Jones. Relevance weighting of search terms. In *Journal of the American Society for Information Science, vol.27*, 1977.
20. S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference*, pages 232–241, 1996.
21. S. E. Robertson. *The Probability Ranking Principle in IR*, pages 281–286. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1997.
22. M. Sipser. *Time Complexity: The Cook-Levin Theorem*, pages 254–260. PWS Publishing Company, Boston, 1997.
23. M. Sipser. *Time Complexity: The Subset Sum Problem*, pages 268–271. PWS Publishing Company, Boston, 1997.
24. F. Song and W. B. Croft. A general language model for information retrieval. In *Proceedings on the 22nd annual international ACM SIGIR conference*, pages 279–280, 1999.
25. H. Turtle and W. B. Croft. Efficient probabilistic inference for text retrieval. In *Proceedings of RIAO 3*, pages 644–651, 1991.
26. J. Yamron, I. Carp, L. Gillick, S.Lowe, and P. van Mulbregt. Topic tracking in a news stream. In *Proceedings of DARPA Broadcast News Workshop, pp 133-136*, 1999.

A Appendix

A.1 Lower bound on cross-entropy

In this section we prove that T itself is the unique optimal solution of equation (5). The proof is rudimentary and can be found in many Information Theory textbooks, but is included in this paper for completeness. Assuming T is in $\mathbb{P}$, we have to show that:

(i) for every $M \in \mathbb{P}$ we have $H(T|T) \leq H(T|M)$
(ii) if $H(T|T) = H(T|M)$ then $M = T$.

To prove the first assertion we need to show that the difference in cross-entropies $H(T|M) - H(T|T)$ is non-negative. This difference is sometimes referred to as *relative entropy* or *Kullback-Leiblar divergence*. By definition of cross-entropy:

$$H(T|M) - H(T|T) = -\sum_w T_w \log \frac{M_w}{T_w} \tag{14}$$

By Jensen's inequality, for any concave function $f(x)$ we have $E[f(x)] \leq f(E[x])$. Since logarithm is a concave function, we obtain:

$$-\sum_w T_w \log \frac{M_w}{T_w} \geq -\log \sum_w T_w \frac{M_w}{T_w} \tag{15}$$

Now we note that right-hand side of the above equation is zero, which proves **(i)**. In order to prove **(ii)** we recall that Jensen's inequality becomes an equality if and only if $f(x)$ is linear in x. The only way $\log \frac{M_w}{T_w}$ can be linear in $\frac{M_w}{T_w}$ is when $M_w = kT_w$, for some constant k. But since M and T are both in $\mathbb{P}$, we must have:

$$1 = \sum_w T_w = \sum_w M_w = \sum_w kT_w = k \sum_w T_w$$

Hence $k = 1$, and therefore $M = T$, which proves **(ii)**.

A.2 3SAT Definition

3SAT is a problem of determining satisfiability of a logical expression. An instance of 3SAT is a logical formula F in conjunctive normal form, with clauses limited to contain three primitives. The formula is a conjunction (an AND) of m clauses, each of which is a disjunction (an OR) of three variables from the set $\{x_1 ... x_n\}$, or their negations $\{\neg x_1 ... \neg x_n\}$. An example of such formula may be: $F = (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_4)$. Every formula of propositional calculus can be represented as a 3SAT formula. The task is to determine whether F is satisfiable. The formula is satisfiable if there exists an assignment of true/false values to the variables $x_1 ... x_n$, which makes the whole formula true. In the given example, setting $x_1 = False$ and all other variables to $True$ satisfies the formula.

A.3 POMM is NP-hard

In this section we will show that solving Pooled Optimal Mixture Model problem is NP-hard. In order to do that, we define a corresponding decision problem EXACT-POMM, prove EXACT-POMM to be NP-hard, and use the results in section 2.2 to assert that POMM itself is NP-hard.

In a nutshell, EXACT-POMM is a problem of testing whether a target distribution T is a member of the pooled mixture model set $\mathbb{P}_{\mathcal{C},pool}$ induced by some collection $\mathcal{C}$ of text samples. Let $\mathcal{M}$ denote the collection of integer-valued *count vectors* corresponding to each text sample in $\mathcal{C}$. Each element of $\mathcal{M}$ is a vector M_j that specifies how many times every word w occurs in $\mathcal{T}_j$, the corresponding text sample in $\mathcal{C}$, i.e. $M_{j,w} = \#w(\mathcal{T}_j)$. The task is to determine whether there exists a subset $S \subset \mathcal{M}$, such that vectors in S, when added up, form the same distribution as T:

$$T_w = \frac{\sum_{M_j \in S} M_{j,w}}{\sum_w \sum_{M_j \in S} M_{j,w}} \quad for\ \ all\ \ w \in W \tag{16}$$

We will prove that EXACT-POMM is NP-hard by reduction from 3SAT, a well-known problem of the NP-complete class [22]. We describe a polynomial time reduction that converts an instance of 3SAT into an instance of EXACT-POMM, such that 3SAT instance is satisfiable if and only if EXACT-POMM instance has a positive answer. The reduction we describe is very similar to the one commonly used to prove that SUBSET-SUM problem is NP-hard. We are given a formula in conjunctive normal form: $F = (p_{1,1} \vee p_{1,2} \vee p_{1,3}) \wedge \ldots \wedge (p_{m,1} \vee p_{m,2} \vee p_{m,3})$, where each proposition $p_{j,k}$ is either some variable x_i or its negation $\neg x_i$. We want to construct a collection of vectors $\mathcal{M}$ and a target distribution T such that F is satisfiable if and only if there exists a subset $S \subset \mathcal{M}$ of vectors which satisfies equation (16). Let $\mathcal{M}$ be the set of rows of the matrix in Figure 5, let T' be the target row at the bottom, and set the target distribution $T_w = T'_w / \sum_w T'_w$ for all w.

In Figure 5, $I_{n \times n}$ denotes a n by n identity matrix. $POS_{n \times m}$ is a n by m positive variable-clause adjacency matrix, that is $POS_{i,j} = 1$ when clause j contains x_i, and $POS_{i,j} = 0$ otherwise. Similarly, $NEG_{n \times m}$ is a n by m negative variable-clause adjacency matrix: $NEG_{i,j}$ is 1 if clause j contains $\neg x_i$, and 0 if it does not. $0_{2m \times n}$ is just a $2m$ by n all-zero matrix.

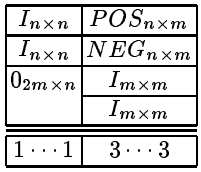

Fig. 5. Matrix used in the proof that 3SAT is reducible to EXACT-POMM

Note that the matrix is identical to the one that is used to prove that the SUBSET-SUM problem [23] is NP-hard by reduction from 3SAT. In line with the SUBSET-SUM argument, it is easy to see that the formula F is satisfiable if and only if there exists a subset $S \subset \mathcal{M}$ of the rows such that $T'_w = \sum_{M_j \in S} M_{j,w}$ for all w. The first n components of target $(T'_1 \ldots T'_n)$ ensure that for every variable i either x_i is true or $\neg x_i$ is true, but not both. The last m components $(T'_{n+1} \ldots T'_{n+m})$ ensure that every clause has at least one proposition set to true.

To demonstrate that satisfiability of F reduces to EXACT-POMM, we just need to show that:

$$\left[T'_w = \sum_{M_j \in S} M_{j,w}\right] \Leftrightarrow \left[T_w = \frac{\sum_{M_j \in S} M_{j,w}}{\sum_w \sum_{M_j \in S} M_{j,w}}\right] \tag{17}$$

The forward implication is trivial, since $T'_w = \sum_{M_j \in S} M_{j,w}$ clearly implies that $\sum_w T'_w = \sum_w \sum_{M_j \in S} M_{j,w}$, and therefore

$$T_w = \frac{T'_w}{\sum_w T'_w} = \frac{\sum_{M_j \in S} M_{j,w}}{\sum_w \sum_{M_j \in S} M_{j,w}} \tag{18}$$

The converse is not obvious, since the equality of two ratios in equation (18) does not by itself guarantee that their numerators are equal, we have to prove that their denominators are equal as well. The proof is as follows. Assume equation (18) holds. Let k be a (constant) ratio of the denominators:

$$k = \frac{\sum_w \sum_{M_j \in S} M_{j,w}}{\sum_w T'_w} \tag{19}$$

Then we can re-write equation (18) as follows:

$$kT'_w = \sum_{M_j \in S} M_{j,w}, \quad for\ all\ w \tag{20}$$

Now, for the left side of the matrix in Figure 5 ($w = 1 \ldots n$), we have $T_w = 1$, and $\sum_w M_{j,w}$ can be 0, 1 or 2. Therefore it must be that k is either 1 or 2, otherwise we cannot satisfy equation (20). Similarly, for the right side of the matrix ($w = n + 1 \ldots n+m$) we have $T_w = 3$, and $\sum_w M_{j,w}$ can be 0, 1, 2, 3, 4 or 5. Therefore, to satisfy equation (20), k must take a value in the set $\{\frac{1}{3}, \frac{2}{3}, \frac{3}{3}, \frac{4}{3}, \frac{5}{3}\}$. Since equation (20) must be satisfied for all w, k must take a value in the intersection $\{1, 2\} \cap \{\frac{1}{3}, \frac{2}{3}, \frac{3}{3}, \frac{4}{3}, \frac{5}{3}\}$.

Therefore $k = 1$, which means that the denominators in equation (18) are equal, and the converse implication holds.

The matrix in Figure 5 is of size $(2n + 2m)$ by $n + m$, therefore we have a polynomial-time reduction from an instance of 3SAT to an instance of EXACT-POMM, such that a 3SAT formula is satisfiable if and only if a corresponding instance of EXACT-POMM is satisfiable. Accordingly, if a polynomial-time algorithm exists for solving EXACT-POMM problems, this algorithm could be used to solve 3SAT in polynomial time. Thus EXACT-POMM is NP-hard.

We have just demonstrated that EXACT-POMM is NP-hard. In section 2.2 we demonstrated a reduction from decision problems (like EXACT-POMM) to optimization problems (like POMM). Therefore POMM is NP-hard.

A.4 AOMM is NP-hard

In this section we will prove that solving AOMM problem is NP-hard. The proof follows the same outline as the proof in section A.3. We define a corresponding decision problem EXACT-AOMM, prove it to be NP-hard by reduction from 3SAT, then use the result of section 2.2 to assert that AOMM is NP-hard.

AOMM is defined as an optimization problem over $I\!P_{\mathcal{C},avg}$ with respect to a given target T. Correspondingly, EXACT-AOMM is a problem of determining whether T itself is an element of $I\!P_{\mathcal{C},avg}$. An instance of EXACT-AOMM is a target distribution $T \in I\!P$, and a finite collection of empirical distributions $\mathcal{M} = \{M_j \in I\!P\}$ corresponding to a set of text samples $\mathcal{C}$. The task is to determine whether there exists a subset $S \in \mathcal{M}$, such that the average of elements in $M_j \in S$ is exactly the same as T:

$$T_w = \frac{1}{\#(S)} \sum_{M_j \in S} M_{j,w} \quad for\ \ all\ \ w \in W \tag{21}$$

We now present a reduction from 3SAT to EXACT-AOMM. We are given F, an instance of 3SAT, and want to construct a set of distributions $\mathcal{M}$ together with a target T, such that F is satisfiable if and only if there exists a subset $S \in \mathcal{M}$ which satisfies equation (21). The construction of EXACT-AOMM is similar to EXACT-POMM, but the proof is complicated by the fact that we are now dealing with distributions, rather than count vectors. We cannot directly use the matrix constructed in Section A.3 (Figure 5), because the rows may have varying numbers of non-zero entries, and if we normalize them to lie in $I\!P$, the proof might fail. For example, suppose we normalize the rows of the matrix in Figure 5. Then a 1 in a row from the upper half of that matrix becomes $1/m$ (where m is the the number of non-zero entries in that row, which will be greater than 1 for any variable x_i that occurs in any of the clauses in F). At the same time, a 1 in the lower half of the matrix remains a 1, since every row in the lower half contains exactly one non-zero entry. To alleviate this problem, we augment the matrix from Figure 5 in such a way that every row has the same number of non-zero entries. This is non-trivial, since we have to augment the matrix in such a way that satisfiability of 3SAT is not violated.

Let $\mathcal{M}'$ be the set of rows of the matrix in Figure 6, and let T' be the target (bottom) row. Here sub-matrices I, 0, POS and NEG have the same meaning as in Section

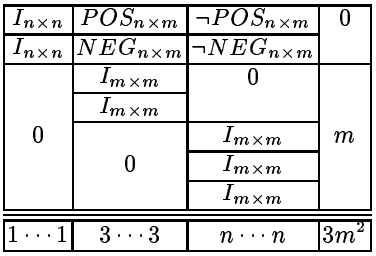

Fig. 6. Matrix used in the proof that 3SAT is reducible to EXACT-AOMM

A.3. In addition, $\neg POS$ is a logical negation of POS, i.e. $\neg POS_{i,j} = 1$ whenever $POS_{i,j} = 0$, and vice-versa. Similarly, $\neg NEG$ is a negation of NEG. The last column of the matrix contains zeros in the top $2n$ rows and m in the bottom $3m$ rows. Note that the new matrix contains the tableau constructed in Section A.3 as a sub-matrix in the upper-left corner. The columns to the right of the original tableau ensure that the sum of every row in the matrix is exactly $m + 1$. The first $n + m$ positions of the target row are also identical to the target row for the POMM problem.

We define a corresponding instance of EXACT-AOMM by setting $T = T'/(n + 3m + nm + 3m^2)$, and $\mathcal{M} = \{M'_j/(m + 1) : M'_j \in \mathcal{M}'\}$, dividing each row by the sum of its elements to ensure that the result is in $\mathbb{P}$. Now we want to prove that F is satisfiable if and only there exists a subset $S \subset \mathcal{M}$ which satisfies equation (21).

Let $S \subset \mathcal{M}$ be the subset that satisfies equation (21). We assert that $\#(S)$, the size of S, must be $(n + 3m)$. The proof is as follows:

- **(i)** The sum of the first column over S can be either $\frac{1}{m+1}$ or $\frac{2}{m+1}$, and the target is $T_1 = \frac{1}{(n+3m)(m+1)}$. Therefore $\#(S)$ can be either $(n + 3m)$ or $2(n + 3m)$.
- **(ii)** The sum of the last column over S can be any one of $\frac{m}{m+1} \times \{1, 2, 3, \ldots, m, \ldots, 5m\}$, while the target is $\frac{3m^2}{(n+3m)(m+1)}$. Therefore $\#(S)$ can be any one of $\frac{n+3m}{3m} \times \{1, 2, 3, \ldots, m, \ldots, 5m\}$.

Since S must satisfy equation (21) for all w, we must have $\#(S)$ satisfy both **(i)** and **(ii)**, and therefore it must be that $\#(S) = (n+3m)$. Now, since by definition $M_{j,w} = \frac{M'_{j,w}}{m+1}$ and $T_w = \frac{T'_w}{(n+3m)(m+1)}$, we can claim that:

$$\left[T_w = \frac{1}{\#(S)} \sum_{M_j \in S} M_{j,w}\right] \Leftrightarrow \left[T'_w = \sum_{M'_j \in S'} M'_{j,w}\right] \tag{22}$$

Here S' is a subset of the rows that corresponds to S. It remains to show that F (an instance of 3SAT) is satisfiable if and only if there exists a subset $S' \in \mathcal{M}'$ which satisfies the right hand side of equation (22).

($\Leftarrow$) If such S' exists, columns $\{1 \ldots n\}$ of $\mathcal{M}'$ guarantee that we have a proper truth assignment, and columns $\{n + 1 \ldots n + m\}$ guarantee that every clause of F has at least one true proposition, thus F is satisfied.

($\Rightarrow$) If F is satisfiable, it is clear that right hand side of equation (22) holds for columns $\{1 \ldots n+m\}$ (e.g. by the the SUBSET-SUM argument). Each one of the columns $\{n+1 \ldots n+m\}$ contained 1, 2, or 3 non-zero entries (in the first $2n$ rows), and therefore required 2, 1 or 0 "helper" variables to add up to 3. Because columns $\{n+m+1 \ldots n+m+m\}$ represent a logical negation of columns $\{n+1 \ldots n+m\}$, they will contain $(n-1)$, $(n-2)$ or $(n-3)$ non-zero entries respectively and may require 1, 2, or 3 "helper" variables to sum to n. Finally, the last column will reflect the total number of "helpers" used for each of the m clauses, which will always be 3 for every clause: either $(2+1)$, or $(1+2)$, or $(0+3)$. Since there are m clauses, the last column will sum up to $3m \times m$.

This proves that F is satisfiable if and only if the right hand side of equation (22) is satisfiable, which in turn holds if and only if left-hand side (EXACT-AOMM) is satisfiable. The matrix in Figure 6 is of size $(2n+5m)$ by $(n+2m+1)$, thus we have a polynomial-time reduction from an instance of 3SAT to an instance of EXACT-AOMM. Thus EXACT-AOMM is NP-hard, and by the result in section 2.2, AOMM is NP-hard as well.