IMPROVEMENT OF NON-NEGATIVE MATRIX FACTORIZATION BASED LANGUAGE MODEL USING EXPONENTIAL MODELS

Miroslav Novak

IBM T.J. Watson Research Center
P. O. Box 218, Yorktown Heights, NY 10598, USA
miroslav@us.ibm.com

Richard Mammone

Dept. of ECE, Rutgers University
Piscataway, NJ 08834, USA
mammone@caip.rutgers.edu

ABSTRACT

This paper describes the use of exponential models to improve Non-negative Matrix Factorization (NMF) based topic language models for Automatic Speech Recognition. This modeling technique borrows the basic idea from Latent Semantic Analysis (LSA), which is typically used in Information Retrieval. An improvement was achieved when exponential models were used to estimate the *a posteriori* topic probabilities for an observed history. This method improved the perplexity of the NMF model, resulting in a 24% perplexity improvement overall when compared to a trigram language model.

1. INTRODUCTION

The NMF approach to topic language modeling has been introduced recently [1]. The concept is similar to the Singular Value Decomposition method [2] in the way the challenges of topic dependent modeling are approached. In particular, finding an optimal segmentation of the training corpus into topic, sparseness of the training data caused by this segmentation, and, finally, finding a method for topic assignment during the recognition. There is a trade-off in topic dependent model design: the finer the distinction between topics, the greater the number of topics used and the smaller the amount of data available to train each topic. Use of exponential models has been proposed [3],[4],[5] to reduce this effect as parameters of only those words which are important to a particular topic are adapted, but there is still a need for explicitly defined set of topics. Cache language models [6] do not require such a set, the recent history is used directly to adapt the language model parameters, without explicitly determining the current topic. But due to the small amount of the data seen in the history, such adaptation is usually performed on a few unigram statics only.

The Latent Semantic Models (both SVD and NMF based) can be viewed as an extension of the cache models. If $\mathcal{Q}$ is the space of all possible unigram distributions on a vocabulary $\mathcal{V}$, the idea is to find a linear subspace $\mathcal{S} \subset \mathcal{Q}$: rank($\mathcal{S}$) $\ll$ rank($\mathcal{Q}$) which can most accurately span all unigram distributions observed in a large training corpus. Local unigram distributions are typically obtained by partitioning the training corpus into a large set of relatively small documents where each model provides a sample of the distribution. The spanning vectors of $\mathcal{S}$ are obtained by lower rank approximation of the matrix representing local word dependencies (typically represented as word-document occurrence counts). During the recognition, the observed history is used to determine a particular distribution from $\mathcal{S}$ by finding an appropriate linear combination of the spanning vectors. The number of estimated parameters (weights of the spanning vectors) is given by the rank of the subspace, which is much smaller than number of parameters in a completely unrestricted unigram model.

Let us illustrate this method on an example. Consider two words, "stocks" and "bonds". If articles related to financial news are included in the training corpus, one of the basis vectors will most likely point in the directions of both words, suggesting that these words tend to appear together. Let us assume the word "stocks" appears in the history, but "bonds" does not. A cache model would increase the probability of "stock" only. An LSA model would increase the weight of the basis vector, so the probabilities of both words would be increased.

2. NMF MODEL

A distinct property of the NMF technique in comparison to SVD is that the elements of the vectors spanning the probability subspace are non-negative. This facilitates the interpretation of the NMF spanning vectors as probability distributions. An iterative algorithm to find the non-negative factorization of the word-document matrix has been presented [7]. The computational costs associated with each iteration are indirectly proportional to the spareness of the factored matrix, a very desirable feature in our situation.

The method described in more detail in [1] will be used. A square matrix describing local word dependencies as word co-occurrence counts is constructed . One advantage of this approach is that there is no need for the training corpus to be explicitly segmented into documents. Another advantage is that the low count elements in the word-word co-occurrence matrix can be replaced by zero, which reducing the number of computations associated with each iteration.

The co-occurrence matrix can be factored into:

$$A_{V\times V} \approx Q_{V\times R} Z_R Q^T_{V\times R}, \tag{1}$$

where V is the vocabulary size and R is the number of latent topics, typically $R << V$. All elements of A, Q and Z are non-negative. We can interpret the columns of Q as topic dependent unigram distributions and diagonal elements of the diagonal matrix Z as topic priors $Z(t)$. It should be noted that the spanning vectors are neither orthogonal nor independent. But we have observed that for $R << V$, most of the vectors are independent, since each vector contains non-zero probabilities for a different set of words.

Having the set of conditional probabilities $P_{NMF}(w|t) = Q_{wt}$, a unigram word probability distributions dependent on the observed history $P(w|H)$ can be constructed as a weighted mixture

$$P(w|H) = \sum_t P(w|t)P(t|H). \tag{2}$$

In the next section, we will introduce a new method for estimation of $P(t|H)$.

3. TOPIC WEIGHTS ESTIMATION

In this section we will describe a method for estimation of topic weights $P(t|H)$. A history instance is a particular sequence of words $H_n = (w_{n-k}, w_{n-k+1}, \ldots, w_{n-1})$. Let $\mathcal{H}^n$ be the set of words which appear in the history H_n and the probability of $P_{NMF}(w|t)$ is non-zero for at least one topic t (excluding the most frequent function words). We will simplify the notation by omitting the index n further, since the algorithm is applied in the same way to each instance of a new history. Let w_i^H be a particular word in H.

As a baseline, the method of weights estimation described in [1] is used. We assume that the observed history can be generated by any of the topics, independently one word at a time. We can then express the contribution of a particular word in the history to the conditional weight of each topic as

$$\mu_{ij} = \frac{P(w_i^H|t_j)P(t_j)}{\sum_l P(w_i^H|d_l)P(d_l)}. \tag{3}$$

Then we can find the probability $P(t|H)$ as an average of normalized contributions:

$$P(t|H) = \frac{\sum_i \mu_{it}}{\sum_l \sum_i \mu_{il}} = \frac{\sum_i \mu_{it}}{|H|}, \tag{4}$$

where $|H|$ is size of the history. The reason for adding the probability contributions from each word in the history is that we look at them as counts, rather than probabilities. We are not able to use a product of those probabilities because, due to the sparseness of Q, there is almost always at least one $P(w_i^H|t) = 0$.

It can be seen that, for all topics which assign very low or zero probability to all words in a particular instance of history, the probability $P(t|H)$ can be zero as well. Thus some smoothing technique should be used to improve the robustness of this model.

The minimum divergence framework can be used. In this case, we are looking for the set of weights model which has the lowest distance from the *a priori* weights $Z(t)$:

$$D(P(t|H), Z(t)) = \sum_t P(t|H) \log \frac{P(t|H)}{Z(t)}, \tag{5}$$

and which satisfies the constraints:

$$\begin{gathered} \sum_t P(t|H) f_w(t) = \hat{E}[f_w], \\ \sum_t P(t|H) = 1. \end{gathered} \tag{6}$$

The solution to this constraint optimization problem can be shown to have a form:

$$P(t|H) = \frac{1}{M} Z(t) e^{\sum_{w\in\mathcal{V}} \lambda_w f_w(t)}, \tag{7}$$

for set of words w selected from the vocabulary, where $M = \sum_t Z(t) e^{\sum_{w\in\mathcal{V}} \lambda_w f_w(t)}$.

We need to make choice of the feature functions $f_w(t)$ and estimate their target expectations $\hat{E}[f_w]$. We would like to utilize all the information available: the observed history tells us that certain words have higher probability of occurrence than the global unigram distribution suggests, and the NMF model provides us with a set of topic dependent unigram distributions. One strategy is to define the feature functions using these topic distributions, find their expected values, and then find the relationship between the target expectations and the history.

Let us consider a choice of the feature functions:

$$f_w(t) = P_{\text{NMF}}(w|t). \tag{8}$$

We will omit the NMF index in further text, we will always assume that the topic conditioned word probabilities are obtained from the NMF model. For this choice, it can be seen that:

$$\sum_{w\in\mathcal{V}} f_w(t) = 1, \tag{9}$$

and the Generalized Iterative Scaling (GIS) [8] algorithm can be used to find solution. The update formulas for $\lambda_w^{(k+1)}$

at the k-th iteration are:

$$\lambda_w^{k+1} = \lambda_w^k + \log \frac{\hat{E}[f_w]}{\frac{1}{M}\sum_t Z(t) e^{\sum_{w \in \mathcal{V}} \lambda_w^k f_w(t)} f_w(t)}. \quad (10)$$

It can be easily verified that:

$$E_t[f_w] = P(w), \quad (11)$$

so its estimate, the target value $\hat{E}[f_w]$, should be equal to $\hat{P}(w)$, i.e. normalized local word occurrence counts $n(w)/|H|$. Such a choice would assign zero target values $\hat{E}[f_w]$ to the words not observed in the history, which is apparently wrong. To remedy this situation, we use features for the words observed in the history only and add a new feature $f_c(t)$, representing all of the words not seen in the history. The new set of constraints is:

$$\begin{aligned} &\sum_t P(t|H) f_w(t) = \hat{E}[f_w] \qquad \forall w \in \mathcal{H}, \\ &\sum_t P(t|H) f_c(t) = \hat{E}[f_c], \\ &\sum_t P(t|H) = 1. \end{aligned} \quad (12)$$

Since the feature functions are probabilities, the following must be satisfied:

$$\sum_{w \in \mathcal{H}} E[f_w] + E[f_c] = 1, \quad (13)$$

$$\sum_{w \in \mathcal{H}} \hat{P}[w] + \hat{E}[f_c] = 1. \quad (14)$$

Condition (13) will be satisfied when:

$$\sum_{w \in \mathcal{H}} f_w(t) + f_c(t) = 1, \quad (15)$$

so the values of $f_c(t)$ can determined.

We introduce an additional assumption, based on the facts that the size of the history is much smaller than the vocabulary size and that the most frequent function words are excluded from the topic dependent model:

$$f_c(t) >> \sum_{w \in \mathcal{H}} P(w), \quad (16)$$

which allows us for the purpose of the target expectations estimation to consider the feature $f_c(t)$ to be a constant:

$$f_c(t) \approx \bar{f}_c, \quad (17)$$

so (15) and (14) can be rewritten in form which will be helpful in the selection of the target expectations.

$$\sum_{w \in \mathcal{H}} \hat{P}(w) + \bar{f}_c \approx \sum_{w \in \mathcal{H}} f_w(t) + \bar{f}_c. \quad (18)$$

We use a choice of

$$\sum_{w \in \mathcal{H}} \hat{P}(w) = \max_t \sum_{w \in \mathcal{H}} f_w(t), \quad (19)$$

which guarantees the non-negativity of the feature $f_c(t)$. We can now determine the rest:

$$\begin{aligned} \hat{E}[f_c] &= \bar{f}_c = \sum_{w \in \mathcal{H}} \hat{P}(w) \\ f_c(t) &= 1 - \sum_{w \in \mathcal{H}} f_w(t) \\ \hat{P}(w) &= \frac{n(w)}{\sum n(w)} \sum_{w \in \mathcal{H}} \hat{P}(w) \end{aligned} \quad (20)$$

We have further modified this method to improve the speed of convergence of the GIS algorithm. We can look at the feature $f_c(t)$ as filler feature, so we replace the condition 15 by:

$$f_c(t) + \sum_{w \in \mathcal{H}} f_w(t) = f_{max} \leq 1. \quad (21)$$

A filler feature is a common technique used to satisfy the requirement of GIS that for any value of the argument, the sum of all feature values is a constant. In our experiment, we choose the value of $\bar{f}_c$ and then determine the value of f_{max} from (21). The second equation of (20) then needs to be changed to:

$$f_c(t) = f_{max} - \sum_{w \in \mathcal{H}} f_w(t) \quad (22)$$

The described method may produce inconsistent constraints, since there is no guarantee that there is any distribution $P(t)$ which will produce the target expectations. Therefore, there should be a hard limit imposed on the number of iterations.

4. RESULTS

Experiments were performed in the context of the Aristotle project [9] conducted at CAIP. In this project, we used the IBM ViaVoice speech recognition system for automated transcription of recorded lectures. The recognizer's vocabulary was extended to cover the specific subject (Biology 101). The nature of the speech used in the lecture presentations is more spontaneous than in read speech and exhibits distinct content word patterns in contexts beyond the reach of trigrams.

A trigram and an NMF model [1] were trained on a biology course textbook (total 600K words, vocabulary 10k words). They were linearly interpolated with the trigram model for general English distributed with ViaVoice (vocabulary 60K words). We chose the number of latent topics to be $R = 300$. The resulting factor Q had 322k non-zero elements (when a threshold was applied).

Transcriptions of recorded biology lectures were divided into two sets. The first one (12K words), was used as a held-out set to estimate the value of $\bar{f}_c$. The second part (13K words) was used as a test set to measure the perplexity gain. The choice of $\bar{f}_c$ does not seem to be critical as far as the perplexity improvement is considered, as can be seen in figure 1. But it has a significant effect on the speed of the convergence, so lower values are desired. We have used $\bar{f}_c = 0.3$, which maximizes the perplexity gain on the held-out data. Table 1 shows perplexity gains for the original method of topic weight computation and the presented method with exponential models.

Perplexity	held-out	test set
trigrams only	227.58	257.64
trigrams & NMF	194.52	209.37
trigrams & epx. NMF	184.65	196.27

Table 1. Perplexity improvements

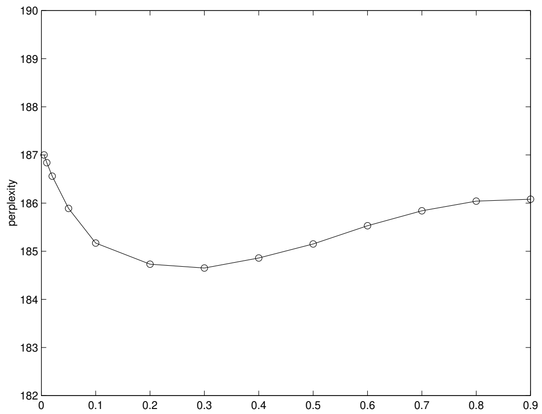

Fig. 1. Perplexity versus estimate $\bar{f}_c$

As far as the complexity is concerned, the convergence is rather slow (about 40 iterations are needed). But it should be noted that the number of constraints and thus number of parameters is small, typically less than ten, so the cost of each iteration is reasonable.

5. CONCLUSION

We have shown that the perplexity of the NMF model can be improved when exponential models are employed to estimate the topic weight probabilities. Use of this method results in a 6.2% improvement over the previously used method. The total perplexity gain of the NMF model over the trigram model is 24 %.

The cost associated with the use of NMF the model, particularly when the exponential model based topic weight estimation is used, prevented us from using the model directly in the speech recognition system. When the model was used for N-best rescoring, the recognition accuracy improvement was negligible. As a next step, we will try to use the NMF model more tightly in the recognition search algorithm.

6. REFERENCES

[1] M. Novak and R. Mammone, "Use of non-negative matrix factorization for language model adaptation in a lecture transcription task," in *Proceedings of ICASSP*, Salt Lake City, Utah, May 2001.

[2] J.R. Bellegarda, "A multispan language modelling framework for large vocabulary speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 5, pp. 456 – 467, September 1998.

[3] S. Della Pietra, V. Della Pietra, R.L. Mercer, and S. Roukos, "Adaptive language modeling using minimum discriminant estimation," *Proceedings of the Speech and Natural Language DARPA Workshop*, February 1992.

[4] S.F. Chen, K. Seymore, and R. Rosenfeld, "Topic adaptation for language modeling using unnormalized exponential models," in *Proceedings of ICASSP*, Seattle, Washington, 1998.

[5] S. Khudanpur and J. Wu, "A maximum entropy language model integrating n-grams and topic dependencies for conversational speech recognition," in *Proceedings of ICASSP*, Phoenix, Arizona, 1999.

[6] R. Kuhn and R. De Mori, "A cache-based natural language model for speech reproduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 570–583, 1990.

[7] D.D. Lee and H.S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 1451 – 1454, October 1999.

[8] J.N. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *Annals of Mathematical Statistics*, , no. 43, pp. 1470–1480, 1972.

[9] G. Faulkner, S. Gopal, A. Ittycheriah, R. Mammone, A. Medl, and M. Novak, "The aristotle project: A distributed learning system," in *Proceedings of Ed-Media2000, World Conference on Educational Multimedia, Hypermedia, and Telecommunications*, Montreal, Canada, June 2000, pp. 292–297.