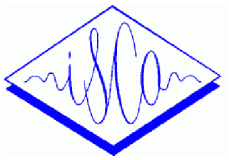

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

A Sparse Plus Low Rank Maximum Entropy Language Model

Brian Hutchinson, Mari Ostendorf, Maryam Fazel

Electrical Engineering Department, University of Washington, Seattle, WA, USA
{brianhutchinson,mo,mfazel}@ee.washington.edu

Abstract

This work introduces a new maximum entropy language model that decomposes the model parameters into a low rank component that learns regularities in the training data and a sparse component that learns exceptions (e.g. multiword expressions). The low rank component corresponds to a continuous-space language model. This model generalizes the standard ℓ_1-regularized maximum entropy model, and has an efficient accelerated first-order training algorithm. In conversational speech language modeling experiments, we see perplexity reductions of 2-5%.

Index Terms: language modeling, maximum entropy, sparse plus low rank decomposition

10.21437/Interspeech.2012-459

1. Introduction

While n-gram language modeling has been used with much success in applications where large training sets are available, there remains a need to better model small, targeted domains where limited data is available. We recently introduced a tensor-based "low rank language model" (LRLM) that outperforms baseline models when training data is limited [1]. A disadvantage of the LRLM is that the non-convex objective complicates training. The model proposed in this paper introduces a low rank weight component into the maximum entropy modeling framework, side-stepping the limitations of the LRLM and leveraging advantages of the maximum entropy approach. Our new model subsumes a large class of ℓ_1-regularized maximum entropy language models and can be interpreted as incorporating a discriminatively-trained continuous-space language model into the maximum entropy setting.

2. Sparse and Low Rank Language Models

2.1. The Model

The standard maximum entropy language model [2] has the form

$$p(w|h) = \frac{\exp\left(a^T f(w,h)\right)}{\sum_{w'} \exp\left(a^T f(w',h)\right)}, \quad (1)$$

where $a \in \mathbb{R}^d$ are the parameters and $f(w,h) \in \mathbb{R}^d$ is the feature vector extracted from word w and history h. We generalize the model with two key changes: i) the vector is recast as a feature matrix with a corresponding weight matrix, and ii) the weight matrix is decomposed into the sum of two matrices that each have special structure. A weight matrix parameterized maximum entropy model can be written as:

$$p(w|h) = \frac{\exp\left(\psi(w)^T A \phi(h)\right)}{\sum_{w'} \exp\left(\psi(w')^T A \phi(h)\right)}. \quad (2)$$

Here $\psi(w) \in \mathbb{R}^{d_\psi}$ is the vector of features of w individually and $\phi(h) \in \mathbb{R}^{d_\phi}$ are the features of h. In this case $A \in \mathbb{R}^{d_\psi \times d_\phi}$ is a parameter *matrix*. This can be linked to the notation of Eqn. 1 by noting that $\psi(w)^T A \phi(h) = \langle A, \psi(w)\phi(h)^T \rangle$, where $\langle \cdot, \cdot \rangle$ denotes a matrix inner-product (element-wise multiply and sum). If one denotes the vectorization of A with a and the vectorization of feature matrix $\psi(w)\phi(h)^T$ with $f(w,h)$, we recover the form in Eqn. 1. Any maximum entropy language model whose features are products of features on words and histories can be written in the form of Eqn. 2. In particular, standard n-gram features take this form, where $\psi(w)$ and $\phi(h)$ are one-hot-encodings of the words and histories. In this paper we focus on these standard n-gram features.

Empirically we observe that the weight matrices A learned for models of the form Eqn. 2 contain two qualitatively different kinds of information. First, there are relatively dense regions of the matrix that model the sequential behavior of high frequency words. Because only a small fraction of the words are frequent, this information can be well modeled by a low rank matrix. Second, there are large sparse regions of the matrix, where only a handful of the elements deviate significantly from zero - these correspond to n-grams whose individual words infrequently appear outside of a small handful of n-grams (e.g. "san francisco"). This information alone can be well modeled by a sparse matrix. Fig. 1 illustrates this result, visualizing the estimated weights in the 200×200 leading submatrix of a bigram weight matrix trained on 100K tokens with a 5K vocabulary. The complete matrix can be decomposed accurately into the sum of a sparse matrix (S, upper) and low rank matrix (L, lower). We show in this paper that language modeling performance can be improved by using a more general model able to efficiently capture both types of structure inherent in the data.

2.1.1. Sparse Component

The model of Eqn. 1 is often trained with ℓ_1 regularization applied to a. Not only is it well-known that this particular penalty leads to *sparse* solutions, but it has also been found to be an empirically desirable criterion to minimize [3]. An entry-wise ℓ_1 penalty can be applied to a weight matrix S to the same effect. In standard models, the "sparse" component is the only component, and is thus tasked with modeling all of the sequential behavior. In our model, its burden is reduced (e.g. it does not need to model n-grams easily explained by syntax), freeing it to focus on n-grams that do not fall into standard patterns.

2.1.2. Low Rank Component

Restricting ourselves to a sparse solution ignores an important attribute of language: that similarities exist between words and between histories in the data. A sparse model has no way to exploit similarities that might exist (e.g. between the words "bicycle" and "bike"). Viewed in the form of Eqn. 1, this is inevitable: features are values at arbitrary positions in a vector. Viewed in the form of Eqn. 2, we see similarities between two

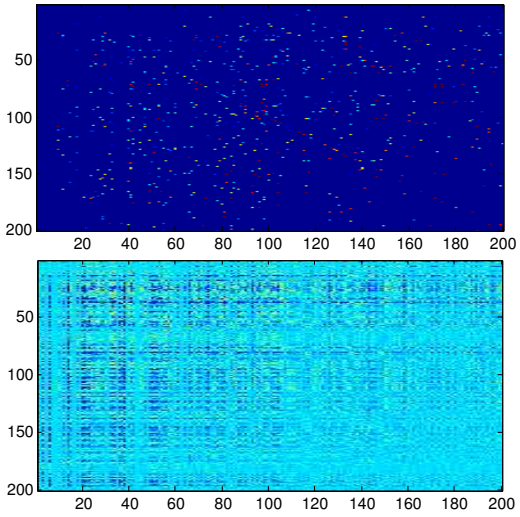

Figure 1: The weight matrix A can be naturally decomposed into sparse (top) and low rank (bottom) matrices.

words can be expressed by similarity between the corresponding rows, and similarities between histories can be viewed as similarities between the corresponding columns. More generally, sets of rows (or columns) may live in subspaces, e.g. one might envision a "space" of adjectives, or a "space" of nouns. This property corresponds to a *low rank* solution; i.e. finding a low rank weight matrix A.

Empirically, a low rank component typically appears in the solution without any encouragement from the model or training. By facilitating the existence of a low-rank component, we can improve the modeling performance. For example, the co-occurrence statistics for a word α that has been observed 50 times may be sufficiently similar to a set of words β that have been observed hundreds or thousands of times for α's weights to be pushed into β's subspace of weights; in effect, this "fills in" missing entries from α's weight rows and columns.[1]

The idea of learning and exploiting similarities between objects (e.g. words and histories) is a common theme in the literature on learning shared representations [5] and is used by language models with continuous representations of words [6, 7]. We show here that a maximum entropy model with a low rank weight matrix A is in fact a continuous-space language model. To see this, note that $A \in \mathbb{R}^{d_\psi \times d_\phi}$ with rank $R \leq \min(d_\psi, d_\phi)$ has a singular value decomposition $A = U\Sigma V^T$, with diagonal matrix of singular values $\Sigma \in \mathbb{R}^{R\times R}$, left singular vectors $U \in \mathbb{R}^{d_\psi \times R}$ and right singular vectors $V \in \mathbb{R}^{d_\phi \times R}$. Substituting the structure of A into Eqn. 2, we get

$$p(w|h) = \frac{1}{Z(h)} \exp\left(\psi(w)^T U\Sigma V^T \phi(h)\right) \quad (3)$$

$$= \frac{1}{Z(h)} \exp\left(\left(U^T\psi(w)\right)\Sigma\left(V^T\phi(h)\right)\right) \quad (4)$$

$$= \frac{1}{Z(h)} \exp\left(\tilde{\psi}(w)^T\Sigma\tilde{\phi}(h)\right). \quad (5)$$

Here $\tilde{\psi}(w) = U^T\psi(w)$ denotes a continuous, low-dimensional representation of w, $\tilde{\phi}(w) = V^T\phi(h)$ denotes a continuous, low-dimensional representation of h, and $Z(h)$ is the normalizing factor. The probability of a word following a history is proportional to the weighted inner-product $\tilde{\psi}(w)^T\Sigma\tilde{\phi}(h)$ in the low-dimensional space. The dimension of the representation is equal to R, the rank; lower rank solutions for A correspond to embeddings of words in lower dimensional spaces. Similar words will have continuous representations that are close to each other in the low-dimensional space; the same is true for similar histories. Crucially, the low dimensional representation of a word can (and should) be different in the history position h than in the predictive position w. In Section 2.2 we present an algorithm that discriminatively learns a low rank matrix, and thus discriminatively learns low dimensional continuous representations of words and histories.

When A is low rank, the model of Eqn. 2 bears some similarity to Mnih and Hinton's "log-bilinear" language model [8], which estimates a matrix that is analogous to our U in Eqn. 4, and matrices analogous to V for each word in a fixed history window. In doing so, they find continuous low-dimensional representations of words and history words. A few advantages of our approach are that 1) we are guaranteed to converge to a globally optimal solution, 2) we support arbitrary feature functions of words and histories, and 3) the dimension of the hidden representation is learned, rather than pre-specified.

2.1.3. Sparse and Low Rank Combination

It is not plausible that all regularities in the data can be learned from a finite training set. For example, if a word is observed only a handful of times, we may simply not know enough about it to know what subspace of words it lives in. Further, some n-grams (e.g. proper nouns, idioms, etc.) do not fit into any regular pattern. A low rank matrix cannot capture all of these exceptions. Thus we propose a hybrid model, where the weight matrix A is the sum of two individual components: a low rank matrix L and a sparse matrix S. The low rank component is free to model all of the regularities present in the data (a result of the inherent structure present in language). The sparse component learns the rest - the exceptions to the rule. Our proposed sparse plus low rank language model (SLR-LM) thus has the following form:

$$p(w|h) = \frac{\exp\left(\psi(w)^T(S+L)\phi(h)\right)}{\sum_{w'}\exp\left(\psi(w')^T(S+L)\phi(h)\right)}. \quad (6)$$

As a byproduct, the SLR-LM separates n-grams into two qualitatively different sets: the regular low rank n-grams which are well predicted by the regular rules, and the sparse n-grams that are not. There are auxiliary benefits to this decomposition. For example, Min *et al.* [9] show that applying a sparse plus low rank decomposition directly to a word-document matrix can be used to extract document keywords; the n-grams in our sparse matrix could be used similarly.

2.2. Training Algorithm

To train the SLR-LM of Eqn. 6 we solve the following non-smooth convex optimization problem:

$$\min_{S,L} \gamma_0\|L\|_* + \gamma_1\|S\|_1 + \gamma_2\|S+L\|_F^2 - \mathcal{L}(\mathcal{X}; S, L). \quad (7)$$

The entry-wise ℓ_1 norm, $\|S\|_1$, promotes sparsity in our S variables. We penalize L's nuclear norm,[2] $\|L\|_*$, which is known

[1] This basic idea is extensively exploited in low-rank approaches to collaborative filtering [4].

[2] The nuclear norm of X is the sum of its singular values; i.e. the ℓ_1 norm applied to the vector of singular values.

Algorithm 1 TRAINSLR-LM()

$S \leftarrow L \leftarrow S' \leftarrow L' = 0; t \leftarrow t' \leftarrow 1$
while not converged **do**
  $mL \leftarrow L' + ((t'-1)/t)(L - L')$
  $mS \leftarrow S' + ((t'-1)/t)(S - S')$
  $mA \leftarrow mL + mS$
  Pick τ_S
  $gS \leftarrow mS + (1/\tau_S)\nabla_{mA}(\mathcal{L} - \gamma_2\|mA\|_F^2)$
  $pS \leftarrow \mathcal{S}_{\gamma_1/\tau_S}(gS)$
  $S' \leftarrow S$ and $S \leftarrow pS$
  $sA \leftarrow mL + pS$
  Pick τ_L
  $gL \leftarrow mL + (1/\tau_L)\nabla_{sA}(\mathcal{L} - \gamma_2\|sA\|_F^2)$
  $[U, \Sigma, V] \leftarrow \text{SVD}(gL)$
  $pL = U\mathcal{S}_{\gamma_0/\tau_L}(\Sigma)V^T$
  $L' \leftarrow L$ and $L \leftarrow pL$
  $t' \leftarrow t$ and $t \leftarrow (1 + \sqrt{1 + 4t^2})/2$
end while

to encourage low rank solutions [10, 11]. The Frobenius norm permits standard ℓ_2-norm regularization; it is included for completeness but is not used in experiments here. $\mathcal{L}$ denotes the average log-likelihood, which has the familiar empirical-minus-model expectation form for its gradient as a function of A:

$$\nabla_A \mathcal{L} = E_{\hat{p}(w,h)}[\psi(w)\phi(h)^T] - E_{p_A(w,h)}[\psi(w)\phi(h)^T]. \quad (8)$$

In Alg. 1 we introduce an iterative algorithm for solving the above convex optimization problem. The basic structure of each iteration is to make four updates: 1) a gradient step on S, 2) an entry-wise threshold step to shrink the entries of S, 3) a gradient step on L, and 4) a singular-value threshold step on L. Both thresholding steps make use of the soft-thresholding operator:

$$\mathcal{S}_\mu(X) = \text{sgn}(X) \circ \max(0, |X| - \mu) \quad (9)$$

where all operations are entry-wise; in particular, $\circ$ denotes entry-wise multiplication. Our algorithm is a block-coordinate variant of the accelerated proximal gradient descent algorithm introduced by Toh and Yun in [12], modified to alternate between the ℓ_1 and $\|\cdot\|_*$ regularized terms. Compared to standard maximum entropy language model training, the SVD increases the computational cost, adding a $O(d_\psi d_\phi R_k)$ term to the per-iteration complexity, where R_k is the rank of L at iteration k. To speed up training, we employ many of the tricks proposed in [12], including computing partial SVDs (with PROPACK [13]) and using a continuation technique that gradually decreases the γ_0 and γ_1 weights from a large initial value to their intended target values. The τ_S and τ_L parameters in Alg. 1 are picked according to a line search analogous to Toh and Yun's. All training computation can be phrased as matrix operations, permitting us to locally parallelize computation over many cores.

3. Language Modeling Experiments

3.1. Experimental details

All of our experiments use conversational telephone conversations from the Fisher corpus [14]. Each conversation in the corpus is labeled by topic; we draw data from eight of the largest topics (see Tab. 1), which had at least 350K word tokens. For each topic, we split the data (at the granularity of conversation) into training, development and test sets, at a 60/20/20 ratio. To avoid conflating the effects of topic and training set size, after basic text normalization we subsampled the training data (by sentence) to create three training subsets per topic, with 200K, 100K and 25K training tokens, respectively. Due to limited training data, we restrict our vocabulary to the most frequency 5K word types; all out-of-vocabulary tokens are mapped to a dedicated OOV symbol.

Topic	# Train Tokens	Test OOV	Description
ENG01	158K	3.5%	Professional sports
ENG02	157K	4.4%	Pets
ENG03	130K	2.9%	Life partners
ENG04	151K	3.2%	Minimum wage
ENG05	131K	3.9%	Comedy
ENG24	63K	3.6%	September 11
ENG30	76K	3.8%	Family
ENG37	81K	3.3%	Reality TV

Table 1: Topics used from the Fisher corpus.

	25K		100K		200K	
Topic	SLR	ME	SLR	ME	SLR	ME
ENG01	109.4	109.9	83.0	86.5	76.7	78.8
ENG02	115.1	116.6	86.6	91.9	79.5	83.1
ENG03	116.7	119.6	88.0	92.6	80.9	83.7
ENG04	109.2	110.4	82.3	85.2	75.4	77.3
ENG05	110.2	111.3	83.4	86.4	75.4	78.4
ENG24	125.7	126.8	93.6	98.0	85.7	88.6
ENG30	114.9	116.8	86.5	90.6	79.5	83.4
ENG37	112.5	114.0	84.1	88.1	77.8	80.9

Table 2: Test set perplexity by topic and training set size.

For each topic and training set size, we train a bigram language model on the training set, use the development data to tune the regularization weights γ_0 and γ_1 (we fixed $\gamma_2 = 0$), and evaluate on the test set. We use as our baseline a bigram "standard" ℓ_1 regularized maximum entropy (ME) language model, trained as an SLR-LM with γ_0 chosen such that the low rank matrix is zero.

3.2. Results and Discussion

The results are presented in Table 2. As expected, in all cases the optimal SLR-LM has a lower perplexity than the baseline

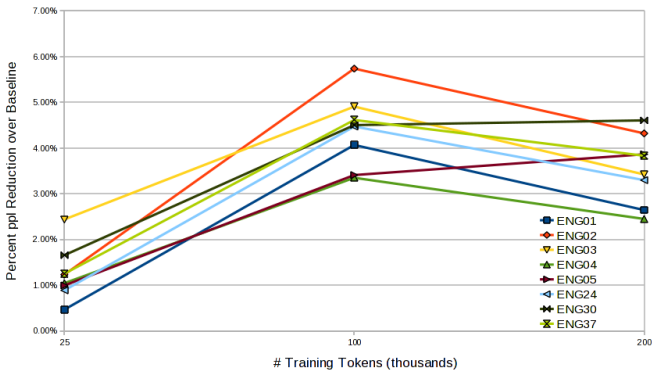

Figure 2: Percent perplexity reduction over baseline by topic and training set size.

History Word h	**Nearest Neighbors**
would	could, didn't, can't, don't, should
but	well, mean, because, guess, think
of	from, for, at, in, make
Prediction Word x	**Nearest Neighbors**
would	can, did, don't, didn't, should, could
but	so, because, now, for, as
know	keep, think, want, get, talk
one	two, four, three, ten, five
dog	cat, thing, ones, animal, baby

Table 3: Words, histories, and their nearest neighbors in the continuous-space embeddings induced by the low rank component from the 200K ENG02 set.

ℓ_1 regularized model. Fig. 2, which plots the percent perplexity reduction over baseline by topic and training set size, makes it easier to see the overall trends. In particular, although gains are observed at all configurations, the biggest gains are achieved on the 100K data set size. Presumably, when there is very little training data (25K), it is difficult to learn the patterns in the data simply because too few instances are observed. In other words, we observe a rough skeleton of the true matrix, with too many holes to be accurately filled by the low rank component. On the other hand, as the amount of training data grows, there become enough examples that the patterns can be captured by the aggregation of "exceptions"; that is, the patterns in the matrix become dense enough that there are few holes left for the low rank component to fill. Again, the SLR-LM still outperforms the standard ME-LM in these cases, just by a smaller margin.

We looked at the high and low weight entries (n-grams) learned for the sparse and low rank components for a model trained on the topic "minimum wage." The sparse component, which models the exceptions in the data, typically learned common noun phrases, including locations ("new york", "united states") and topic-related phrases ("social security", "grocery store"). Although nothing prevents the sparse component from having large negative weight entries to revise the probabilities of n-grams downward, empirically we find this behavior very rare. The low rank component assigned high weights to n-grams that are syntactically or semantically plausible ("you know," "I think," "I don't"), and low weights to ones that are not ("the a," "my that").

If the SLR-LM is indeed learning a low-dimensional continuous representation, words (or histories) that function similarly should be mapped close to each other in the continuous representation. Using a model trained with 200K tokens on the topic of "pets," we list in Table 3 several words and histories with their nearest neighbors in the low dimension space. Note that sometimes a word's neighbors are similar in either position (e.g. "would"), but they need not be (e.g. "but"). As expected, natural clusters form, e.g. numbers.

4. Conclusions

In this paper we introduce a new sparse plus low rank maximum entropy language model that generalizes a large class of ℓ_1 regularized models, and an efficient algorithm to train it. The SLR-LM automatically performs natural and flexible "soft-tying" of parameters between similar words (and histories) that improves generalization, and can be viewed as a continuous language model discriminatively trained in the maximum entropy framework. In bigram language modeling experiments on conversational speech, with varying topic and training set sizes, we observe consistent 2-5% reductions in perplexity.

To facilitate comparison, we focused on basic n-gram features. However, the SLR-LM supports arbitrary feature functions on words and on histories. Through feature functions one can naturally model higher order n-grams (e.g. by letting $\phi(h)$ map to a one-hot encoding of (n-1)-grams). Richer feature sets (e.g. morphological features, part of speech tags, semantic features, etc.) would provide more traction for the low rank component to flexibly pool information between words and histories. Thus, we anticipate that better feature sets will lead to even larger performance gains over the baseline. Motivated by previous results [1, 2], we also expect that interpolating the SLR-LM with a standard smoothed n-gram model would yield further improvements.

5. Acknowledgements

Research funded in part by NSF CAREER grant ECCS-0847077 and the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA). All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of the NSF, IARPA, the ODNI or the U.S. Government.

6. References

[1] B. Hutchinson, M. Ostendorf, and M. Fazel, "Low rank language models for small training sets," *IEEE Signal Processing Letters*, vol. 18, no. 9, pp. 489–492, 2011.

[2] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," *Computer Speech and Language*, vol. 10, pp. 187–228, 1996.

[3] S. F. Chen, "Performance prediction for exponential language models," in *Proc. NAACL*, 2009.

[4] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proc. ICML*, 2005, pp. 713–719.

[5] Y. Amit, M. Fink, N. Srebro, and S. Ullman, "Uncovering shared structures in multiclass classification," in *Proc. ICML*, 2007, pp. 17–24.

[6] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A nueral probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[7] H. Schwenk, "Continuous space language models," *Comput. Speech Lang.*, vol. 21, no. 3, pp. 492–518, Jul. 2007.

[8] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proc. ICML*, 2007, pp. 641–648.

[9] K. Min, Z. Zhang, J. Wright, and Y. Ma, "Decomposing background topics from keywords by principal component pursuit," in *ACM CIKM*, October 2010.

[10] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, March 2002.

[11] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[12] K. C. Toh and S. W. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pacific Journal of Optimization*, November 2009.

[13] R. M. Larsen, "Propack: a software for large and sparse SVD calculations." [Online]. Available: http://sun.stanford.edu/rmunk/PROPACK/

[14] C. C. David, D. Miller, and K. Walker, "The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text," in *Proc. International Conference on Language Resources and Evaluation*, 2004, pp. 69–71.