Sparse Forward-Backward for Fast Training of Conditional Random Fields

Charles Sutton, Chris Pal and Andrew McCallum
University of Massachusetts Amherst
Dept. Computer Science
Amherst, MA 01003
{casutton, pal, mccallum}@cs.umass.edu

Abstract

Complex tasks in speech and language processing often include random variables with large state spaces, both in speech tasks that involve predicting words and phonemes, and in joint processing of pipelined systems, in which the state space can be the labeling of an entire sequence. In large state spaces, however, discriminative training can be expensive, because it often requires many calls to forward-backward. Beam search is a standard heuristic for controlling complexity during Viterbi decoding, but during forward-backward, standard beam heuristics can be dangerous, as they can make training unstable. We introduce *sparse forward-backward*, a variational perspective on beam methods that uses an approximating mixture of Kronecker delta functions. This motivates a novel *minimum-divergence* beam criterion based on minimizing KL divergence between the respective marginal distributions. Our beam selection approach is not only more efficient for Viterbi decoding, but also more stable within sparse forward-backward training. For a standard text-to-speech problem, we reduce CRF training time fourfold—from over a day to six hours—with no loss in accuracy.

1 Introduction

Complex tasks in speech and language processing often include random variables with large state spaces. Training such models can be expensive, even for linear chains, because standard estimation techniques, such as expectation maximization and conditional maximum likelihood, often require repeatedly running foward-backward over the training set, which requires quadratic time in the number of states. During Viterbi decoding, a standard technique to address this problem is *beam search*, that is, ignoring variable configurations whose estimated max-marginal is sufficiently low. For sum-product inference methods such as forward-backward, beam methods can be dangerous, however, because standard beam selection criteria can inappropriately discard probability mass in a way that makes training unstable.

In this paper, we introduce a perspective on beam search that motivates its use within sum-product inference. In particular, we cast beam search as a variational procedure that approximates a distribution with a large state space by a mixture of many fewer Kronecker delta functions. This motivates *sparse forward-backward*, a novel message-passing algorithm in which after each message pass, approximate marginal potentials are compressed after each pass. Essentially, this extends beam search from max-product inference to sum-product. Our perspective also motivates the *minimum-divergence beam*, a new beam criterion that selects a compressed marginal distribution with a fixed Kullback-Leibler (KL) divergence of the true marginal. Not only does this criterion perform better than standard beam criteria for Viterbi decoding, it iteracts more stably with training. On one real-world task, the NetTalk text-to-speech data set [5], we can now train a conditional random field (CRF) in about 6 hours for which training previously required over a day, with no loss in accuracy.

2 Sparse Forward-Backward

Standard beam search can be viewed as maintaining sparse *local marginal* distributions such that together they are as close as possible to a large distribution. In this section, we formalize this intuition using a variational argument, which motivates our new beam criterion for sparse forward-backward.

Consider a discrete distribution $p(y)$, where y is assumed to have very many possible configurations. We approximate p by a sparse distribution q, which we write as a mixture of Kronecker delta functions:

$$q(y) = \sum_{i \in I} q_i \delta_i(y), \tag{1}$$

where $I = \{i_1, \ldots, i_k\}$ is the set of indices i such that $q(y = i)$ is non-zero, and $\delta_i(y) = 1$ if $y = i$. We refer to the set I as *the beam*.

Consider the problem of finding the distribution $q(y)$ of smallest weight such that $\mathrm{KL}(q\|p) \leq \epsilon$. First, suppose the set $I = \{i_1, \ldots, i_k\}$ is fixed in advance, and we wish to choose the probabilities q_i to minimize $\mathrm{KL}(q\|p)$. Then the optimal choice is simply $q_i = p_i / \sum_{i \in I} p_i$, a result which can be verified using Lagrange multipliers on the normalization constraint of q.

Second, suppose we wish to determine the set of indices I of a fixed size k which minimize $\mathrm{KL}(q\|p)$. Then the optimal choice is when $I = \{i_1, \ldots, i_k\}$ consists of the indices of the largest k values of the discrete distribution p. First, define $Z(I) = \sum_{i \in I} p_i$, then the optimal approximating distribution is:

$$\arg\min_q \mathrm{KL}(q\|p) = \arg\min_I \left\{ \arg\min_{\{q_i\}} \sum_{i \in I} q_i \log \frac{q_i}{p_i} \right\} \tag{2}$$

$$= \arg\min_I \left\{ \sum_{i \in I} \frac{p_i}{Z(I)} \log \frac{p_i/Z(I)}{p_i} \right\} \tag{3}$$

$$= \arg\max_I \left\{ \log Z(I) \right\} \tag{4}$$

That is, the optimal choice of indices is the one that retains most probability mass. This means that it is straightforward to find the discrete distribution q of minimal weight such that $\mathrm{KL}(q||p) \leq \epsilon$. We can sort the elements of the probability vector p, truncate after $\log Z(I)$ exceeds $-\epsilon$, and renormalize to obtain q.

To apply these ideas to forward-backward, essentially we compress the marginal beliefs after every message pass. We call this method *sparse forward-backward*, which we define as follows. Consider a linear-chain probability distribution $p(\mathbf{y}, \mathbf{x}) \propto \prod_t \Psi_t(y_t, y_{t-1}, \mathbf{x})$, such as an hidden Markov model (HMM) or conditional random field (CRF). Let $\alpha_t(i)$ denote the forward messages, $\beta_t(i)$ the backward messages, and $\gamma_t(i) = \alpha_t(i)\beta_t(i)$ be the computed marginals. Then the sparse forward recursion is:

1. Pass the message in the standard way:

$$\alpha_t(j) \leftarrow \sum_i \Psi_t(i, j, \mathbf{x})\alpha_{t-1}(i) \tag{5}$$

2. Compute the new dense belief γ_t as
$$\gamma_t(j) \propto \alpha_t(j)\beta_t(j) \tag{6}$$
3. Compress into a sparse belief $\gamma'(j)$, maintaining $\mathrm{KL}(\gamma' \| \gamma) \leq \epsilon$. That is, sort the elements of γ and truncate after $\log Z(I)$ exceeds $-\epsilon$. Call the resulting beam I_t.
4. Compress $\alpha_t(j)$ to respect the new beam I_t.

The backward recursion is defined similarly. Note that in every compression operation, the beam I_t is recomputed from scratch; therefore, during the backward pass, variable configurations can both leave and enter the beam on the basis of backward information. Just as in standard forward-backward, it can be shown by recursion that the sum of final alphas yields the mass of the beam. That is, if I is the set of all state sequences in the beam, then $\sum_j a_T(j) = \sum_{\mathbf{y} \in I} \prod_t \Psi_t(y_t, y_{t-1}, \mathbf{x})$. Therefore, because backward revisions to the beam do not decrease the local sum of betas, they do not damage the quality of the global beam over sequences.

The criterion in step 3 for selecting the beam is novel, and we call it the *miniumum-divergence* criterion. Alternatively, we could take the top N states, or all states within a threshold. In the next section we will compare to these alternate criteria.

Finally, we discuss a few practical considerations. We have found improved results by adding a minimum belief size constraint K, which prevents a belief state $\gamma'_t(j)$ from being compressed below K non-zero entries. Also, we have found that the minimum-divergence criterion usually finds a good beam after a single forward pass. Minimizing the number of passes is desirable, because if finding a good beam requires many forward and backward passes, one may as well do exact forward-backward.

3 Results and Analysis

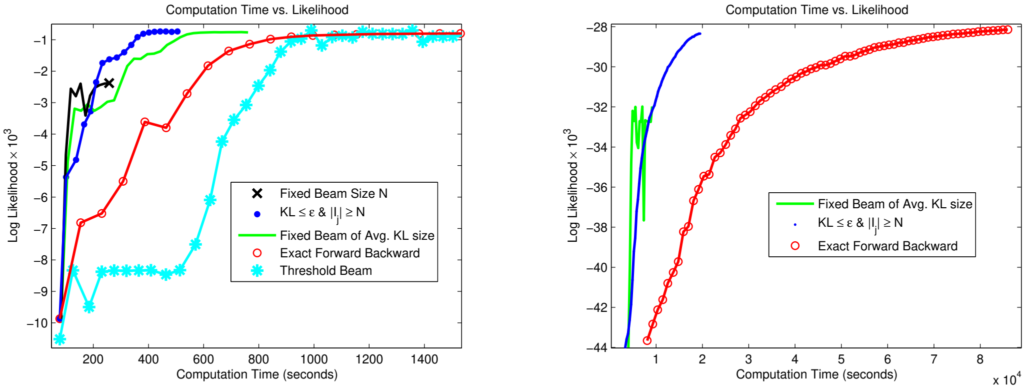
In this section we evaluate sparse forward-backward for both max-product and sum-product inference in HMMs and CRFs and the well known NetTalk text-to-speech dataset [5] which contains 20,008 English words. The task is to produce the proper phones given a string of letters as input.

3.1 Decoding Experiments

In this section we compare the our minimum-divergence criterion to traditional beam search criteria during Viterbi decoding. We generate synthetic data from an HMM of length 75. Transition matrix entries are sampled from a Dirichlet with every $\alpha_j = .1$. Emission matrices are generated from a mixture of two distributions: (a) a low entropy, sparse conditional distribution with 10 non-zero elements and (b) a high entropy Dirichlet with every $\alpha_j = 10^4$, with mixture weights of .75 and .25 respectively. The goal is to simulate a regime where most states are highly informative about their destination, but a few are less informative. We compared three beam criteria: (1) a fixed beam size, (2) an adaptive beam where message entries are retained if their log score is within a fixed threshold of the best so far, and (3) our minimum-divergence criterion with $KL \leq 0.001$ and an additional minimum beam size constraint of $K \geq 4$. Our minimum-divergence criterion finds the exact Viterbi path an average only 9.6 states per variable. On the other hand, the fixed beam requires between 20 and 25 states to reads the same accuracy, and the simple threshold beam requires 30.4 states per variable. We have similar results on the NetTalk data (omitted due to space).

3.2 Training Experiments

In this section, we present results showing that sparse forward-backward can be embedded within CRF training, yielding significant speedups in training time with no loss in testing performance.



(a) (b)

Figure 1: Comparison of sparse forward-backward methods for CRF training on both synthetic data (left) and on the NetTalk data set (right). Both graphs plot log likelihood on the training data as a function of training time. In both cases, sparse forward-backward performs equivalently to exact training on both training and test accuracy using only a quarter of the training time.

First, we train CRFs using synthetic data generated from a 100 state HMM in the same manner as in the previous section. We use 50 sequences for training and 50 sequences for testing. In all cases we use exact Viterbi decoding to compute testing accuracy. We compare five different methods for discarding probability mass: (1) the minimum-divergence beam with $KL \leq 0.5$ and minimum beam size $K \geq 30$ (2) a fixed beam of size $K = 30$, (3) a fixed beam whose size was the average size used by the minimum-divergence beam, (4) a threshold based beam which explores on average the same number of states as the minimum-divergence beam, and (5) exact forward backward. Learning curves are shown in Figure 1(a).

Compared to exact training, sparse forward-backward uses one-fourth of the time of exact training with no loss in accuracy. Also, we find it is important for the beam to be adaptive, by comparing to the fixed beam whose size is the average number of states used by our minimum-divergence criterion. Although minimum divergence and the fixed beam converge to the same solution, minimum divergence finishes faster, indicating that the adaptive beam does help training time. Most of the benefit occurs later in training, as the model becomes farther from uniform.

In the case of the smaller, fixed beam of size N, our L-BFGS optimizer terminated with an error as a result of the noisy gradient computation. In the case of the threshold beam, the likelihood gradients were erratic, but L-BFGS did terminate normally. However the recognition accuracy of the final model was low, at 67.1%.

Finally, we present results from training on the real-world NetTalk data set. In Figure 1(b) we present run time, model likelihood and accuracy results for a 52 state CRF for the NetTalk problem that was optimized using 19075 examples and tested using 934 examples. For the minimum divergence beam, we set the divergence threshold $\epsilon = .005$ and the minimum beam size $K \geq 10$. We initialize the CRF parameters using a subset of 12% of the data, before training on the full data until convergence. We used the beam methods during the complete training run and during this initialization period.

During the complete training run, the threshold beam gradient estimates were so noisy that our L-BFGS optimizer was unable to take a complete step. Exact forward backward training produced a test set accuracy of 91.6%. Training using the larger fixed beam $(N = 20)$ terminated normally but very noisy intermediate gradients were found in the terminating iteration. The result was a much lower accuracy of 85.7%. In contrast, the minimum divergence beam achieved an accuracy of 91.7% in less than 25% of the time it took to exactly train the CRF using forward-backward.

4 Related Work

Related to our work is zero-compression in junction trees [3], described in [2], which considers every potential in a clique tree, and sets the smallest potential values to zero, with the constraint that the total mass of the potential does not fall below a fixed value δ. In contrast to our work, they prune the model's potentials once before performing inference, whereas we dynamically prune the beliefs during inference, and indeed the beam can change during inference as new information arrives from other parts of the model. Also, Jordan et al. [4], in their work on hidden Markov decision trees, introduce a variational algorithm that uses a delta on a single best state sequence, but they provide no experimental evaluation of this technique. In computer vision, Coughlan and Ferreira [1] have used a belief pruning method within belief propagation for loopy models which is very similar to our threshold beam baseline.

5 Conclusions

We have presented a principled method for significantly speeding up decoding and learning tasks in HMMs and CRFs. We also have presented experimental work demonstrating the utility of our approach. As future work, we believe a promising avenue of exploration would be to explore adaptive strategies involving interaction of our L-BFGS optimizer, detecting excessively noisy gradients and automatically setting ϵ values. While results here were only with linear-chain models, we believe this approach should be more generally applicable. For example, in pipelines of NLP tasks, it is often better to pass lattices of predictions rather than single-best predictions, in order to preserve uncertainty between the tasks. For such systems, the current work has implications for how to select the lattice size, and how to pass information *backwards* through the pipeline, so that higher-level information from later tasks can improve performance on earlier tasks.

ACKNOWLEDGMENTS This work was supported in part by the Center for Intelligent Information Retrieval, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grants #IIS-0326249 and #IIS-0427594, and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

[1] James M. Coughlan and Sabino J. Ferreira. Finding deformable shapes using loopy belief propagation. In *European Conference on Computer Vision*, 2002.

[2] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

[3] F. Jensen and S. K. Andersen. Approximations in bayesian belief universes for knowledge-based systems. *Proceedings of the 6th Conference on Uncertainty in Artifcial Intelligence*, 1990. Appears to be unavailable.

[4] Michael I. Jordan, Zoubin Ghahramani, and Lawrence K. Saul. Hidden Markov decision trees. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Proc. Conf. Advances in Neural Information Processing Systems, NIPS*, volume 9. MIT Press, 1996.

[5] T.J. Sejnowski and C.R. Rosenberg. Nettalk: a parallel network that learns to read aloud. *Cognitive Science*, 14:179–211, 1990.