In 2006 International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

SPARSE FORWARD-BACKWARD USING MINIMUM DIVERGENCE BEAMS FOR FAST TRAINING OF CONDITIONAL RANDOM FIELDS

Chris Pal, Charles Sutton, and Andrew McCallum

University of Massachusetts Amherst
Dept. Computer Science
Amherst, MA 01003
{pal, casutton, mccallum}@cs.umass.edu

ABSTRACT

Hidden Markov models and linear-chain conditional random fields (CRFs) are applicable to many tasks in spoken language processing. In large state spaces, however, training can be expensive, because it often requires many iterations of 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

Model optimization for finite state transducers with large state spaces can be slow, because standard estimation techniques, such as expectation maximization and conditional maximum likelihood, often require repeatedly running foward-backward over the training set. This is especially problematic when the state space is large, because forward-backward 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. Beam search is essential to practical recognition systems [1, 2]. For sum-product inference methods such as forward-backward, folk wisdom exists in the community that beam methods can be effective. However, they can also be dangerous, because standard beam selection criteria can inappropriately discard probability mass in a way that makes optimization unstable. Perhaps for this reason they have received little attention in the literature.

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 approximate marginal distributions are compressed after each message 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 within 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 model optimization.

The contributions of this paper are: (1) proposing *sparse forward-backward* as a fast method for computing marginals during training of hidden Markov models (HMMs) and conditional random fields (CRFs), (2) proposing the minimum-divergence criterion for selecting the beam, (3) experimental comparison of minimum divergence to other criteria for Viterbi beam search, and (4) experimental comparison of minimum divergence to other criteria for CRF training on the well-known NetTalk text-to-speech data [3].

2. BACKGROUND AND NOTATION

In this section, we present our notation for hidden Markov models (HMMs) and conditional random fields (CRFs). We also briefly review current techniques for CRF training.

HMMs are a classical type of directed graphical model for sequence data. Define an observation sequence of discrete random variables as $\mathbf{x} = (x_1, \ldots, x_T)$ and a sequence of discrete random variables for the state (label) variables as $\mathbf{y} = (y_1, \ldots, y_T)$. Then an HMM models the sequence probability as

$$p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^{T} p(x_t|y_t)p(y_t|y_{t-1}), \qquad (1)$$

where for simplicity we define $p(y_1|y_0) = p(y_1)$. During inference and parameter estimation, we are often interested in computing marginal distributions $p(y_t|\mathbf{x})$ for all time steps t. During decoding, we are interested in efficiently computing the most probable state sequence $\mathbf{y}$, that is, $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$.

A conditional random field (CRF) [4] models the conditional distribution $p(\mathbf{y}|\mathbf{x})$ directly. A first-order, linear-chain CRF is defined as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_t \Psi_t(y_t, y_{t+1}, \mathbf{x}), \tag{2}$$

where $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_t \Psi_t(y_t, y_{t+1}, \mathbf{x})$ is a normalizing factor over all output configurations. A CRF is parameterized using feature functions $\{f_k\}$ such that

$$\Psi_t(y_t, y_{t+1}, \mathbf{x}) = \exp\left(\sum_k \lambda_k f_k(y_t, y_{t+1}, \mathbf{x}) \right), \tag{3}$$

where λ_k are the parameters or feature weights for the model.

Training of a CRF is typically done by maximizing the conditional log-likelihood of fully-observed training data $\mathcal{D} = \{\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i\}_{i=1}^N$. If $F_t(\mathbf{y}, \mathbf{x}) = \{f_k(y_t, y_{t+1}, \mathbf{x})\}$ denotes the vector of feature values at time t, and $\mathbf{F}(\mathbf{y}, \mathbf{x}) = \sum_t F_t(\mathbf{y}, \mathbf{x})$ denotes the *global feature function*, then the gradient of the conditional log likelihood $\mathcal{L} = \sum_i \log p(\tilde{\mathbf{y}}_i|\tilde{\mathbf{x}}_i, \boldsymbol{\lambda})$ with respect to the model parameters $\boldsymbol{\lambda} = \{\lambda_k\}$ is given by

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L} = \sum_i \left(\mathbf{F}(\tilde{\mathbf{y}}_i, \tilde{\mathbf{x}}_i) - E_p\big\langle \mathbf{F}(\mathbf{y}_i, \tilde{\mathbf{x}}_i) \big\rangle \right), \tag{4}$$

where $E_p\langle\cdot\rangle$ denotes the expectation under the distribution $p = p(\mathbf{y}_i|\tilde{\mathbf{x}}_i, \boldsymbol{\lambda})$. It is important to observe that this requires performing inference once for each sequence, per iteration of the optimizer. For data sets with large state spaces, this procedure can require *days* of computation. Following previous work [5], we optimize the parameters using limited-memory BFGS (L-BFGS) [6], a limited-memory variant of a standard quasi-Newton gradient-based optimizer. This has implications for inference algorithms that compute $E_p\langle\cdot\rangle$ approximately, because inaccurate gradients will degrade the BFGS approximation to the Hessian, severely hurting convergence of the optimizer.

3. 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:

$$q(y) = \sum_{i \in I} q_i \delta_i(y), \tag{5}$$

where $I = \{i_1, \dots, i_k\}$ 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* and its cardinality $|I|$ as the *weight* of 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 $I = \{i_1, \dots, i_k\}$ 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 $I = \{i_1, \dots, i_k\}$ consists of the indices of the largest k values of the discrete distribution p. To see this, first define $Z(I) = \sum_{i \in I} p_i$. Then the optimal approximating distribution is:

$$\begin{aligned} \arg\min_q \mathrm{KL}(q\|p) &= \arg\min_I \left\{ \arg\min_{\{q_i\}} \sum_{i \in I} q_i \log \frac{q_i}{p_i} \right\} & (6) \\ &= \arg\min_I \left\{ \sum_{i \in I} \frac{p_i}{Z(I)} \log \frac{p_i / Z(I)}{p_i} \right\} & (7) \\ &= \arg\max_I \big\{ \log Z(I) \big\} & (8) \end{aligned}$$

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 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 in sequence models, essentially we compress the marginal beliefs after every message pass. We call this method *sparse forward-backward*, which we define as follows. 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. We initialize $\beta_t(j) = 1$ for all time steps t and states i and j. Then the sparse forward recursion is:

1. Pass the message in the standard way:

$$\alpha_t(j) \leftarrow \sum_i \Psi_t(i, j) \alpha_{t-1}(i) \tag{9}$$

2. Compute the new dense belief γ_t as

$$\gamma_t(j) \propto \alpha_t(j) \beta_t(j) \tag{10}$$

3. Compress into a sparse belief $\gamma'(j)$, maintaining $\mathrm{KL}(\gamma'\|\gamma) \leq \epsilon$. 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 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 *minimum-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.

4. RESULTS AND ANALYSIS

In this section we evaluate sparse forward-backward for both max-product and sum-product inference in HMMs and CRFs, using both synthetic data and the well known NetTalk text-to-speech data set.

4.1. Decoding Experiments

While our primary focus is on sparse forward-backward during training, in this section we compare minimum divergence 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 $\alpha = .1$ and emission matrices are generated from a mixture of two distributions: (1) a low entropy, sparse conditional distribution with 10 non-zero elements and (2) a high entropy Dirichlet with $\alpha = 10^4$, with priors 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 compare our minimum-divergence criterion against two traditional beam search criteria: (1) a fixed beam size, and (2) an adaptive beam where message entries are retained if their log score is within a fixed threshold of the best so far. Minimum divergence using $KL \leq 0.001$ and minimum beam size $|I_i| \geq 4$ finds the exact Viterbi solution with an average of only 9.6 states per variable. On the other hand, the fixed beam requires between 20 and 25 states, and the simple threshold beam requires 30.4 states per variable to achieve the same accuracy. We have similar results on the NetTalk data.

4.2. 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.

First, we train CRFs on synthetic data generated from a 100 state HMM generated 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.

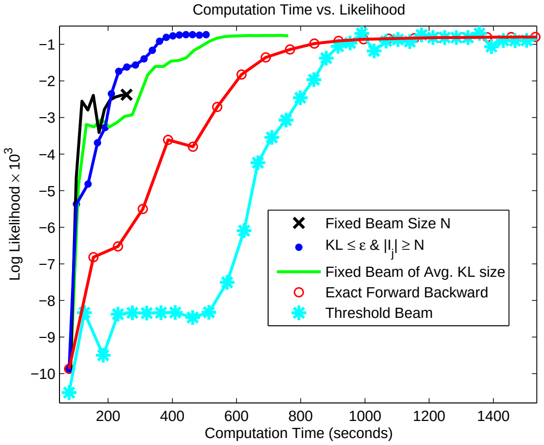

Fig. 1. Learning curves for CRF training on synthetic data. Sparse forward-backward has the same accuracy as exact training with less than a quarter of the training time. Other beam criteria are either slower or less robust than minimum divergence.

Figure 1 shows learning curves plotting log likelihood on the training set against computation time in seconds. We compare five different methods: (1) the minimum-divergence beam with $KL \leq 0.5$, $|I_i| \geq 30$, (2) a small fixed beam of $|I_i| = 30$, (3) a larger fixed beam, (4) a threshold beam, and (5) exact forward backward. Both the larger fixed beam and the threshold beam are calibrated to explore on average the same number of states as the minimum-divergence beam.

Compared to exact forward backward, the minimum divergence beam uses one-fourth of the time of exact training with no loss in accuracy. The larger fixed beam is designed to test how important it is for the beam to be adaptive, because this fixed beam uses the average number of states used by our minimum-divergence criterion. Although minimum divergence and the larger fixed beam converge to the same solution, minimum divergence finishes faster, indicating that the adaptive beam does improve training time. Most of the benefit occurs later in training, as the model becomes farther from uniform.

The small fixed beam performs poorly, because the noisy gradient computation causes our L-BFGS optimizer to terminate early. Finally, the threshold beam results in somewhat inaccurate gradients, but L-BFGS does terminate normally. However, the recognition accuracy of the final model is low, at 67.1%.

Finally, we present results training on the real-world NetTalk data set [3]. The task is to produce the proper phones given a string of letters as input. The data consists of 20,008 English words. In Figure 2 we present run time, model likelihood and accuracy results for a 52-state CRF for the NetTalk problem that is trained on 19075 examples and tested on 934 examples. In CRFs without latent variables, as here, choice of

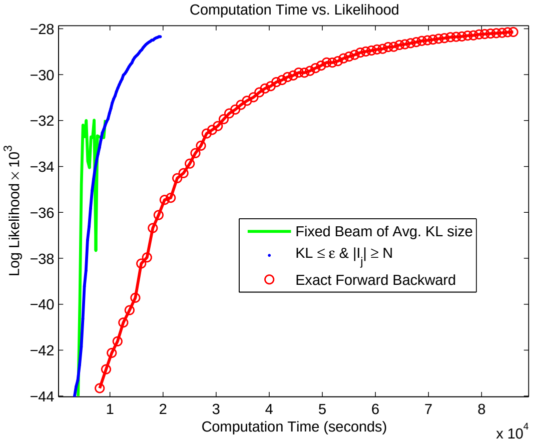

Fig. 2. Learning curves for CRF training on NetTalk. Sparse forward-backward (final test accuracy of 91.7%) performs equivalently to exact training (91.6%) using only a quarter of the training time. A fixed-size beam yields unstable results (85.7%).

initialization does not change the final solution, because the penalized likelihood for CRFs is strictly concave. But good initialization can still reduce the number of gradient steps required to find the optimum. Therefore, we initialize the CRF parameters using a subset of 12% of the data, before training on the full data until convergence. Beam methods are used both during this initialization period and during the complete training run. We compare the minimum divergence beam with $KL \leq .005$ and $|I_i| \geq 10$ to a fixed beam ($|I_i| = 20$), a threshold beam (set to average 36 states per time step), and exact forward backward. After the initialization period, the threshold beam has test set accuracy of 67%, while minimum divergence, the fixed-sized beam, and exact forward backward all have accuracy on the test set of 74%.

After the complete training run, exact forward-backward training results in a test set accuracy of 91.6%. The fixed beam terminates normally, but with very noisy gradients in the final iteration, resulting in a test accuracy of only 85.7%. The threshold beam results in gradient estimates that are so noisy that our L-BFGS optimizer is unable to take a single complete step. In contrast, minimum divergence achieves an accuracy of 91.7% in less than one-quarter of the time of exact forward-backward.

5. RELATED WORK

Although beam search is commonly used for Viterbi decoding [1], we are unaware of published descriptions of its use during forward-backward. In the probabilistic graphical models community, there is related work on zero-compression in clique trees [7], described in [8]. Their technique considers every factor in a clique tree, and sets the smallest factor values to zero, with the constraint that the total mass of the factor does not fall below a fixed value δ. In contrast to our work, they prune the model's factors once before performing inference, whereas we dynamically prune the beliefs during inference. Indeed, in our method the beam can change during inference as new information arrives from other parts of the model. There is also closely related work in sparse loopy belief propagation in computer vision [9], but this does not use the minimum-divergence beam.

6. 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 illustrating 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 the results we have presented here are with HMMs and linear-chain CRFs, we believe this line of work can be generalized to other structures.

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.

7. REFERENCES

[1] X. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing: A Guide to Theory Algorithms and System Development*, chapter 12, Prentice Hall, New Jersey, 2001.

[2] Mosur K. Ravishankar, *Effcient Algorithms for Speech Recognition*, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1996.

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

[4] John Lafferty, Andrew McCallum, and Fernando Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proc. 18th International Conf. on Machine Learning*, 2001, pp. 282–289.

[5] Fei Sha and Fernando Pereira, “Shallow parsing with conditional random fields,” in *Proceedings of Human Language Technology-NAACL 2003*, Edmonton, Canada, 2003.

[6] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, 1999.

[7] 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.

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

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