TRAINING ALGORITHMS FOR HIDDEN CONDITIONAL RANDOM FIELDS

Milind Mahajan, Asela Gunawardana and Alex Acero

Microsoft Research
One Microsoft Way
Redmond, WA 98052
USA

{milindm,aselag,alexac}@microsoft.com

ABSTRACT

We investigate algorithms for training hidden conditional random fields (HCRFs) – a class of direct models with hidden state sequences. We compare stochastic gradient ascent with the RProp algorithm, and investigate stochastic versions of RProp. We propose a new scheme for model flattening, and compare it to the state of the art. Finally we give experimental results on the TIMIT phone classification task showing how these training options interact, comparing HCRFs to HMMs trained using extended Baum-Welch as well as stochastic gradient methods.

1. INTRODUCTION

Hidden conditional random fields (HCRFs) [1, 2] are a class of discriminative models that generalize both hidden Markov models (HMMs) and conditional random fields [3]. As such, they are ideally suited for speech recognition and classification problems, as they allow the use of hidden state sequences as with HMMs and arbitrary dependencies upon the acoustics as with CRFs. Unlike HMMs, HCRFs are capable of modeling long range acoustic dependencies, and do not require that acoustic processing be uniform across states.

HCRFs are direct models, giving the conditional probability of state sequences given the observed acoustics. The model has a simple exponential ("maximum entropy") form, with the conditional probability of the state sequence being modeled through a feature vector that is a function of the state sequence as well as the acoustic observation sequence. The dependence of the feature vector on the entire observation sequence is arbitrary, but the dependence on the state sequence is constrained to ensure that the state sequence is Markov given the observation sequence. This is the key difference between HCRFs and HMMs – HCRFs model the state sequence as being *conditionally* Markov given the observation sequence, while HMMs model the state sequence as being Markov, and each observation being independent of all others given the corresponding state. Since the model is never used to evaluate the probability of observations, the second assumption is unnecessary.

Since HCRFs are conditional models, they can only be trained using discriminative criteria such as conditional maximum likelihood. In [1], we showed that HCRFs can be trained by direct optimization of the conditional log likelihood using gradient methods. HCRFs trained using stochastic gradient ascent [4] outperformed HMMs trained using the extended Baum Welch (EBW) algorithm on the TIMIT phone classification task, and improved the state of the art. Prior work on estimating HMMs under the maximum mutual information (MMI) criterion [5, 6] and the minimum classification error (MCE) criterion [7] clearly show that the algorithm used in estimation has a strong influence on performance, even when the training criterion and model family are fixed. In this paper, we perform a similar set of experiments to determine how best to train HCRFs. Although we only present results for phone classification, there is no inherent limitation in the HCRF framework that precludes the use of HCRFs for recognition.

The paper is organized as follows. In Section 2 we briefly review the details of HCRFs and their relationship to HMMs. In Section 3 we explore different options for training HCRFs. Section 3.1 reviews gradient ascent and RProp [8], which is a gradient based algorithm that uses an adaptive step size. Section 3.2 discusses the possibility of making stochastic parameter updates based on random samples of the training data rather than using the entire training set. In Section 3.3, we review the need for model flattening during training, and introduce a novel flattening technique for HCRFs. Section 4 presents results on the TIMIT phone classification task that illustrate how best these techniques can be combined in HCRF estimation. Finally, we discuss our findings and conclude in Section 5.

2. THE MODEL

In [1], we introduce the HCRF as an exponential model that gives the conditional probability of a segment (phone) label w given the observation sequence $\mathbf{o} = (o_1, \cdots, o_T)$ through

$$p(w|\mathbf{o};\, \lambda) = \frac{1}{z(\mathbf{o};\lambda)} \sum_{\mathbf{s}\in w} \exp\left\{\lambda \cdot f(w, \mathbf{s}, \mathbf{o})\right\}. \qquad (1)$$

The *partition function* $z(\mathbf{o};\lambda)$ ensures that the model is a properly normalized probability over w, and is given by

$$z(\mathbf{o};\lambda) = \sum_{w,\mathbf{s}\in w} \exp\left\{\lambda \cdot f(w, \mathbf{s}, \mathbf{o})\right\}.$$

λ is the *parameter vector* and $f(w, \mathbf{s}, \mathbf{o})$ is a *feature vector*[1] of arbitrary functions of w, $\mathbf{s}$, and $\mathbf{o}$. As discussed in [1], the conditional probability induced by an HMM can be written in the exponential

[1] Note that in this context, the term feature vector refers to the vector of sufficient statistics used by the model, and not to the output of the acoustic front-end. The latter will be referred to as an *observation vector*.

form above, with the feature vector f having components

$$\begin{aligned}
f_{w'}^{(LM)}(w,\mathbf{s},\mathbf{o}) &= \delta(w=w') && \forall w' \\
f_{ss'}^{(Tr)}(w,\mathbf{s},\mathbf{o}) &= \sum_{t=1}^{T}\delta(s_{t-1}=s)\delta(s_t=s') && \forall s,s' \\
f_{s}^{(Occ)}(w,\mathbf{s},\mathbf{o}) &= \sum_{t=1}^{T}\delta(s_t=s) && \forall s \\
f_{s}^{(M1)}(w,\mathbf{s},\mathbf{o}) &= \sum_{t=1}^{T}\delta(s_t=s)o_t && \forall s \\
f_{s}^{(M2)}(w,\mathbf{s},\mathbf{o}) &= \sum_{t=1}^{T}\delta(s_t=s)o_t^2 && \forall s,
\end{aligned} \tag{2}$$

where $\delta(s=s')$ is equal to one when $s=s'$ and zero otherwise. Each (unigram) language model feature $f_w^{(LM)}$ triggers on the occurrence of the label w. The transition features $f_{ss'}^{(Tr)}$ count the number of times the transition ss' occurs in the state sequence $\mathbf{s}$, while the occupancy features $f_s^{(Occ)}$ count the occurrences of the state s. The first and second moments $f_s^{(M1)}$ and $f_s^{(M2)}$ are the sum and sum of squares of observations that align with the state s. In this paper, we will only address HCRFs that use the same choice of feature vector. As described in [1] this choice of feature vector ensures that the state sequence is Markov given the observation sequence. This allows the use of efficient forward and backward recursions to compute the statistics needed during training and decoding.

Although we describe HCRFs with the same feature vector as HMMs, not all such HCRFs yield posterior probability distributions achievable by HMMs. This is because HMM posterior distributions are HCRFs with constrained parameter vectors. These restrictions ensure that the HMM transition probabilities are normalized and that the variances are positive (or that the covariance matrix is positive definite). Specifically, an HMM with transition probabilities $a_{ss'}$, emission means μ_s, emission covariance σ_s^2 and unigram probability u_w can be viewed as an HCRF with parameter vector λ with components

$$\begin{aligned}
\lambda_{w'}^{(LM)} &= \log u_{w'} && \forall w' \\
\lambda_{ss'}^{(Tr)} &= \log a_{ss'} && \forall s,s' \\
\lambda_{s}^{(Occ)} &= -\tfrac{1}{2}\left(\log 2\pi\sigma_s^2 + \frac{\mu_s^2}{\sigma_s^2}\right) && \forall s \\
\lambda_{s}^{(M1)} &= \frac{\mu_s}{\sigma_s^2} && \forall s \\
\lambda_{s}^{(M2)} &= -\frac{1}{2\sigma_s^2} && \forall s.
\end{aligned}$$

Conversely, every HCRF with the feature vector of equation (2) can be viewed as an HMM with unnormalized transition probabilities (and mixture weights), and possibly negative variances. Although the "negative variances" preclude the use of these "HMMs" in generative mode, they still yield valid conditional distributions.

Note that for simplicity, we have only given expressions for using scalar observations and single Gaussian emission densities, although the arguments hold for vector valued observations and mixture densities. We treat the multiple mixture component case by interpreting $\mathbf{s}$ as a joint sequence of states and mixture components. In fact, all experiments were performed with the familiar vector valued observations and diagonal covariance Gaussian mixture emissions (and corresponding HCRFs).

3. ESTIMATING HMMS AND HCRFS

3.1. Training Algorithms

We examine several algorithms for optimizing an HCRF to maximize the conditional log likelihood

$$\mathcal{L}(\lambda) = \sum_{n=1}^{N}\log p(w^{(n)}|\mathbf{o}^{(n)};\,\lambda)$$

of the training data $(w^{(1)},o^{(1)}),\cdots,(w^{(N)},o^{(N)})$. In the case of HMMs, this is a well studied problem, with extended Baum-Welch (EBW) and its variants being popular solutions [6], as they ensure that the parameter constraints are maintained from iteration to iteration.

In the case of HCRFs, we will examine two gradient based techniques. The gradient of the conditional log-likelihood is given by

$$\begin{aligned}
\nabla\mathcal{L}(\lambda) = \sum_{n=1}^{N}\sum_{\mathbf{s}\in w^{(n)}} & f(w^{(n)},\mathbf{s},\mathbf{o}^{(n)})p(\mathbf{s}|w^{(n)},\mathbf{o}^{(n)};\,\lambda) \\
& - \sum_{w,\mathbf{s}\in w} f(w,\mathbf{s},\mathbf{o}^{(n)})p(w,\mathbf{s}|\mathbf{o}^{(n)};\,\lambda).
\end{aligned}$$

Substituting the vector of sufficient statistics f from equation (2) into the gradient, it can be shown that the first and second terms are the "numerator" and "denominator" counts used in EBW estimation of HMMs [6], which are easily computed.

If $\mathcal{L}(\lambda)$ were to be optimized by gradient ascent, the components of λ would be updated according to

$$\lambda_i^{(r+1)} = \lambda_i^{(r)} + \eta\nabla_i\mathcal{L}(\lambda^{(r)}).$$

where ∇_i represents the ith component of the gradient with respect to λ. The size of an update step is proportional to the gradient. On the other hand, RProp [8] is an algorithm that updates each component of λ using updates such as

$$\lambda_i^{(r+1)} = \lambda_i^{(r)} + \eta_i^{(r)}\,\mathrm{sgn}\left(\nabla_i\mathcal{L}(\lambda^{(r)})\right)$$

that use only the sign of the gradient. The step size $\eta_i^{(r)}$ is updated adaptively according to

$$\eta_i^{(r+1)} = \begin{cases} 0.5\eta_i^{(r)} & \text{if sgn}\left(\nabla_i\mathcal{L}(\lambda^{(r-1)})\nabla_i\mathcal{L}(\lambda^{(r)})\right) < 0, \\ 1.2\eta_i^{(r)} & \text{if sgn}\left(\nabla_i\mathcal{L}(\lambda^{(r-1)})\nabla_i\mathcal{L}(\lambda^{(r)})\right) > 0. \end{cases}$$

Thus, RProp accelerates through "flat" areas of the parameter space, and decelerates when a local optimum is passed. Note that the RProp algorithm [8] includes features such as backtracking which are not shown in the discussion above. RProp is promising for estimating HCRFs because the optimal step size can be very different for different components of λ in different areas of the parameter space.

While it is clear how to apply the gradient based algorithms above to HCRFs, it is less clear how to apply them to HMMs, where the parameters are constrained. While gradient based updates can be used for the means and the logs of the variances, it is unclear how to optimize transition probabilities and mixture weights. One option is to ignore the sum-to-one constraints and optimize the log probabilities using gradient updates [5]. However, this yields HCRFs rather than HMMs. The other is to re-normalize the transition probabilities and mixture weights after every gradient-based update [9]. However, the renormalized update is no longer guaranteed to increase the objective function.

3.2. Batch vs. Stochastic Updates

Since the training sets used in speech recognition are typically large, the complexity of estimation is typically dominated by the computation of the gradient of the log-likelihood over the training set. On the other hand, the gradient computed over a random subset (even a single example) can be viewed as an estimate of the true gradient, and used to update the parameter, allowing many updates to be made during each pass over the training set, possibly yielding faster convergence. In [1], we reported that HCRFs optimized using stochastic gradient ascent outperformed those optimized using an approximate second order gradient method known as L-BFGS [10]. We also reported that stochastic gradient ascent converged much faster than L-BFGS. In this paper, we examine whether RProp can also be used in stochastic mode.

3.3. Flattening

It is well known that obtaining good test set performance with EBW based estimation of HMMs requires the use of a flattening constant κ so that the criterion actually optimized is

$$\mathcal{L}(\lambda) = \sum_{n=1}^{N} \log \frac{p^{\kappa}(w^{(n)}|\mathbf{o}^{(n)};\,\lambda)}{\sum_w p^{\kappa}(w|\mathbf{o}^{(n)};\,\lambda)} = \sum_{n=1}^{N} \log \frac{\left[\sum_{\mathbf{s}\in w^{(n)}} e^{\lambda\cdot f(w^{(n)},\mathbf{s},\mathbf{o}^{(n)})}\right]^{\kappa}}{\sum_w \left[\sum_{\mathbf{s}\in w} e^{\lambda\cdot f(w,\mathbf{s},\mathbf{o}^{(n)})}\right]^{\kappa}}.$$

This is necessary because the initial models are often overly sharp, assigning too high a probability to the best hypothesis, which causes other hypotheses to have little influence during training.

In the case of HCRFs, it is natural to also consider flattening using

$$\mathcal{L}(\lambda) = \sum_{n=1}^{N} \log \frac{\sum_{\mathbf{s}\in w^{(n)}} e^{\kappa\lambda\cdot f(w^{(n)},\mathbf{s},\mathbf{o}^{(n)})}}{\sum_w \sum_{\mathbf{s}\in w} e^{\kappa\lambda\cdot f(w,\mathbf{s},\mathbf{o}^{(n)})}}.$$

While the former option flattens the posterior distribution over hypotheses (labels in the case of classification, aligned label sequences in the case of recognition), the second option flattens the posterior distribution over state and mixture component sequences. Hypothesis flattening has the property that it does not change the ranking of hypotheses by the model, while state sequence flattening may change the ranking of hypotheses. On the other hand, state sequence flattening of HCRFs has the advantage of being easily implemented by scaling the initial parameter vector.

We note that while hypothesis flattening has the effect of increasing the entropy of the posterior probability of hypotheses given the observation sequence, it does not increase the entropy of the posterior probability of state (and mixture component) sequences given a hypothesis and the observation sequence. In contrast, state sequence flattening increases the entropy of the posterior probability of state sequences given the observation. In other words, state sequence flattening can overcome the undesirable effect of locking of the state alignment caused by an overly sharp initial model. We note that the use of this form of flattening to control the entropy of the state sequences is in the same spirit as the flattening of classifier posterior distributions in deterministic annealing [11].

4. EXPERIMENTAL RESULTS

We compare the training procedures described above on the TIMIT phone classification task, using the experimental setup described in [1]. Results are reported on the MIT development test set [12] and the NIST core test set. The training, development, and evaluation sets have 142,910, 15,334, and 7333 phonetic segments respectively. We follow the standard practice of building models for 48 different phones, and then mapping down to 39 phones for scoring purposes [12]. We use a standard 39-dimensional Mel-Frequency Cepstral Coefficient (MFCC) front end with mean and variance normalization. We adjust segment boundaries given in the corpus to coincide with per-utterance segment boundaries, enabling easier processing with HTK [13]. This caused slight changes in performance compared to using the hand annotated boundaries. All systems were initialized from an ML trained HMM model with a three state left to right model and 20 diagonal Gaussians per state. We tested HCRF models with exactly the same topologies and feature vectors f. All parameters of the algorithms such as stopping point, step size, and flattening weight were tuned on the development set by performing ten iterations of training. The best settings discovered were then used on the evaluation set.

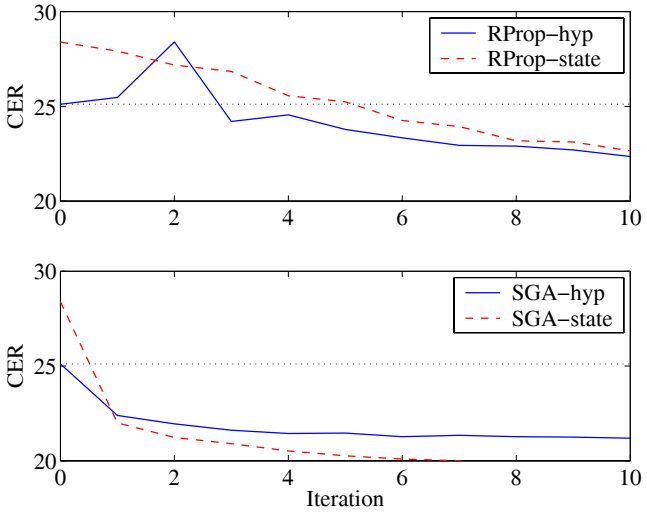

Fig. 1. The development test classification error rate (CER) of batch RProp and stochastic gradient ascent (SGA) using hypothesis flattening (hyp) and state sequence flattening (state), as a function of the number of passes through the training data. The horizontal line shows the performance of the ML trained HMM with which the HCRFs were initialized. The momentary degradation in hypothesis flattened RProp is due to an aggressive choice of initial step size and disappears with more conservative settings. However, this is the setting that gave the best eventual performance.

We first compared the performance of stochastic gradient ascent with (batch) RProp, using either state sequence or hypothesis flattening. As shown in Figure 1, state sequence flattening causes the initial performance to degrade, while hypothesis flattening does not, as was discussed in Section 3.3. Under stochastic gradient ascent, the performance of state sequence flattening quickly improves and beats the performance of hypothesis flattening, while (batch) RProp takes longer to recover from the initial degradation. This may be explained by the fact that stochastic gradient ascent makes multiple parameter updates per iteration through the training set.

The results in Figure 1 represent two extremes. Stochastic gradient ascent updates the parameter based on just one utterance at a time, while RProp uses the entire training set. As discussed in

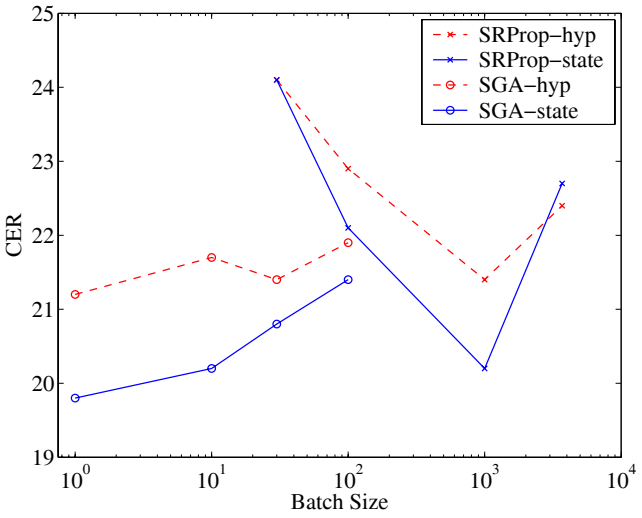

Fig. 2. The development test CER of stochastic RProp (SRProp) and stochastic gradient ascent (SGA) using hypothesis flattening (hyp) and state sequence flattening (state), as a function of batch size in number of utterances.

Model	EBW	SGA	SRProp
HMM	24.9%	23.2%	22.9%
HCRF		21.7%	21.3%

Table 1. Classification error on the evaluation test set for HMMs and HCRFs trained using stochastic gradient ascent (SGA), stochastic RProp (SRProp) and extended Baum-Welch (EBW). Training parameters such as step sizes, flattening weights, flattening schemes, and batch sizes were tuned on the development test set. HMM results are presented for hypothesis flattening only.

Section 3.2, both gradient ascent and RProp could be used to make stochastic updates to the parameters on the basis of batches consisting of any number of training examples at a time. In addition, Figure 1 indicates that the optimal flattening scheme may depend on the batch size used.

Figure 2 explores this dependency of error rate on training batch size and flattening scheme for stochastic gradient ascent and stochastic RProp. It can be seen that state sequence flattening gives better results at almost all batch sizes, that stochastic gradient ascent outperforms stochastic RProp, and that stochastic RProp outperforms batch RProp. The optimal batch size for stochastic RProp is much larger than for stochastic gradient ascent. This is probably due to the fact that RProp disregards the magnitude of the estimated gradient allowing small errors in estimating the gradient to potentially cause large steps in the wrong direction.

Finally, we compare the best performance obtained with HCRFs trained with stochastic gradient ascent and stochastic RProp with that obtained using HMMs trained with these algorithms, as well as with EBW. The results are shown in Table 1. For comparison, the best generative (ML) HMM CER is 25.81%. The gradient based HMM estimates followed the procedure outlined in [9], ignoring the variance terms in the gradients of the mean. We were unable to get an improvement with the rescaled transition and mixture weight updates. State sequence flattening was not attempted with HMMs.

5. CONCLUSIONS

The results indicate that stochastic RProp performs as well as stochastic gradient ascent and better than batch RProp for estimating HCRFs, giving state of the art performance. However, RProp may be preferable as it is more robust to the step size parameter. Our proposed state sequence flattening performs significantly better than hypothesis flattening after an initial degradation. The anecdotal evidence that hypothesis flattening performs better for HMM estimation with EBW may be due to this initial degradation. We conjecture that this may be overcome even in HMM estimation by either carrying out more frequent (stochastic) updates or by carrying out more iterations of EBW, as we have shown for HCRFs.

6. ACKNOWLEDGMENTS

The authors would like to thank Jasha Droppo for interesting and useful discussions and comments.

7. REFERENCES

[1] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Interspeech*, 2005.

[2] A. Quattroni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *NIPS*, 2004.

[3] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, pp. 282–289, 2001.

[4] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.

[5] V. Valtchev, *Discriminative Methods in HMM-based Speech Recognition*. PhD thesis, Cambride University, 1995.

[6] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, 2003.

[7] J. L. Roux and E. McDermott, "Optimization methods for discriminative training," in *Interspeech*, pp. 3341–3344, 2005.

[8] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *IEEE Int. Conf. Neur. Net.*, vol. 1, pp. 586–591, 1993.

[9] W. Chou, B. H. Huang, and C. H. Lee, "Segmental GPD training of HMM based speech recognizer," in *ICASSP*, pp. 473–476, 1992.

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

[11] K. Rose, E. Gurewitz, and G. C. Fox, "Statistical mechanics and phase transitions in clustering," *Phys. Rev. Lett.*, vol. 65, pp. 945–948, Aug. 1990.

[12] A. K. Halberstadt and J. R. Glass, "Heterogeneous acoustic measurements for phonetic classification," in *Eurospeech*, pp. 401–404, 1997.

[13] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book, Version 3.0*, July 2000.