A comparison of training approaches for discriminative segmental models

Hao Tang, Kevin Gimpel, Karen Livescu

Toyota Technological Institute at Chicago
{haotang,kgimpel,klivescu}@ttic.edu

Abstract

Segmental models such as segmental conditional random fields have had some recent success in lattice rescoring for speech recognition. They provide a flexible framework for incorporating a wide range of features across different levels of units, such as phones and words. However, such models have mainly been trained by maximizing conditional likelihood, which may not be the best proxy for the task loss of speech recognition. In addition, there has been little work on designing cost functions as surrogates for the word error rate. In this paper, we investigate various losses and introduce a new cost function for training segmental models. We compare lattice rescoring results for multiple tasks and also study the impact of several choices required when optimizing these losses.

Index Terms: speech recognition, segmental conditional random fields, empirical Bayes risk, large-margin training

1. Introduction

Automatic speech recognition (ASR) typically works by finding a word sequence that maximizes some goodness-of-fit function $f(\overline{x}, \overline{w})$ between the observed acoustics $\overline{x}$ and the word (or, more generally, label) sequence $\overline{w}$. This function is often an estimated joint probability over $(\overline{x}, \overline{w})$, and is factored into a language model and an acoustic model. The acoustic model, in turn, is also factored into chunks, each typically corresponding to a single frame of acoustic observations.

This frame-by-frame modeling has some well-known limitations, including the assumption of conditional independence between frames and the inability to use expressive features over larger segments of the speech signal (e.g., phonemes or words). There have been a number of efforts to develop *segmental* approaches that address these limitations [1, 2, 3]. A recent successful approach uses segmental conditional random fields (SCRFs) [4, 5], also referred to as semi-Markov CRFs [6]. SCRFs use a log-linear model for the conditional probability $p(\overline{w}|\overline{x})$, with feature functions defined on entire segments of observations. SCRFs provide a flexible way to incorporate segmental features, and have been used to reformulate some earlier approaches such as template-based ASR [7].

SCRF-based speech recognizers are trained by maximizing the conditional likelihood (CL), i.e. the probability of word sequences given acoustics in a training set. But there are several other loss functions that can be used, some of which better approximate the task loss (word error rate) of speech recognition. Many have previously been used for frame-based speech recognition [8]. For example, minimum Bayes risk [9, 10, 11], large-margin criteria [12, 13, 14], and hybrid criteria [15] have been successful for frame-based recognition. On the other end of the spectrum, our own previous work has shown some benefits of large-margin training over CL in isolated-word tasks with word-level feature functions [16]. However, there is little prior work on alternative training approaches for *segmental* models, and the work that exists is limited to a specific type [17, 18].

This paper attempts to fill this gap by studying several training criteria for segmental models. We consider the same setting as in recent work on SCRFs: We use a lattice rescoring framework and define all training criteria with respect to lattices. We perform these comparisons using a new toolkit that we developed.[1] This work is applicable to any discriminative log-linear model with segmental features. In this paper, for expediency we experiment with relatively "small" tasks: TIMIT phonetic recognition and a sign language recognition task.

In the remaining sections, we formalize our problem setting, define loss and cost functions, and present experimental results. We introduce a new cost function for use with cost-sensitive losses, experiment with losses not typically used in ASR, and compare optimization with Rprop (commonly used for SCRFs) vs. AdaGrad [19]. Ultimately, we find that log-loss is remarkably robust for segmental models, though other losses such as hinge and ramp loss often slightly outperform it; that optimization with AdaGrad provides a large speedup over Rprop; and that our new cost function improves over the one typically used in MPE/MWE [10].

2. Problem setting

Let $\mathcal{V}$ be a vocabulary of words (or more generally labels). We define a **segmentation** as a sequence of contiguous nonoverlapping closed intervals, or **segments**. The two end points of a segment $e = [s, t]$ are its start time s and end time t. Let $\mathcal{Q}$ be the set of all possible segments, $\mathcal{X}$ the set of all observation sequences (e.g., all MFCC sequences), and $\mathcal{Y}$ the set of **output structures**. Here an output structure is a pair containing a word sequence and a segmentation, i.e., $\mathcal{Y} = \mathcal{V}^* \times \mathcal{Q}^*$. For an observation sequence $\overline{x} \in \mathcal{X}$, a word sequence $\overline{w} \in \mathcal{V}^*$, and its segmentation $\overline{q} \in \mathcal{Q}^*$, we will write the word for segment $e = [s, t] \in \overline{q}$ as $\overline{w}_e$, and the corresponding acoustic vector subsequence $(x_s, \dots, x_t)$ as $\overline{x}_e$.

Given $\overline{x} \in \mathcal{X}$ and $\overline{y} = (\overline{w}, \overline{q}) \in \mathcal{Y}$, a **semi-Markov CRF**, or **SCRF**, defines the conditional distribution $p(\overline{y}|\overline{x})$ using a specific factorization:

$$p(\overline{y}|\overline{x}) = p(\overline{w}, \overline{q}|\overline{x}) = \frac{1}{Z(\overline{x})} \exp\left(\sum_{e \in \overline{q}} \boldsymbol{\theta}^\top \boldsymbol{\phi}(\overline{x}_e, \overline{w}_e, e)\right),$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ is the vector of model parameters (weights), $\boldsymbol{\phi} : \mathcal{X} \times \mathcal{V} \times \mathcal{Q} \to \mathbb{R}^n$ is a vector of feature functions, and

$$Z(\overline{x}) = \sum_{(\overline{w}', \overline{q}') \in \mathcal{Y}} \exp\left(\sum_{e \in \overline{q}'} \boldsymbol{\theta}^\top \boldsymbol{\phi}(\overline{x}_e, \overline{w}'_e, e)\right)$$

[1] Will be available at ttic.uchicago.edu/~haotang

is the partition function. SCRFs generalize standard linear-chain CRFs [20] by permitting several observations to share a single hidden state instead of just one. For simplicity, let $\boldsymbol{\phi}(\overline{x}, \overline{y}) = \boldsymbol{\phi}(\overline{x}, (\overline{w}, \overline{q})) = \sum_{e \in \overline{q}} \boldsymbol{\phi}(\overline{x}_e, \overline{w}_e, e)$. Inferring the output structure is done by finding the mode of the distribution:

$$(\hat{w}, \hat{q}) = \operatorname*{argmax}_{\overline{w}', \overline{q}'} p(\overline{w}', \overline{q}' | \overline{x}) = \operatorname*{argmax}_{\overline{y}'} \boldsymbol{\theta}^\top \boldsymbol{\phi}(\overline{x}, \overline{y}'). \quad (1)$$

Assume the observation sequences and output structures follow an unknown joint distribution ρ. The goal of the learning problem is to find a parameter vector that minimizes the expected risk, i.e., solving

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\overline{x}, \overline{y}) \sim \rho} [\ell(\boldsymbol{\theta}; \overline{x}, \overline{y})], \quad (2)$$

where ℓ is a task-specific loss function. However, we are only given a set of samples $S = \{(\overline{x}_1, \overline{y}_1), \dots, (\overline{x}_N, \overline{y}_N)\}$ drawn from ρ, and thus instead of solving Eq. (2), we turn to finding the parameters that minimize the empirical risk plus regularization terms:

$$\min_{\boldsymbol{\theta}} \lambda_1 \|\boldsymbol{\theta}\|_1 + \frac{\lambda_2}{2} \|\boldsymbol{\theta}\|_2^2 + \frac{1}{N} \sum_{i=1}^{N} \ell(\boldsymbol{\theta}; \overline{x}_i, \overline{y}_i), \quad (3)$$

where λ_1 and λ_2 are the regularization parameters.

The de facto task loss for word recognition is always

$$\ell(\boldsymbol{\theta}; \overline{x}, (\overline{w}, \overline{q})) = \text{dist}(\hat{w}, \overline{w}), \quad (4)$$

where $\hat{w}$ is the predicted word sequence given in Eq. (1) and dist is the Levenshtein distance. Although the task loss does not directly measure the quality of the segmentation, we do require the segmentation to be provided at training time. That is, the segmentation is not a hidden variable to be inferred during training. Using the task loss in Eq. (4), it is intractable to find the optimal solution for Eq. (3). What we can do is to use a surrogate loss $\hat{\ell}$ that approximates ℓ while being tractable to optimize. Several surrogate losses are defined in the next section.

3. Loss functions

We now describe several loss functions and provide their gradients, all with respect to a single sample $(\overline{x}, \overline{y})$. The standard loss function for training SCRFs is the log loss,

$$\ell_{\text{CL}}(\boldsymbol{\theta}; \overline{x}, \overline{y}) = -\log p(\overline{y} | \overline{x}). \quad (5)$$

Minimizing log loss is equivalent to maximizing conditional likelihood (CL) on the training set. Its gradient is

$$\nabla \ell_{\text{CL}}(\boldsymbol{\theta}; \overline{x}, \overline{y}) = -\boldsymbol{\phi}(\overline{x}, \overline{y}) + \mathbb{E}_{\overline{y}'} [\boldsymbol{\phi}(\overline{x}, \overline{y}')]. \quad (6)$$

Log loss does not explicitly contain the task loss (Levenshtein distance), which is our primary interest in learning. One way to target the task loss is to minimize the empirical Bayes risk (EBR), as in minimum phone error (MPE) or minimum word error (MWE) training [10]

$$\ell_{\text{EBR}}(\boldsymbol{\theta}; \overline{x}, \overline{y}) = \mathbb{E}_{\overline{y}'} \left[\text{cost}(\overline{y}', \overline{y}) \right], \quad (7)$$

where the cost function cost : $\mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ directly measures the quality of the prediction. Its gradient is

$$\begin{aligned} \nabla \ell_{\text{EBR}}(\boldsymbol{\theta}; \overline{x}, \overline{y}) = &-\mathbb{E}_{\overline{y}'} [\text{cost}(\overline{y}', \overline{y}) \boldsymbol{\phi}(\overline{x}, \overline{y}')] \\ &+ \mathbb{E}_{\overline{y}'} [\text{cost}(\overline{y}', \overline{y})] \mathbb{E}_{\overline{y}'} [\boldsymbol{\phi}(\overline{x}, \overline{y}')]. \quad (8) \end{aligned}$$

The gradient is the covariance between the features and the cost. The optimal model parameters are those that make the features and the cost uncorrelated. Although we can always write out the gradient of EBR analytically, it is only tractable if the cost function factors in a way that permits efficient computation. For segmental models, it is natural to require the cost to factor in the same way as the feature functions, i.e., over segments.

The second loss that relates directly to the task loss is the hinge loss [21, 17]

$$\begin{aligned} &\ell_{\text{hinge}}(\boldsymbol{\theta}; \overline{x}, \overline{y}) = \\ &\quad \max_{\overline{y}'} \left[\mu \, \text{cost}(\overline{y}', \overline{y}) - \boldsymbol{\theta}^\top \boldsymbol{\phi}(\overline{x}, \overline{y}) + \boldsymbol{\theta}^\top \boldsymbol{\phi}(\overline{x}, \overline{y}') \right] \quad (9) \end{aligned}$$

which is an upper bound on the cost of the model's predictions [22]. The hinge loss is not differentiable, but a subgradient can be computed efficiently, again assuming the cost factors over the segments:

$$\nabla \ell_{\text{hinge}}(\boldsymbol{\theta}; \overline{x}, \overline{y}) = -\boldsymbol{\phi}(\overline{x}, \overline{y}) + \boldsymbol{\phi}(\overline{x}, \hat{y}), \quad (10)$$

where $\hat{y} = \text{argmax}_{\overline{y}'} \boldsymbol{\theta}^\top \boldsymbol{\phi}(\overline{x}, \overline{y}') + \mu \, \text{cost}(\overline{y}', \overline{y})$, and the cost weight μ equals 1 for standard hinge loss. Computing $\hat{y}$ is referred to as **cost-augmented inference**, which finds a hypothesis that has both a high score under the model and a high cost. Minimizing the hinge loss attempts to calibrate the model score so that score differences are equivalent to differences in cost. The loss in Eq. (9) is the same as that used in margin-rescaled structured support vector machines [21, 23].

The third loss is the structured ramp loss [22, 24]

$$\begin{aligned} \ell_{\text{ramp}}(\boldsymbol{\theta}; \overline{x}, \overline{y}) = &-\max_{\overline{y}'} \left[\mu_1 \, \text{cost}(\overline{y}', \overline{y}) + \boldsymbol{\theta}^\top \boldsymbol{\phi}(\overline{x}, \overline{y}') \right] \\ &+ \max_{\overline{y}''} \left[\mu_2 \, \text{cost}(\overline{y}'', \overline{y}) + \boldsymbol{\theta}^\top \boldsymbol{\phi}(\overline{x}, \overline{y}'') \right] \quad (11) \end{aligned}$$

where μ_1 and μ_2 are cost weights. When $\mu_1 = 0$ and $\mu_2 = 1$, this becomes the standard ramp loss [22], which is a tighter upper bound on the cost function than hinge. As $\mu_1 \to -\infty$, ramp becomes hinge loss. Therefore, if μ_1 and μ_2 are tuned, ramp loss should perform at least as well as hinge.

Log loss is commonly used in ASR in the context of standard maximum mutual information (MMI) [25] and SCRF training, and EBR is used in MPE/MWE [10]. Hinge loss has been used in ASR, but less commonly [18]. Ramp loss has not been previously used in ASR. In terms of convexity, log loss and hinge loss are convex while ramp and EBR are not.

All of the above gradients involve summing or searching over the support of $p(\overline{y} | \overline{x})$, i.e., the entire space $\mathcal{Y}$, which is typically intractable. To handle this, we follow the common approach of using a baseline recognizer to generate a lattice of possible outputs. A lattice is a compact structure for storing high-probability samples from $p(\overline{y} | \overline{x})$, and we use lattices to narrow our space for summing and searching.

Formally, let $\mathcal{P}_{\text{d}} \subset \mathcal{Y}$ be the set of paths in a lattice output by a baseline recognizer (the "denominator lattice"), and $\mathcal{P}_{\text{n}}$ the set of paths in the ground-truth lattice (the "numerator lattice"). For $\mathcal{P}_{\text{n}}$ we can use a forced alignment with the ground truth, which is often not in $\mathcal{P}_{\text{d}}$, in which case we also add the path in $\mathcal{P}_{\text{n}}$ to $\mathcal{P}_{\text{d}}$. An alternative is to use the oracle path (a path with minimum error) in $\mathcal{P}_{\text{d}}$ as $\mathcal{P}_{\text{n}}$. Given the numerator and denominator lattices, the gradient for CL becomes

$$\nabla \ell_{\text{CL}}(\boldsymbol{\theta}; \overline{x}, (\overline{w}, \overline{q})) = -\boldsymbol{\phi}(\overline{x}, \overline{y}') + \sum_{\overline{y}'' \in \mathcal{P}_{\text{d}}} p(\overline{y}'' | \overline{x}) \boldsymbol{\phi}(\overline{x}, \overline{y}''),$$

and other loss gradients are modified similarly. When computing the subgradient for hinge and ramp loss, cost-augmented inference becomes a search over the lattice $\mathcal{P}_{\text{d}}$ with the cost function added to the scoring function.

We optimize the loss functions with either Rprop or subgradient-based methods, e.g., SGD or AdaGrad. We also include L_1 regularization with either dual averaging [26] for AdaGrad, thresholding [27] for Rprop [28], or SMIDAS [29] for vanilla subgradient descent.

4. Cost Functions

One popular cost function was proposed in [10] in the context of MPE/MWE training, and we refer to it as **MPE cost**. The cost of a path is the sum of costs of individual edges. The cost of an edge is the non-overlaping part of a matching ground-truth edge that gives the lowest error, where the error is one if the label is correct and 0.5 otherwise. Formally, for any hypothesized edge e, we define

$$\begin{aligned} &\text{cost}_{\text{MPE}}(e, \overline{y}) \qquad (12)\\ &= 1 - \max_{e' \in \overline{y}} \left[\mathbb{1}_{\hat{w}_e = \overline{w}_{e'}} \frac{|e \cap e'|}{|e'|} + \frac{1}{2} \mathbb{1}_{\hat{w}_e \neq \overline{w}_{e'}} \frac{|e \cap e'|}{|e'|} \right], \end{aligned}$$

and $\text{cost}_{\text{MPE}}(\hat{y}, \overline{y}) = \sum_{e \in \hat{y}} \text{cost}_{\text{MPE}}(e, \overline{y})$, where $|e|$ denotes the length of segment e.

The MPE cost only penalizes false negatives and does not account for false positives. Therefore, we propose an alternative that we refer to as the **overlap cost**:

$$\text{cost}_{\text{overlap}}(e, \overline{y}) = 1 - \mathbb{1}_{\hat{w}_e = \overline{w}_{\tilde{e}}} \frac{|e \cap \tilde{e}|}{|e \cup \tilde{e}|}, \qquad (13)$$

where $\tilde{e} = \text{argmax}_{e' \in \overline{y}} |e' \cap e|$. This cost function finds the most overlapping edge in the ground truth and considers any part of the union of the two edges that is not overlapping to be in error. The cost for the whole path is again $\text{cost}_{\text{overlap}}(\hat{y}, \overline{y}) = \sum_{e \in \hat{y}} \text{cost}_{\text{overlap}}(e, \overline{y})$.

5. Experiments

We study the various losses and cost functions on two tasks. One is a standard speech recognition task, namely TIMIT phonetic recognition. The second is a sign language recognition task from video, in particular recognition of fingerspelled letter sequences in American Sign Language (ASL). Both are tasks on which there is prior work using SCRFs [30, 31], and both are small enough (in terms of data set size and decoding search space) to run many empirical comparisons in a reasonable amount of time. For the ASL task, we use the data and experimental setup of [31]: We obtain baseline lattices using their tandem HMM-based system [31], and we use the same set of segmental feature functions. Numerator lattices are forced alignments of the ground truth transcriptions. We train all models from all-zero weights and optimize with Rprop for 20 epochs. We use L_1 and L_2 regularization, with parameters tuned over the grid $\{0, 10^{-6}, 10^{-5}, \dots, 0.1, 1\}^2$. For hinge and ramp loss, we use the standard forms without tuning the cost weights (i.e., $\mu = 1$, $\mu_1 = 0$, and $\mu_2 = 1$).

For TIMIT, we use the standard 3696-utterance training set and 192-utterance core test set, plus a random 192 utterances from the full test set (excluding the core test set) as a development set. We use lattices generated by a baseline monophone HMM system with 39-dimensional MFCCs. The resulting lattices have an average density (average number of hypothesized edges per ground truth edge) of 60.1. The oracle phone error rate is 6.3% for the development set and 7.0% for the core test set. The numerator lattices are the oracle paths (paths with minimum phone error) from the denominator lattices; each numerator lattice contains a single path. We implement segmental models with various feature functions. The base features are the acoustic and language model score, and a bias (a feature that is always one). We also include a set of features based on spectro-temporal receptive fields implemented as follows. We begin with 40-dimensional log mel filter bank features. For each segment, we divide it evenly into thirds in both time and frequency, resulting in nine patches for each segment. For each patch, we have a 3×13 receptive field of all ones, and convolve it with the patch. The resulting $3 \times 13 \times 9$ numbers are lexicalized to form the final features for the segmental model. We optimize the loss functions using AdaGrad, using step size 0.1 for 10 epochs. L_1 and L_2 regularization parameters are tuned over the grid $\{0, 0.001, 0.1, 1\} \times \{0, 0.1, 1, 100\}$.

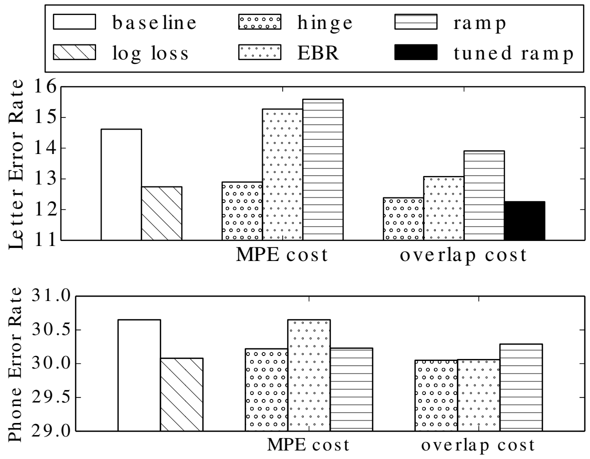

Figure 1: *Top*: ASL results. *Bottom*: TIMIT results.

The results for ASL recognition, averaged over four signers, are shown in the upper plot of Figure 1. The evaluation metric is the letter error rate, which is the percentage of letters that are substituted, inserted, or deleted. The results for TIMIT are shown in the lower plot of Figure 1. We observe three consistent conclusions:

- Across losses, overlap cost is better than MPE cost.
- Hinge loss with overlap cost is the best performer, but this is only by a small margin, and log loss is very competitive even without using an explicit cost function.
- Non-convex losses (ramp and EBR) are difficult to optimize and therefore achieve inconsistent results. We suspect a warm start might be able to remedy this.

For the ASL task, we tuned on a development set the cost weights for ramp loss over the grid $\{-100, -10, -1, -0.1, -0.01\} \times \{0.01, 0.1, 1, 10, 100\}$, using overlap cost. The test result of tuned ramp slightly improves over hinge loss, confirming that if ramp is tuned carefully, it is able to outperform hinge. However, even though tuned ramp loss achieves very good results, considering the time spent tuning μ_1 and μ_2, we still favor hinge and log loss.

The running times for calculating the gradients for different losses differ by a constant factor, the number of forward-backward passes required. Hinge loss requires one forward search, log loss requires one forward sum and one backward sum, ramp requires one forward and one backward search, and

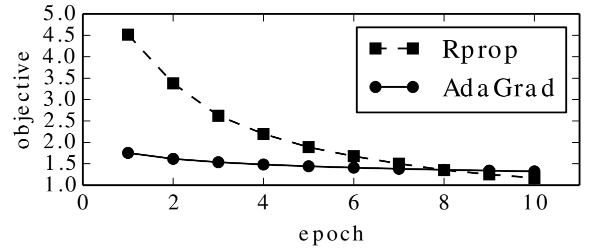

Figure 2: A comparison of convergence rate for Rprop and AdaGrad for ASL recognition on a single signer and a single fold.

EBR requires three forward sums and three backward sums. In terms of convergence, although Rprop may ultimately achieve a better objective, AdaGrad converges faster; see Figure 2. Rprop normally requires 4-5 epochs to gain enough statistics for determinining robust step sizes, while AdaGrad requires just one epoch to hone in on the target region.

We also conducted experiments to determine how the results are affected by different levels of noise in the feature functions, using simulated phone detector-based features. Similarly to [32], we define a detection event as a (time, phone label) pair, and a feature function that is an indicator of whether a phone detection event occurs in the time span of the edge. If we set a high weight for the phone event that occurs in an edge with the same phone label, then we can exactly recover the oracle path. This allows us to conduct a series of simulated experiments with different amounts of noise added to the oracle phone events, or gold events. For all experiments below, we use the same TIMIT setting except that we only use the acoustic and language model score with the simulated phone detector features, with no regularizer and one epoch of AdaGrad. The ramp cost weights are set to $\mu_1 = 0, \mu_2 = 1$. The cost weight for hinge is tuned over $\{0.01, 0.1, 1, 10, 100\}$ and results are only shown for the best-performing value.

The first set of experiments (Figure 3, top left) randomly perturbs the correct phone label of each event to an incorrect label with the corruption probability shown on the x-axis; the event times are not perturbed. The second set of experiments (Figure 3, top right) perturbs the time for each event but not the label. We add Gaussian noise with mean set to the time at which the event occurs and with several standard deviations shown on the x-axis. For the third and fourth set of experiments (Figure 3, bottom), we randomly include an edge in the lattice as a false positive event, or randomly delete an event from the gold events.

The conclusions are consistent with our previous observation, namely, that hinge is the consistent winner but only by a very small margin, that log loss is very competitive, that nonconvex losses are hard to optimize, and that overlap cost is better than MPE cost. As a byproduct, we note that we could achieve the current state-of-the-art 17.7% [33] given a phone detector with any of the following characteristics: up to 50% phone error rate but perfect time information, up to 5-frame time perturbations (in standard deviation) but perfect labels, 1.8 false positives per gold edge, or 20% false negatives.

6. Conclusion

Based on our own and others' prior work , we have motivated comparing different training approaches for discriminative segmental models. We have compared log loss to cost-sensitive losses (hinge, empirical Bayes risk, and ramp loss) on two quite different tasks and under different conditions. We have proposed the overlap cost function and have shown that it consistently outperforms the MPE cost function which is commonly used in frame-based recognition. In general, hinge loss with overlap cost achieves consistently strong results, but only slightly better than log loss. Along the way, we have implemented the various losses, costs, and optimization algorithms in a new toolkit for discriminative segmental models (to be released publicly upon publication), and have found that optimization with AdaGrad is much faster than with the more commonly used Rprop for segmental models. For larger tasks (more feature functions and/or larger vocabularies), we expect to also see a speed benefit for hinge loss over log loss, as hinge loss encourages sparsity in the weight vectors.

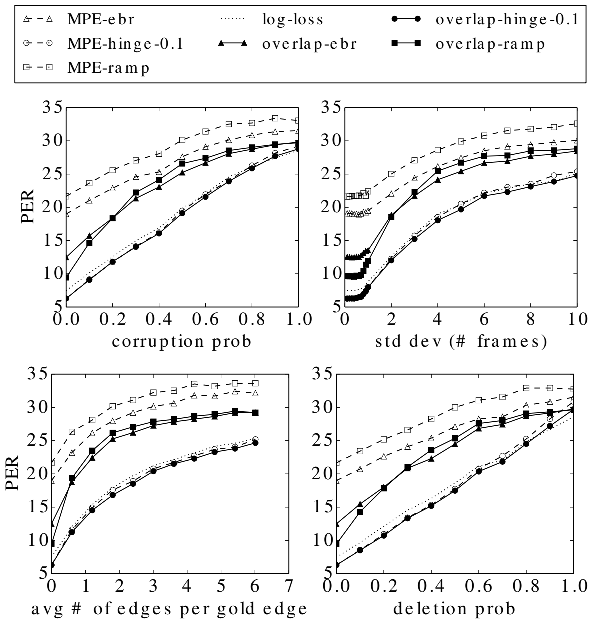

Figure 3: *Top Left*: perturbing phone labels. *Top Right*: perturbing time. *Bottom Left*: Adding false positive events. *Bottom Right*: Adding false negative events.

One of the interesting findings is that log loss is indeed difficult to beat, although for other types of models it has been found to more clearly underperform cost-sensitive losses. It is interesting to consider the possible reasons for this. The main differences between the current work and prior work with cost-sensitive losses include segmental (vs. frame-based) modeling and lattice (vs. first-pass) decoding. It will therefore be interesting in future work to tease apart these effects by comparing with a frame-based version of our experiments and a lattice-rescoring version of typical discriminative approaches. It is also interesting to note that minimizing log loss is equivalent to minimizing EBR with the cost $\mathbb{1}_{y \neq y'}$. We are investigating how hinge and ramp loss perform for this cost function to shed some additional light on why log loss performs so well without the explicit cost function. Finally, another advantage for log loss is its smoothness; to test the effect of smoothness, it will be interesting to compare boosted MMI [34] and hinge loss as in [35].

7. Acknowledgements

This research was supported by NSF grants IIS-0905633 and NSF-1433485. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency.

8. References

[1] M. Ostendorf and S. Roukos, "A stochastic segment model for phoneme-based continuous speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 1857–1869, 1989.

[2] J. R. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech and Language*, vol. 17, no. 2, pp. 137–152, 2003.

[3] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle, "Template-based continuous speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 15, no. 4, pp. 1377–1390, 2007.

[4] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009, pp. 152–157.

[5] G. Zweig, P. Nguyen, D. Van Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, M. Wang, F. Sha, H. Hermansky *et al.*, "Speech recognition with segmental conditional random fields: A summary of the JHU CLSP 2010 Summer Workshop," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 5044–5047.

[6] S. Sarawagi and W. W. Cohen, "Semi-Markov conditional random fields for information extraction," in *Proc. Neural Information Processing Systems (NIPS)*, 2004, pp. 1185–1192.

[7] K. Demuynck, D. Seppi, D. Van Compernolle, P. Nguyen, and G. Zweig, "Integrating meta-information into exemplar-based speech recognition with segmental conditional random fields," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 5048–5051.

[8] G. Heigold, H. Ney, R. Schlüter, and S. Wiesler, "Discriminative training for automatic speech recognition: Modeling, criteria, optimization, implementation, and performance," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 58–69, 2012.

[9] J. Kaiser, B. Horvat, and Z. Kacic, "A novel loss function for the overall risk criterion based discriminative training of HMM models," in *Proc. International Conference on Spoken Language Processing (ICSLP)*, 2000.

[10] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 2002, pp. I–105.

[11] V. Doumpiotis and W. Byrne, "Lattice segmentation and minimum bayes risk discriminative training for large vocabulary continuous speech recognition," *Speech Communication*, vol. 48, no. 2, pp. 142–160, 2006.

[12] S.-X. Zhang and M. Gales, "Structured support vector machines for noise robust continuous speech recognition," in *Proc. Interspeech*, 2011, pp. 989–992.

[13] F. Sha and L. K. Saul, "Large margin hidden Markov models for automatic speech recognition," *Proc. Neural Information Processing Systems (NIPS)*, vol. 19, p. 1249, 2007.

[14] H. Jiang, X. Li, and C. Liu, "Large margin hidden Markov models for speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 14, no. 5, pp. 1584–1595, 2006.

[15] D. Yu, L. Deng, X. He, and A. Acero, "Large-margin minimum classification error training for large-scale speech recognition tasks," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2007, pp. IV–1137.

[16] H. Tang, J. Keshet, and K. Livescu, "Discriminative pronunciation modeling: A large-margin, feature-rich approach," in *Proc. Association for Computational Linguistics (ACL)*, 2012.

[17] A. Ragni and M. Gales, "Derivative kernels for noise robust ASR," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[18] S.-X. Zhang, A. Ragni, and M. J. F. Gales, "Structured log linear models for noise robust speech recognition," *Signal Processing Letters, IEEE*, vol. 17, no. 11, pp. 945–948, 2010.

[19] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[20] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. International Conference on Machine Learning (ICML)*, 2001.

[21] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin, "Learning structured prediction models: A large margin approach," in *Proc. International Conference on Machine Learning (ICML)*, 2005.

[22] C. B. Do, Q. Le, C. H. Teo, O. Chapelle, and A. Smola, "Tighter bounds for structured estimation," in *Proc. Neural Information Processing Systems (NIPS)*, 2008.

[23] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[24] K. Gimpel and N. A. Smith, "Structured ramp loss minimization for machine translation," in *Proc. Human Language Technology/Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, 2012.

[25] L. Bahl, P. Brown, P. V. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, vol. 11. IEEE, 1986, pp. 49–52.

[26] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization." *Journal of Machine Learning Research*, vol. 11, no. 2543-2596, p. 4, 2010.

[27] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty," in *ACL/AFNLP*. Association for Computational Linguistics, 2009, pp. 477–485.

[28] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. IEEE International Conference on Neural Networks (ICNN)*, 1993, pp. 586–591.

[29] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for l1-regularized loss minimization," *Journal of Machine Learning Research*, vol. 12, pp. 1865–1892, 2011.

[30] G. Zweig, "Classification and recognition with direct segment models," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 4161–4164.

[31] T. Kim, G. Shakhnarovich, and K. Livescu, "Fingerspelling recognition with semi-Markov conditional random fields," in *Proc. International Conference on Computer Vision (ICCV)*, 2013.

[32] G. Zweig and P. Nguyen, "SCARF: A segmental conditional random field toolkit for speech recognition," *Proc. Interspeech*, 2010.

[33] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013, pp. 6645–6649.

[34] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 4057–4060.

[35] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney, "Modified MMI/MPE: A direct evaluation of the margin in speech recognition," in *Proc. International Conference on Machine Learning (ICML)*, 2008, pp. 384–391.