Extending Noise Robust Structured Support Vector Machines to Larger Vocabulary Tasks

Shi-Xiong Zhang, M. J. F. Gales

Cambridge University Engineering Department
Trumpington St., Cambridge, CB2 1PZ, UK
{sxz20,mfjg}@eng.cam.ac.uk

***Abstract*—This paper describes a structured SVM framework suitable for noise-robust medium/large vocabulary speech recognition. Several theoretical and practical extensions to previous work on small vocabulary tasks are detailed. The joint feature space based on word models is extended to allow context-dependent triphone models to be used. By interpreting the structured SVM as a large margin log-linear model, illustrates that there is an implicit assumption that the prior of the discriminative parameter is a zero mean Gaussian. However, depending on the definition of likelihood feature space, a non-zero prior may be more appropriate. A general Gaussian prior is incorporated into the large margin training criterion in a form that allows the cutting plan algorithm to be directly applied. To further speed up the training process, 1-slack algorithm, caching competing hypothesis and parallelization strategies are also proposed. The performance of structured SVMs is evaluated on noise corrupted medium vocabulary speech recognition task: AURORA 4.**

I. Introduction

Most automatic speech recognition (ASR) systems use generative models, in the form of hidden Markov models (HMMs) combined with class priors, the language model to yield the sentence posterior based on Bayes' rule. Although discriminative training can be performed, the underlying models are still generative. This has led to interest in discriminative models, e.g., Structured Conditional Random Fields (SCRF) [1], and structured Log Linear Model (LLM) [2], [3], where the posterior of the word-sequence given the observation is *directly* modelled. For these discriminative models three important decisions need to be made: the form of the features to use; the appropriate training criterion; and how to handle continuous speech.

A number of features have been investigated at the frame, model and word level [1], [4]. Features based on generative models are an attractive option as they allow state-of-the-art speaker adaptation and noise robustness approaches for generative models to be used [5]. Discriminative models are often trained using Conditional Maximum Likelihood (CML) [1], [2]. Alternatively, there has been interest in large margin [4], [6] and minimum Bayes' risk [7] criteria. To handle continuous speech, structured discriminative models require a segmentation of the frames into word, or sub-word units. Usually these segmentations are generated by standard HMM acoustic models. Gains are observed by optimising these segmentations based on the discriminative models parameters in both training and decoding [8]. In previous work with structured SVMs, a small vocabulary noise corrupted digit string recognition task based on whole-word HMMs was examined [4], [8].

This paper extends the previous framework of structured SVMs to handle medium/large vocabulary continuous speech recognition tasks. By interpreting the structured SVM as a large margin log-linear model, illustrates that there is an implicit assumption that the prior of the discriminative parameter is a zero mean Gaussian. However, depending on the property of log-likelihood feature space, the mean of prior should not be zeros. We relax this assumption by incorporating a more general Gaussian prior into the large margin training criterion, in a form that allows the cutting plan algorithm to be directly applied. The generalized criterion will not only lead to better trained parameters, but also help to reduces the convergence time in large scale application. In order to solve the resulted optimisation problem on larger tasks, 1-slack algorithm has to be used to replace the previous n-slack algorithm for reducing the number of constraints. To further speed up the training process, caching and parallelization strategies are also proposed. Experimental results are presented on medium to large vocabulary noise-corrupted ASR tasks: AURORA 4.

II. Structured Support Vector Machines

Consider a training set with R data pairs, $\left\{\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\texttt{ref}}\right\}_{r=1}^{R}$, where $\mathbf{O}^{(r)} = \{\boldsymbol{o}_1^{(r)}, \ldots, \boldsymbol{o}_T^{(r)}\}$ is an observation sequence and $\mathbf{w}^{(r)}_{\texttt{ref}} = \{w_1^{(r)}, \ldots, w_{|\mathbf{w}|}^{(r)}\}$ is the reference labels. In structured SVM for continuous speech recognition [8], our goal is to find a discriminant function $\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta})$ that measures how well the $\mathbf{w}$ matches the given $\mathbf{O}$, such that

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w},\boldsymbol{\theta}} \left\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta})\right\} \tag{1}$$

is the predicted label sequence for observations $\mathbf{O}$, where $\boldsymbol{\alpha}$ is the discriminative parameter vector, $\boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta})$ is a *joint* feature vector and $\boldsymbol{\theta}$ is the hidden variable that segments the observations $\mathbf{O}_{1:T}$ into $|\mathbf{w}|$ corresponding labels. To extend previous structured SVMs system [4], [8] to medium/large vocabulary continuous speech recognition three important decisions need to be made: at which level the hidden variable $\boldsymbol{\theta}$ segments the continuous speech; the form of the features to use based on context-dependent sub-word models; and the appropriate training criterion with efficient learning algorithm.

978-1-4673-0367-5/11/$26.00 ©2011 IEEE

A. Joint Feature Space

This section describes the features to be used by structured SVMs for medium/large vocabulary ASR. In previous small vocabulary system [4], [8], the observations are segmented at the word level, however in order to extend the structured SVMs to medium-large vocabulary tasks data has to be segmented at sub-word level, such as phones. Given an alignment $\boldsymbol{\theta}$ splits the observation sequence into $|\mathbf{w}|$ segments $\mathbf{O} = \{\mathbf{O}_{t(w_1,\boldsymbol{\theta})}, \dots, \mathbf{O}_{t(w_i,\boldsymbol{\theta})}, \dots, \mathbf{O}_{t(w_{|\mathbf{w}|},\boldsymbol{\theta})}\}$, where w_i is the context-dependent phone label in this work. The resulting *joint* feature space is defined as

$$\boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta}) \triangleq \begin{bmatrix} \sum_{i=1}^{|\mathbf{w}|} \boldsymbol{\delta}(w_i) \otimes \boldsymbol{\varphi}(\mathbf{O}_{t(w_i,\boldsymbol{\theta})}) \\ \log P(\mathbf{w}) \end{bmatrix} \tag{2}$$

where $P(\mathbf{w})$ is the standard n-gram language model probability, $\otimes$ is the tensor product, $\boldsymbol{\delta}(w_i)$ is a sparse vector indicate the position of w_i in the dictionary $\{v_k\}_{k=1}^{M}$ and $\boldsymbol{\varphi}(\mathbf{O}_{t(w_i,\boldsymbol{\theta})})$ is the generative model based log likelihood feature space for segment $\mathbf{O}_{t(w_i,\boldsymbol{\theta})}$,

$$\boldsymbol{\delta}(w) = \begin{bmatrix} \delta(w - v_1) \\ \vdots \\ \delta(w - v_{\mathrm{M}}) \end{bmatrix}, \boldsymbol{\varphi}(\mathbf{O}) = \begin{bmatrix} \log(p(\mathbf{O}; \boldsymbol{\lambda}^{(v_1)})) \\ \vdots \\ \log(p(\mathbf{O}; \boldsymbol{\lambda}^{(v_{\mathrm{M}})})) \end{bmatrix} \tag{3}$$

where $\boldsymbol{\lambda}$ is the generative model parameters. These generative model based features allows standard noise and speaker adaptation schemes to be used to derive robust feature space. Thus the dot-product of the $\boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta})$ and structured SVM parameter $\boldsymbol{\alpha}$ can be evaluated by accumulating every segment score [4]

$$\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta}) = \sum_{i=1}^{|\mathbf{w}|} {\boldsymbol{\alpha}^{(w_i)}}^{\mathsf{T}} \boldsymbol{\varphi}(\mathbf{O}_{t(w_i,\boldsymbol{\theta})}) + \alpha_{\mathtt{lm}} \log P(\mathbf{w}), \tag{4}$$

where $\boldsymbol{\alpha} = [{\boldsymbol{\alpha}^{(v_1)}}^{\mathsf{T}}, \dots {\boldsymbol{\alpha}^{(v_k)}}^{\mathsf{T}} \dots, {\boldsymbol{\alpha}^{(v_{\mathrm{M}})}}^{\mathsf{T}}, \alpha_{\mathtt{lm}}]^{\mathsf{T}}$. Fig. 1 demonstrates an example of using Eq. (2) to construct joint feature space for data pair $(\mathbf{O}, \mathbf{w})$ given segmentation $\boldsymbol{\theta}$.

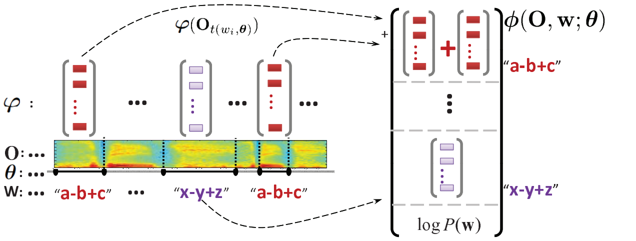
$\boldsymbol{\varphi}(\mathbf{O}_{t(w_i,\boldsymbol{\theta})})$
$\boldsymbol{\varphi}$:
+
+
$\boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta})$
"a-b+c"
O: ...
θ: ...
W: ... "a-b+c" ... "x-y+z" "a-b+c" ...
"x-y+z"
log P(w)

Fig. 1. Illustration of constructing *joint* feature space from triphone HMMs based log-likelihood feature space.

When medium/large vocabulary ASR are considered there is an issue with above context-dependent feature space [3]. The set of context-dependent phone models $\{v_k\}_{k=1}^{M}$ yields a very large joint feature space. Although in theory this could be used, the number of determinative model parameters becomes large. Two approaches proposed in [3] to address this problem are adopted this work. One is to reduce the dimension of the feature space $\boldsymbol{\varphi}(\cdot)$ by selecting a small set of "suitable" generative models. Here only the generative models that share the same observed context are included. For example, the feature space of segment $\mathbf{O}_{t(a-x+c)}$ with *matched context* can be expressed as

$$\boldsymbol{\varphi}(\mathbf{O}_{t(a-x+c)}) = \begin{bmatrix} \log p(\mathbf{O}_{t(a-x+c)}; \boldsymbol{\lambda}^{(a-a+c)}) \\ \vdots \\ \log p(\mathbf{O}_{t(a-x+c)}; \boldsymbol{\lambda}^{(a-y+c)}) \\ \log p(\mathbf{O}_{t(a-x+c)}; \boldsymbol{\lambda}^{(a-z+c)}) \end{bmatrix}_{M_1}. \tag{5}$$

This reduces the dimensionality of the feature space $\boldsymbol{\varphi}(\cdot)$ from the number of context-dependent phones M to the number of mono phones M_1. The second approach is to reduce the dimension of sparse vector $\boldsymbol{\delta}(\cdot)$ by clustering the $\{v_k\}_{k=1}^{M}$ using a phonetic decision tree. This is actually the model-level parameter tying described in [3] where $\boldsymbol{\alpha}^{(v_i)}$ and $\boldsymbol{\alpha}^{(v_j)}$ are tied if v_i and v_j belongs to the same leaf node. Thus the total dimensionality of joint feature space is $M_2 \times M_1$. In this work, M_1 and M_2 are both set to 47, the number of mono phones.

The joint feature space described above is based on a fixed segmentation. For a specific data pair $(\mathbf{O}, \mathbf{w})$, the "most likely" segmentation $\hat{\boldsymbol{\theta}}$ is considered,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta}). \tag{6}$$

B. Large Margin Training

Given the training data pairs, $\left\{\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}\right\}_{r=1}^{R}$, the parameters of structured SVM can be trained by solving the following optimisation problem [8], [9]:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \frac{1}{2}||\boldsymbol{\alpha}||^2 + \frac{C}{R}\sum_{r=1}^{R} \xi_r \tag{7}$$

$$\text{s.t.} \max_{\boldsymbol{\theta}^{(r)}} \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}; \boldsymbol{\theta}^{(r)}) - \max_{\boldsymbol{\theta}_*^{(r)}} \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_*^{(r)}; \boldsymbol{\theta}_*^{(r)})$$
$$\geq \mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)}, \mathbf{w}_*^{(r)}) - \xi_r, \qquad \forall\ \mathbf{w}_*^{(r)} \neq \mathbf{w}_{\mathtt{ref}}^{(r)},\ r = 1, \dots, R,$$

where $\xi_r \geq 0$ are the slack variables and $\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)}, \mathbf{w}_*^{(r)})$ is the loss function between reference $\mathbf{w}_{\mathtt{ref}}^{(r)}$ and its competing hypothesis $\mathbf{w}_*^{(r)}$. The constraints in Eq. (7) can be explained as follows. For every training pair $(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)})$, the best score of the correct pair should be greater than all competing pairs $(\mathbf{O}^{(r)}, \mathbf{w}_*^{(r)})$ by a margin determined by the loss. Note that since the number of possible competing hypothesis $\mathbf{w}_*^{(r)}$ is very large, there are lots of constraints in Eq. (7).

Substituting the slack variable in the constraints to the objective function, the structured SVMs problem in Eq. (7) can also be expressed as a *minimisation* of

$$\frac{1}{2}||\boldsymbol{\alpha}||_2^2 + \frac{C}{R}\sum_{r=1}^{R} \Big[\overbrace{- \max_{\boldsymbol{\theta}^{(r)}} \left(\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}; \boldsymbol{\theta}^{(r)}) \right)}^{\text{concave}}$$
$$\underbrace{+ \max_{\mathbf{w} \neq \mathbf{w}_{\mathtt{ref}}^{(r)}, \boldsymbol{\theta}} \left\{ \mathcal{L}(\mathbf{w}, \mathbf{w}_{\mathtt{ref}}^{(r)}) + \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta}) \right\}}_{\text{convex}} \Big]_+ \tag{8}$$

where $[\,]_+$ is the hinge-loss function. The constraints in Eq. (8) include two maximum of a set of linear functions. Each maximum function is convex with respect to $\boldsymbol{\alpha}$. However, the objective function in Eq. (7), as also shown in Eq. (8), is non-convex. To solve this non-convex optimization problem, an algorithm based on concave-convex procedure [10] and cutting plane algorithm [9] is proposed in previous work [8].

C. Relationship with Log Linear Models

The structured SVMs problems formulated in Eq. (1) and (7) can be interpreted as decoding and large margin training of log linear models. To see this, we write the posterior of the hypothesized labels $\mathbf{w}$ given $\mathbf{O}$ as a member of exponential family,

$$P(\mathbf{w}|\mathbf{O};\boldsymbol{\alpha},\hat{\boldsymbol{\theta}}) = \frac{\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\hat{\boldsymbol{\theta}})\right)}{Z(\mathbf{O};\boldsymbol{\alpha})}, \tag{9}$$

where $Z(\mathbf{O};\boldsymbol{\alpha}) = \sum_{\mathbf{w}'}\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w}';\hat{\boldsymbol{\theta}}')\right)$ ensures that the model is a properly normalized probability, $\hat{\boldsymbol{\theta}}$ is the best alignment that maximises posterior probability $P(\mathbf{w}|\mathbf{O};\boldsymbol{\alpha},\boldsymbol{\theta})$, $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta})$. Recognition with this log linear model can be simply expressed as

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}}\ P(\mathbf{w}|\mathbf{O};\boldsymbol{\alpha},\hat{\boldsymbol{\theta}}) = \arg\max_{\mathbf{w},\boldsymbol{\theta}}\ \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta}) \tag{10}$$

This is equivalent to structured SVM decoding in Eq. (1).

In order to train a robust model capable of generalizing well on high-dimension space even with limited data, large margin based approaches can be applied [6], [11]. If the margin for log linear models is defined as the log posterior probability ratio of the reference $\{\mathbf{w}^{(r)}_{\mathtt{ref}},\ \hat{\boldsymbol{\theta}}^{(r)}\}$ and best competing hypothesis/alignment $\{\mathbf{w},\hat{\boldsymbol{\theta}}\}$, the large margin training for log linear model can be expressed as minimising

$$\mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha}) = \frac{1}{R}\cdot\sum\nolimits_{r=1}^{R} \tag{11}$$
$$\left[\max_{\mathbf{w}\neq\mathbf{w}^{(r)}_{\mathtt{ref}}}\left\{\mathcal{L}(\mathbf{w},\mathbf{w}^{(r)}_{\mathtt{ref}}) - \log\left(\frac{P(\mathbf{w}^{(r)}_{\mathtt{ref}}|\mathbf{O}^{(r)};\boldsymbol{\alpha},\hat{\boldsymbol{\theta}}^{(r)})}{P(\mathbf{w}|\mathbf{O}^{(r)};\boldsymbol{\alpha},\hat{\boldsymbol{\theta}})}\right)\right\}\right]_{+}$$

where $[\ \cdot\ ]_{+}$ is the hinge-loss function. Substituting Eq. (9) into Eq. (11) and canceling out the normalization term $Z(\mathbf{O};\boldsymbol{\alpha})$, we will have

$$\begin{aligned}\mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha}) = \frac{1}{R}\cdot\sum_{r=1}^{R}\Bigg[&-\max_{\boldsymbol{\theta}^{(r)}}\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}^{(r)}_{\mathtt{ref}};\boldsymbol{\theta}^{(r)})\right)\\ &+\max_{\mathbf{w}\neq\mathbf{w}^{(r)}_{\mathtt{ref}},\boldsymbol{\theta}}\left\{\mathcal{L}(\mathbf{w},\mathbf{w}^{(r)}_{\mathtt{ref}}) + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w};\boldsymbol{\theta})\right\}\Bigg]_{+}\end{aligned} \tag{12}$$

Note that above criterion is the unregularized part of Eq. (8). To retrieve the regularization term $\frac{1}{2}||\boldsymbol{\alpha}||^2_2$, a standard Gaussian distribution $\mathcal{N}(\boldsymbol{\alpha};\mathbf{0},C\mathbf{I})$ can be incorporated into the criterion as the prior probability $P(\boldsymbol{\alpha})$,

$$\mathcal{F}(\boldsymbol{\alpha}) = \mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha}) - \log\left(\mathcal{N}(\boldsymbol{\alpha};\mathbf{0},C\mathbf{I})\right) \tag{13}$$

where the log prior $\log\left(\mathcal{N}(\boldsymbol{\alpha};\mathbf{0},C\mathbf{I})\right) = -\frac{1}{2C}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\alpha} + \text{const.}$ Ignoring the terms that constant to $\boldsymbol{\alpha}$, yields the following regularized objective

$$\mathcal{F}(\boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{\alpha}||^2_2 + C\cdot\mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha}) \tag{14}$$

Comparison between Eq. (14) and Eq. (8) suggests that, the structured SVM used in this work can also be viewed as a large margin trained log linear model with "most discriminative" segmentation.

III. Gaussian Prior

The previous section has shown that, when training the standard structured SVMs, an implicit assumption is made that the prior distribution of $\boldsymbol{\alpha}$ is standard Gaussian, with zero mean and identity covariance matrix. However, depending on the feature space defined in Eq. (3), the mean of prior, $\boldsymbol{\mu}$, should not be zeros. A proper mean of prior should be the one that can yield the HMM baseline performance

$$\arg\max_{\mathbf{w},\boldsymbol{\theta}}\boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta}) = \arg\max_{\mathbf{w}}\log\left(P(\mathbf{O}|\mathbf{w};\boldsymbol{\lambda})^{\frac{1}{\alpha_{\mathtt{lm}}}}P(\mathbf{w})\right)$$ [1]

which implies that the value of $\boldsymbol{\mu}$ is one for the correct class, zero otherwise, thus for class v_1, $\boldsymbol{\mu}^{(v_1)} = [1,0,\ldots,0]^{\mathsf{T}}$. This motivates us to look for a more general large margin training criterion that can relax the structured SVMs prior assumption and incorporate a general Gaussian prior

$$\tilde{\mathcal{F}}(\boldsymbol{\alpha}) = \mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha}) - \log\left(P(\boldsymbol{\alpha})\right) \tag{15}$$

where $P(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\alpha};\boldsymbol{\mu},\boldsymbol{\Sigma})$. Thus, the objective function of structured SVMs training (Eq. (8)) can be generalized as

$$\tilde{\mathcal{F}}(\boldsymbol{\alpha}) = \frac{1}{2}\left(\boldsymbol{\alpha}-\boldsymbol{\mu}\right)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\alpha}-\boldsymbol{\mu}\right) + \mathcal{F}_{\mathtt{lm}}(\boldsymbol{\alpha}). \tag{16}$$

Note that the normalization term $\boldsymbol{\Sigma}^{-1}$ can always be decomposed and merged into the feature space by using transformed features $\tilde{\boldsymbol{\phi}}(\mathbf{O},\mathbf{w};\boldsymbol{\theta}) = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta})$. In this work, we assume the log-likelihood features are already properly scaled, and simply using $\boldsymbol{\Sigma} = C\boldsymbol{I}$. In order to utilize the training framework for Eq. (8) proposed in [8], we introduce $\tilde{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}-\boldsymbol{\mu})$ to express Eq. (16) in the form of Eq. (13)

$$\tilde{\mathcal{F}}(\tilde{\boldsymbol{\alpha}}) = \frac{1}{2}||\tilde{\boldsymbol{\alpha}}||^2_2 + C\cdot\mathcal{F}_{\mathtt{lm}}(\tilde{\boldsymbol{\alpha}}+\boldsymbol{\mu}) \tag{17}$$

Therefore the training criterion of structured SVMs (Eq. 8) can be generalized as *minimising*

$$\begin{aligned}\frac{1}{2}||\tilde{\boldsymbol{\alpha}}||^2_2 + \frac{C}{R}\sum_{r=1}^{R}\Bigg[&-\max_{\boldsymbol{\theta}^{(r)}}\left((\tilde{\boldsymbol{\alpha}}+\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}^{(r)}_{\mathtt{ref}};\boldsymbol{\theta}^{(r)})\right)\\ &+\max_{\mathbf{w}\neq\mathbf{w}^{(r)}_{\mathtt{ref}},\boldsymbol{\theta}}\left\{\mathcal{L}(\mathbf{w},\mathbf{w}^{(r)}_{\mathtt{ref}}) + (\tilde{\boldsymbol{\alpha}}+\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w};\boldsymbol{\theta})\right\}\Bigg]_{+}\end{aligned} \tag{18}$$

Note that once the optimal reference alignment $\hat{\boldsymbol{\theta}}^{(r)}$ is given, then Eq. (18) can be reformulated in the form of Eq. (8) (see Algorithm 1 for more detail), where the loss is now become a score-augmented loss function

$$\tilde{\mathcal{L}}(\mathbf{w},\mathbf{w}^{(r)}_{\mathtt{ref}}) = \underbrace{\boldsymbol{\mu}^{\mathsf{T}}\Delta\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}^{(r)}_{\mathtt{ref}},\mathbf{w})}_{\text{acoustic and language loss}} + \underbrace{\mathcal{L}(\mathbf{w},\mathbf{w}^{(r)}_{\mathtt{ref}})}_{\text{transcription loss}} \tag{19}$$

[1] acoustic deweighting.

Algorithm 1: Structured SVM learning algorithm for ASR.

0. Initial: $\tilde{\boldsymbol{\alpha}} = [0, 0, 0 \dots]$, $\boldsymbol{\mu} = [1, 0, 0 \dots]$;

1. Fixing $\tilde{\boldsymbol{\alpha}}$, searching the optimal reference alignment $\boldsymbol{\theta}^{(r)}$ for each training pair $(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}})$ in numerator lattices using forward-backward algorithm:

$$\hat{\boldsymbol{\theta}}^{(r)} = \arg\max_{\boldsymbol{\theta}^{(r)}} \left((\tilde{\boldsymbol{\alpha}} + \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}; \boldsymbol{\theta}^{(r)}) \right), \ \forall \ r \tag{21}$$

2. Fixing $\hat{\boldsymbol{\theta}}^{(r)}$, optimise $\tilde{\boldsymbol{\alpha}}$ by *minimizing* the following convex upper bound using cutting plane algorithm in Algorithm 2:

$$\frac{1}{2}||\tilde{\boldsymbol{\alpha}}||_2^2 + \frac{C}{R}\sum_{r=1}^{R} \Big[\overbrace{-\tilde{\boldsymbol{\alpha}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}; \hat{\boldsymbol{\theta}}^{(r)})}^{\text{linear}} \tag{22}$$
$$+ \max_{\mathbf{w} \neq \mathbf{w}^{(r)}_{\mathtt{ref}}, \boldsymbol{\theta}} \left\{ \tilde{\mathcal{L}}(\mathbf{w}, \mathbf{w}^{(r)}_{\mathtt{ref}}) + \tilde{\boldsymbol{\alpha}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta}) \right\} \Big]_+$$

where $\tilde{\mathcal{L}}(\mathbf{w}, \mathbf{w}^{(r)}_{\mathtt{ref}}) = \boldsymbol{\mu}^{\mathsf{T}}\Delta\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}, \mathbf{w}) + \mathcal{L}(\mathbf{w}, \mathbf{w}^{(r)}_{\mathtt{ref}})$

3. go back to Step 1 until converge;
4. return $\tilde{\boldsymbol{\alpha}}$;

Algorithm 2: 1-slack Cutting plane algorithm [9] for Eq. (22).

```
Input: ${(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}; \hat{\boldsymbol{\theta}}^{(r)})}_{r=1}^{R}$ , $C$ and precision $\varepsilon$;
Initial empty constraint set: $\mathscr{W} \leftarrow \emptyset$;
repeat
    /* solving the 1-slack QP using current pool */
        $(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\xi}) \leftarrow \arg\min_{\boldsymbol{\alpha}, \boldsymbol{\xi} \geq 0} \frac{1}{2}||\tilde{\boldsymbol{\alpha}}||_2^2 + \frac{C}{R}\xi$        (23)
            S.T. $\forall \mathscr{W} : \tilde{\boldsymbol{\alpha}}^{\mathsf{T}} \sum_{r=1}^{R} \Delta\boldsymbol{\phi} + \sum_{r=1}^{R} \tilde{\mathcal{L}}(\mathbf{w}^{(r)}_{*}, \mathbf{w}^{(r)}_{\mathtt{ref}}) \leq \xi$
        where $\Delta\boldsymbol{\phi} = \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{*}; \boldsymbol{\theta}^{(r)}_{*}) - \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}; \hat{\boldsymbol{\theta}}^{(r)})$.
    for $r = 1..R$ do  /* Generating most competing hypothesis: */
        $\mathbf{w}^{(r)}_{*}, \boldsymbol{\theta}^{(r)}_{*} \leftarrow \arg\max_{\mathbf{w}, \boldsymbol{\theta}} \{ \tilde{\mathcal{L}}(\mathbf{w}, \mathbf{w}^{(r)}_{\mathtt{ref}}) + \tilde{\boldsymbol{\alpha}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta}) \}$  (24)
    end
    $\mathscr{W} \leftarrow \mathscr{W} \cup {\mathbf{w}^{(r)}_{*}, \boldsymbol{\theta}^{(r)}_{*}}_{r=1}^{R}$;    /* put it in the pool */
until  /* no constraint can be found that is violated by more than $\varepsilon$ */
        $\tilde{\boldsymbol{\alpha}}^{\mathsf{T}} \sum_{r=1}^{R} \Delta\boldsymbol{\phi} + \sum_{r=1}^{R} \tilde{\mathcal{L}}(\mathbf{w}^{(r)}_{*}, \mathbf{w}^{(r)}_{\mathtt{ref}}) \leq \xi + \varepsilon$ ;
return $\tilde{\alpha}$
```

where $\Delta\boldsymbol{\phi} = \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta}) - \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}; \hat{\boldsymbol{\theta}}^{(r)})$, $\boldsymbol{\mu}^{\mathsf{T}}\Delta\boldsymbol{\phi}$ can be viewed as an acoustic and language score loss. The decoding of structured SVMs based on $\tilde{\boldsymbol{\alpha}}$ can be written as

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}, \boldsymbol{\theta}} \left((\tilde{\boldsymbol{\alpha}} + \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta}) \right). \tag{20}$$

One interesting property of expression (18) is that even if $\tilde{\boldsymbol{\alpha}}$ is not well trained, e.g., in the earlier training stage, with a proper $\boldsymbol{\mu}$ the algorithm can still generate sensible competing hypothesis and reference alignments using the max terms in Eq. (18). This is particularly helpful to reduce the convergence time in medium/large vocabulary ASR.

To solve the non-convex optimisation problem with respective to $\tilde{\boldsymbol{\alpha}}$ in Eq. (18), an algorithm based on concave-convex procedure [10] is proposed in Algorithm 1. It works similar to the iteration process of EM. First, we find the "most likely" segment $\hat{\boldsymbol{\theta}}$ for current parameter $\tilde{\boldsymbol{\alpha}}$. This correspond to find the linear upper bound of the concave term of Eq. (18). Second, with the current segment $\hat{\boldsymbol{\theta}}$, the resulted convex optimization can be solved using 1-slack cutting plane algorithm [9] described in Algorithm 2. These two steps will go several iterations. The detail is shown in Algorithm 1.

Note that the objective function in Eq. (22) is convex for $\boldsymbol{\alpha}$, however, solving this problem is not trivial. Because the number of constraints is exponentially large, although the number of valid constraints that actually affect the solution is limited. One existing algorithm for this type of problem is the 1-slack cutting plane algorithm summarized in Algorithm 2, where the quadratic programming (23) only has 1-slack variable. The algorithm iteratively construct a working set $\mathscr{W}$ of constraints. In each iteration, it computes the solution over the current $\mathscr{W}$ (Eq. (23)), finds the most violated constraint (Eq. (24)), and adds it to the working set. The 1-slack algorithm stops once no constraint can be found that is violated by more than the desired precision ε.

IV. Implementation Issues

An efficient implementation of the algorithm is important for medium to large vocabulary speech recognition. In the following we summarized several design decisions that have a substantial influence on practical efficiency.

A. *1-slack optimisation*

There are two form of cutting plane algorithms [9], n-slack and 1-slack algorithms. An advantage of the 1-slack algorithm is the number of constraints and support vectors it produces is much smaller than n-slack case. In theory, the n-slack algorithm may add R constraints in every iteration, where R the size of training set. The 1-slack algorithm only adds a single constraint per iteration at most. In practice, for Aurora 4 experiments, 1-slack algorithms produce less then 500 active constraints at the solutions, whereas n-slack algorithms produce more than $50,000$ constraints after 20 iterations which make it impractical for medium/large vocabulary ASR. For AURORA 2 small vocabulary task, both algorithms can be applied. 1-slack algorithm only produce 24 support vectors whereas the number in n-slack case is 629. This means that in the 1-slack algorithm the QP problem (Eq. 23) on current working sets that need to be solved in each iteration are much smaller.

Another interesting property about 1-slack algorithm (Algorithm 2) is that constraints depend on $\sum_{r=1}^{R} \Delta\boldsymbol{\phi}$ instead of individual $\Delta\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}, \mathbf{w}^{(r)}_{*})$. Therefore, some competing hypothesis $\mathbf{w}^{(r)}_{*}$ could be involved in the constraints many times. To avoid the time wasted on repeatedly searching the same $\mathbf{w}^{(r)}_{*}$, the 10 most recently used $\Delta\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}, \mathbf{w}^{(r)}_{*})$ for each training observation $\mathbf{O}^{(r)}$ are cached.

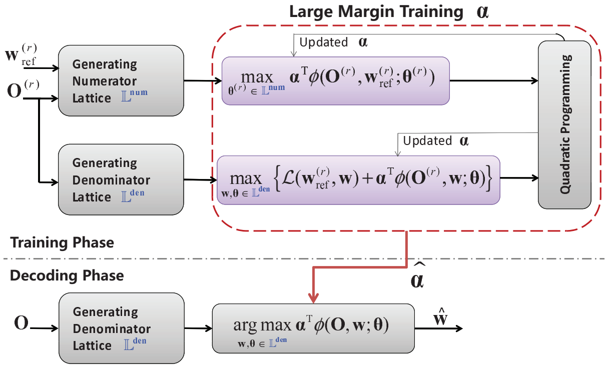
Large Margin Training $\boldsymbol{\alpha}$
$\mathbf{w}_{\text{ref}}^{(r)}$
$\mathbf{O}^{(r)}$
Generating Numerator Lattice $\mathbb{L}^{\text{num}}$
Generating Denominator Lattice $\mathbb{L}^{\text{den}}$
Updated $\boldsymbol{\alpha}$
$\max_{\boldsymbol{\theta}^{(r)} \in \mathbb{L}^{\text{num}}} \boldsymbol{\alpha}^{\mathsf{T}}\phi(\mathbf{O}^{(r)}, \mathbf{w}_{\text{ref}}^{(r)}; \boldsymbol{\theta}^{(r)})$
Updated $\boldsymbol{\alpha}$
$\max_{\mathbf{w},\boldsymbol{\theta} \in \mathbb{L}^{\text{den}}} \left\{\mathcal{L}(\mathbf{w}_{\text{ref}}^{(r)}, \mathbf{w}) + \boldsymbol{\alpha}^{\mathsf{T}}\phi(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta})\right\}$
Quadratic Programming
Training Phase
Decoding Phase
$\hat{\boldsymbol{\alpha}}$
$\mathbf{O}$
Generating Denominator Lattice $\mathbb{L}^{\text{den}}$
$\arg\max_{\mathbf{w},\boldsymbol{\theta} \in \mathbb{L}^{\text{den}}} \boldsymbol{\alpha}^{\mathsf{T}}\phi(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta})$
$\hat{\mathbf{w}}$

Fig. 2. The diagram of training and decoding for structured SVMs.

For both the n-slack and the 1-slack algorithms, constraints that were added to the working set in earlier iterations often become inactive later. These constraints can be removed without affecting the theoretical convergence of the algorithm. This is practically useful since it leading to a relatively smaller QP to be solved in later iterations.

B. Efficient search

Theoretically, the large margin training criterion discussed in the previous section can be directly applied to the model training. In practice, to make the algorithms applicable to larger vocabulary ASR, there are two search sub-problems must be solved efficiently (see Fig. 2), the best reference alignment Eq. (21) in Algorithm 1 and the best competing hypothesis/alignment Eq. (24) in Algorithm 2.

In previous work on small vocabulary digit string recognition task, it is feasible to search all the possible alignments and competing hypothesis in those two subproblems by using a Viterbi-style search [8]. However, it is not practical for larger tasks because the request to handle the exponential large searching space for all possible $\mathbf{w}$ and $\boldsymbol{\theta}$. Similar to the discriminative training in [7], numerator and denominator lattices $\mathbb{L}^{\text{num}}$ and $\mathbb{L}^{\text{den}}$ are generated to restrict the large searching space. Then a lattice-based searching algorithm is proposed to find the best competing path (hypothesis) among the lattices. Note that $\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta})$ can be calculated arc by arc using Eq. (4) and a standard MPE approximate loss [7] can also be computed on the arc level. Thus the best competing path Eq. (24) can be efficiently achieved through an arc-level forward-backward searching over the lattice. Similarly, Eq. (21) can also be efficiently searched in the numerator lattice $\mathbb{L}^{\text{num}}$.

For large scale applications, the computational load during training is dominated by finding the best competing hypothesis/alignment. For the n-slack algorithm, in order to run it in parallel on many machines, the sequential update mode of the standard cutting plane algorithm needs to be modified to a batch-mode update. Note that for n-slack algorithm, this parallelization will decrease the performance slightly [8][2]. However for 1-slack algorithm used in this work can be easily parallelized without any degradation. In theory, one could make use of up to R parallel threads, each searching the best competing hypothesis for a subset of training data. Paralleling the loop for Eq. (24) will lead to a substantial speed-up in the number of threads.

V. Noise Robustness

In ASR, the acoustic conditions during training and testing are seldom matched. For standard generative models, model-based compensation schemes such as Vector Taylor Series (VTS) compensation [12] are a popular and successful approach to handling this problem. When applying the same concept to structured SVMs there are two options. First, the discriminative model parameters, $\boldsymbol{\alpha}^{\mathsf{T}} = [\boldsymbol{\alpha}^{(\tilde{w}_1)^{\mathsf{T}}}, \ldots, \boldsymbol{\alpha}^{(\tilde{w}_{\mathrm{M}})^{\mathsf{T}}}]$, can be modified to be noise dependent. However with very limited data in the target domain, in these experiments a single utterance, this is not possible.

Alternatively, the parameters $\boldsymbol{\lambda}$ associated with the joint feature space are modified [5]. This can be achieved using any model-based compensation scheme. In this work VTS is used. Considering just the static components of the acoustic models, the compensated mean vector and covariance matrix of component m of the generative model are given by

$$\boldsymbol{\mu}^{(m)} = \mathbf{C}\,\mathbf{log}\left(\mathbf{exp}(\mathbf{C}^{-1}(\boldsymbol{\mu}_{\mathrm{x}}^{(m)} + \boldsymbol{\mu}_{\mathrm{h}}) + \mathbf{exp}(\mathbf{C}^{-1}\boldsymbol{\mu}_{\mathrm{n}})\right)$$
$$\boldsymbol{\Sigma}^{(m)} = \mathbf{J}^{(m)}\boldsymbol{\Sigma}_{\mathrm{x}}^{(m)}\mathbf{J}^{(m)\mathsf{T}} + (\mathbf{I} - \mathbf{J}^{(m)})\boldsymbol{\Sigma}_{\mathrm{n}}(\mathbf{I} - \mathbf{J}^{(m)})^{\mathsf{T}}$$

where $\boldsymbol{\mu}_{\mathrm{x}}^{(m)}$ and $\boldsymbol{\Sigma}_{\mathrm{x}}^{(m)}$ are the "clean" speech component mean vector and covariance matrix, and $\boldsymbol{\mu}_{\mathrm{n}}$, $\boldsymbol{\Sigma}_{\mathrm{n}}$ and $\boldsymbol{\mu}_{\mathrm{h}}$ are the additive and convolutional noise parameters respectively. $\mathbf{C}$ is the DCT matrix and $\mathbf{J}^{(m)}$ is Jacobian matrix [12]. $\mathbf{exp}()$ and $\mathbf{log}()$ are element-wise exponential and logarithm respectively. The noise model parameters are estimated using maximum likelihood estimation [13]. Thus in this work structured SVM parameters are assumed to be noise-independent, whereas the generative model parameters are noise-dependent.

VI. Experiments

This section describes experiments with the structured SVMs in AURORA 2 and 4. The AURORA 2 results are included to contrast the performance of 1-slack with n-slack algorithms and the gains from modeling the prior. The AURORA 4 results are used to illustrate the performance of proposed structured SVMs algorithms for noise robust medium vocabulary speech recognition.

AURORA 2 is a standard small vocabulary digital recognition task. The vocabulary size M is only 12 (one to nine, plus zero, oh and silence). The utterances are one to seven digits long based on the TIDIGITS database with noise artificially added. The generative models (HMMs), are 16 emitting states whole word digit models, with 3 mixtures per state. There are three test sets each includes $0-20$dB five SNRs. Set A was used as the development set for tuning parameters for all systems, such as the penalty factor C in structured SVMs. The joint feature space is based on appended-all features in

[2] Because in the sequential mode n-slack algorithm, $\boldsymbol{\alpha}$ can be updated after every training sample. This allows the algorithm to potentially find better competing $\mathbf{w}$ for the subsequence samples, but it can not be parallelized.

TABLE I
AURORA 2 RECOGNITION RESULTS (WER %) OF VTS BASED HMM, LOG LINEAR MODEL (LLM) [3] AND STRUCTURED SVMS (SSVM) USING 1-SLACK, SEQUENTIAL-MODE n-SLACK ALGORITHMS AND GENERALIZED LARGE MARGIN TRAINING WITH GAUSSIAN PRIOR (ALG. 1).

Model	**Set A**	**Set B**	**Set C**	**Avg.**
HMM	9.8	9.1	9.5	9.5
LLM (CML)	8.1	7.7	8.3	8.1
SSVM (n-slack)	7.6	7.2	8.0	7.5
SSVM (1-slack)	7.6	7.3	7.9	7.5
SSVM ($\boldsymbol{\mu}$,1-slack)	7.5	7.1	7.9	7.4

Eq. 3, no language model is used. The performances of VTS-compensated HMM, log linear models proposed in [3] and the structured SVMs with different training algorithms and criteria are shown in Table I. Examining the results in this table, shows the benefit of using 1-slack algorithm in structured SVM where the WER are almost the same with n-slack algorithm but with much fewer support vectors (24 compared with 169) and less computation. Small but consistent gains are observed when training structured SVMs with general Gaussian prior using 1-slack algorithm (last two lines in the table). The mean of Gaussian prior is set as the $\boldsymbol{\alpha}$ learned using 1-slack algorithm (the second last line in the table).

AURORA 4 is a noise-corrupted medium to large vocabulary task based on the Wall Street Journal (WSJ) data. Our configuration repeats the previous setup where the HMM is trained from clean data (SI-84 WSJ0 part, 14 hours). The HMMs are state-clustered triphones (3140 states) with 16 components/mixture. Four iterations of VTS compensation are performed for the test data. To compare with the log linear model proposed in [3], the joint feature space of structured SVMs follows the same setup described in [3] with 47×47 dimensions. The log linear models and structured SVMs are trained on the multi-style data. Evaluation is performed using the standard 5000- word WSJ0 bigram model on four noise-corrupted test sets3 based on NIST Nov92 WSJ0 test set. Table II shows the AURORA4 results of structured SVMs trained using large margin criteria with a general Gaussian prior. The mean of prior is set as the parameters of CML trained log linear model. The results from last two lines shown that optimising the segmentation in the numerator lattice yields small gain in performance. Compared to the CML trained log linear models with same dimension features, proposed structured SVMs yielded a 3.4% relative reduction in WER. Note that due to too many constraints, n-slack algorithm can not be applied in AURORA 4. For this larger task, it is also impractical to train structured SVMs without considering a proper prior, since the standard large margin training with zero mean of prior convergences too slow.

VII. CONCLUSION

This paper described a structured SVM framework suitable for noise-robust medium/large vocabulary speech recognition. Several theoretical and practical extensions to previous work in small vocabulary task have been made. First, the joint feature space based on word models is extended to allow context-dependent triphone models to be used. Second, by interpreting the structured SVM as a large margin log-linear model, illustrates that there is an implicit assumption that the prior of the discriminative parameter is a zero mean Gaussian. However, depending on the definition of likelihood feature space, a non-zero prior may be more appropriate. The assumption is relaxed by incorporating a more general Gaussian prior into the large margin training criterion in a form that allows the cutting plan algorithm to be directly applied. To speed up the training process, strategies such as 1-slack algorithm, caching competing hypothesis and parallelization are proposed. The performance of structured SVMs is evaluated on AURORA 4. Gains are observed over both VTS-compensated HMM and log linear models. Kernelization of this structured SVM to support high dimensional feature spaces such as derivative feature space will be investigated in the future.

TABLE II
AURORA 4 RECOGNITION RESULTS. FOR SSVM, $\boldsymbol{\mu}$, MEANS LARGE MARGIN TRAINING $\boldsymbol{\alpha}$ WITH GAUSSIAN PRIOR (ALG. 1), $\hat{\boldsymbol{\theta}}$ MEANS UPDATING THE SEGMENTATION IN THE NUMERATOR LATTICE.

Model	Criterion	Test Set WER (%)				Avg
		A	B	C	D	
HMM	ML	7.1	15.3	12.2	23.1	17.8
LLM	CML	7.2	14.7	11.1	22.8	17.4
	MPE	7.3	14.7	11.2	22.7	17.4
SSVM	LM ($\boldsymbol{\mu}$, 1-slack)	7.5	14.3	11.4	21.9	16.9
	LM ($\boldsymbol{\mu}, \hat{\boldsymbol{\theta}}$, 1-slack)	7.4	14.2	11.3	21.9	16.8

ACKNOWLEDGMENT

The authors would like to thank Toshiba Research Europe Ltd, Cambridge Research Lab, for partly funding this work.

REFERENCES

[1] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *ASRU*, 2009.

[2] M. Layton and M. Gales, "Augmented statistical models for speech recognition," in *Proc. ICASSP*, Toulouse, 2006.

[3] A. Ragni and M. J. F. Gales, "Structured discriminative models for noise robust continuous speech recognition," in *Proc. ICASSP*, Prague, Czech Repubic, 2011.

[4] 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, pp. 945–948, 2010.

[5] M. J. F. Gales and F. Flego, "Discriminative classifiers with adaptive kernels for noise robust speech recognition," *Comput. Speech Lang.*, vol. 24, no. 4, pp. 648–662, 2010.

[6] B. Taskar, "Learning structured prediction models: a large margin approach," Ph.D. dissertation, CA, USA, 2005.

[7] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University, 2004.

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

[9] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, 2009.

[10] A. Yuille, A. Rangarajan, and A. L. Yuille, "The concave-convex procedure (CCCP)," in *Advances in Neural Information Processing Systems*. MIT Press, 2002.

[11] F. Sha and L. K. Saul, "Large margin hidden Markov models for automatic speech recognition," in *NIPS*, 2007, pp. 1249–1256.

[12] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM Adaptation using Vector Taylor Series for Noisy Speech Recognition," in *Proc. ICSLP*, Beijing, China, 2000.

[13] H. Liao and M. Gales, "Joint uncertainty decoding for robust large vocabulary speech recognition," Cambridge University, Tech. Rep. CUED/F-INFENG/TR552, November 2006.