Structured SVMs for Automatic Speech Recognition

Shi-Xiong Zhang, *Student Member, IEEE* and M. J. F. Gales, *Fellow, IEEE*

***Abstract*—Combining generative and discriminative models offers a flexible sequence classification framework. This paper describes a structured support vector machines (SSVM) approach in this framework suitable for medium to large vocabulary speech recognition. One important aspect of SSVMs is the form of the *joint* feature space. In this work features based on context-dependent generative models are used. These features require a segmentation to be specified, a Viterbi-like scheme for obtaining the "optimal" segmentation is described. Large margin log linear models with a zero mean Gaussian prior of discriminative parameters is shown to be an example of this model. However, depending on the nature of the feature space, a non-zero prior may be more appropriate. An extended SSVM training algorithm is proposed to allow a general Gaussian prior to be incorporated into the large margin criterion. To speed up the training process, a 1-slack algorithm, caching competing hypotheses and parallelization strategies are also described. The performance of SSVMs is evaluated on small and medium to large speech recognition tasks: AURORA 2 and 4.**

I. Introduction

Continuous speech recognition (CSR) systems are typically trained using a large (compared to many machine learning tasks) amount of training data, millions of words of language model training data and millions of frames of acoustic model training data [2]. In addition, CSR is a *structured* classification problem [3], [4] in which class labels (sentences) have meaningful internal structure (e.g., words). Thus, although the number of possible class labels in this problem is unlimited, the labels are related, they all consist of a common set of basic structures, e.g., words and phones.

Most CSR systems use structured generative models, in the form of hidden Markov models (HMMs), as the acoustic models. HMMs for individual sub-sentence units can be simply combined together to form a model for a class label. Likelihoods from these HMMs are combined with the prior, usually an n-gram language model, to yield the sentence posterior based on Bayes' rule [5]. This enables posteriors of all possible sentences to be obtained. Although discriminative training [6]–[10] of HMMs has been shown to yield performance gains, the underlying acoustic models are still generative, with the standard HMM conditional independence assumptions, and the form of posteriors are found by Bayes' rule. This has led to interest in discriminative models, e.g., flat direct models (FDM) [11], segmental conditional random fields (SCRF) [12], conditional augmented models (CAug) [13] and log linear model (LLM) [14], [15], where the sentence posterior given the observation is modelled *directly*.

For discriminative models three important decisions need to be made: the form of the features to use; the appropriate training criterion; and how to handle the structure in continuous speech. A number of features have been investigated at the frame, model and word level [12], [14]. 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 [16]. Discriminative models are often trained using the conditional maximum likelihood (CML) [12], [13] criterion. However for high-dimensional features, there may be issues with generalisation. Additionally the training criterion is not linked with the evaluation criterion. To address this there has been interest in large margin [14], [17], [18] and minimum Bayes' risk [19] criteria for discriminative models. Depending on whether the structure in sentence level labels is explicitly modelled discriminative models will be divided into *unstructured*, and *structured* approaches in this work. Several commonly used unstructured and structured models are summarized in Table I.

TABLE I
SUMMARY OF UNSTRUCTURED AND STRUCTURED MODELS. M^3N IS SHORT FOR MAX-MARGIN MARKOV NETWORK [18], WHICH IS AN INSTANTIATION OF STRUCTURED SVM FOR THE CASE WHERE THE STRUCTURE IS CAPTURED BY A MARKOV NETWORK [20].

Training	Unstructured Models → Structured Models
ML	Naive Bayesian Network → HMM [5]
CML (MMI)	Logistic Regression [21] → CRF [22]
	Flat Direct Model [11] → SCRF [23] or CAug [13]
Large Margin	(Multi-Class) SVM [24] → M^3N [18] or Structured SVM [20]

Unstructured models, e.g. logistic regression model and support vector machines (SVMs), assume class labels are independent and have no structure. When applying these models to complete utterance in CSR, the space of possible classes becomes very large, e.g., a 6-digit length utterance yields 10^6 classes. One solution to deal with this, similar to acoustic code-breaking [25], is to segment the continuous speech into words/sub-words observation sequences. For each segment, multi-class SVMs or logistic regression can be applied in the some fashion as an isolated classification tasks [14], [16], [21]. However, this approach has two problems. First, the classification is based on one, fixed, segmentation. Second, each segment is treated independently. Another solution is to incorporate the structure into the model. For logistic regressions, this structured extension leads to CRFs. For SVMs, this yields SSVMs [20].

This paper proposes a structured SVMs (SSVM) framework for medium to large vocabulary CSR. The features are derived from generative kernels, which provides an elegant way of combining generative and discriminative models. These generative model-level features usually depend on the segmentation of the observations [12], [14]. This segmentation is itself a function of the model. A Viterbi-like algorithm is described

Part of this work has been presented in ASRU (Hawaii, December 2011) [1].
S.-X. Zhang, and M.J.F. Gales are with the Cambridge University, Engineering Department, U.K. (email: {sxz20, mjfg}@eng.cam.ac.uk).

to obtain the *optimal segmentation* using the current discriminative model parameters. This paper also describes an efficient large margin training scheme based on lattices. Standard SSVMs are shown to be related to large margin log linear model with a zero mean Gaussian prior of the discriminative parameter. However, depending on the property of the feature space, a non-zero mean may be more appropriate. An approach to incorporate a more general Gaussian prior into SSVM training is detailed. An important feature is that this prior is used in a form that allows the cutting plane algorithm to be directly applied. Using an appropriate prior can reduce the convergence time in large scale application. Furthermore, in order to reduce the number of constraints during parameter optimisation on larger tasks, 1-slack cutting plane algorithm is used rather than the standard n-slack algorithm. To speed up the training process, caching and parallelization strategies are also proposed. Experimental results are presented on small and medium to large vocabulary CSR tasks: AURORA 2 and 4.

II. Structured Support Vector Machines

Denote $\mathbf{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ as an observation sequence and $\mathbf{w} = \{w_1, \ldots, w_{|\mathbf{w}|}\}$ as the corresponding label sequence. In SSVMs for CSR, the goal is to learn a weight vector $\boldsymbol{\alpha}$. A linear discriminant function $\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}, \mathbf{w})$ is then used to measure how well a label sequence $\mathbf{w}$ matches an observation sequence $\mathbf{O}$, such that

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

is the recognized label sequence for a given $\mathbf{O}$. Here $\boldsymbol{\alpha}$ is the discriminative parameter vector, $\boldsymbol{\phi}(\mathbf{O}, \mathbf{w})$ is a *joint* feature vector characterizing the statistical dependencies of the $(\mathbf{O}, \mathbf{w})$ pair. Unlike multi-class SVMs where a weight vector $\boldsymbol{\alpha}_{\mathbf{w}}$ is used for each class $\mathbf{w}$ to compute a score [24], SSVMs use a *joint* feature-space and a single weight vector, $\boldsymbol{\alpha}$. To apply SSVMs to LVCSR three important decisions need to be made: the form of the joint features $\boldsymbol{\phi}(\mathbf{O}, \mathbf{w})$ to use; efficient decoding algorithm based on the joint features; and the appropriate training criterion and efficient learning algorithm.

A. Joint Feature Space

This section describes the features that are used by SSVMs for medium and large CSR tasks. For general continuous speech recognition, given an observation sequence $\mathbf{O}$, the number of possible classes, i.e., hypothesized sentences $\mathbf{w}$ can be very large. To handle this problem, the labels are decomposed into shared structure units, phonetic units. Thus, an additional level of hidden variable $\boldsymbol{\theta}$ that represent this segmentation is introduced. In previous small vocabulary systems [14], [26], the observations were segmented at the word level, however medium to large vocabulary tasks data must be segmented at a sub-word level, such as phones to yield complete vocabulary coverage. Given an alignment $\boldsymbol{\theta}$ that splits the observation sequence into $|\mathbf{w}|$ segments $\mathbf{O} = \{\mathbf{O}_{1|\boldsymbol{\theta}}, \ldots, \mathbf{O}_{i|\boldsymbol{\theta}}, \ldots, \mathbf{O}_{|\mathbf{w}||\boldsymbol{\theta}}\}$, with corresponding labels $\mathbf{w} = \{w_1, \ldots, w_i, \ldots, w_{|\mathbf{w}|}\}$, where $\mathbf{O}_{i|\boldsymbol{\theta}}$ is the i-th segment associated to context-dependent phone label w_i. The resulting *joint* feature space can be defined as

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}^{(v_1)} \\ \vdots \\ \boldsymbol{\alpha}^{(v_{\mathrm{M}})} \\ \alpha^{\mathtt{lm}} \end{bmatrix}, \boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta}) \triangleq \begin{bmatrix} \sum_{i=1}^{|\mathbf{w}|} \delta(w_i - v_1) \boldsymbol{\psi}(\mathbf{O}_{i|\boldsymbol{\theta}}) \\ \vdots \\ \sum_{i=1}^{|\mathbf{w}|} \delta(w_i - v_{\mathrm{M}}) \boldsymbol{\psi}(\mathbf{O}_{i|\boldsymbol{\theta}}) \\ \log P(\mathbf{w}) \end{bmatrix} \tag{2}$$

Here $\{v_k\}_{k=1}^{M}$ denotes for example all possible triphones in the dictionary, $\delta(w_i - v_k)$ is the Kronecker delta function, and $\boldsymbol{\psi}(\mathbf{O}_{i|\boldsymbol{\theta}})$ is the feature vector extracted for segment $\mathbf{O}_{i|\boldsymbol{\theta}}$. $P(\mathbf{w})$ is the standard n-gram language model probability. Although more general language model features can be appended, they are not considered in this work. Fig. 1 shows an example of using equation (2) to construct a joint feature space for data pair $(\mathbf{O}, \mathbf{w})$ given a segmentation $\boldsymbol{\theta}$. Note that the position of $\boldsymbol{\psi}(\mathbf{O}_{i|\boldsymbol{\theta}})$ in the *joint* feature space depends on its label w_i.

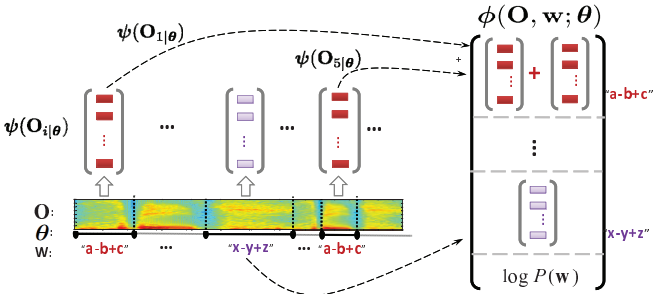

Fig. 1. Constructing the *joint* feature space from feature space.

To map each variable-length segment $\mathbf{O}_{i|\boldsymbol{\theta}}$ to a fixed dimensional vector $\boldsymbol{\psi}(\mathbf{O}_{i|\boldsymbol{\theta}})$, a feature spaces related to sequence kernels [27] can be used. Of particular interest in this work are those sequence kernels based on generative models. As well as yielding a mapping to a fixed vector size, these generative kernels allow standard speaker and noise adaptation approaches developed for CSR to be used to derive robust features [16]. There are a number of possible generative kernel-induced feature spaces $\boldsymbol{\psi}_{\boldsymbol{\lambda}}(\mathbf{O}_{i|\boldsymbol{\theta}})$ that can be used to form the *joint* feature space $\boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta})$. The simplest example is the log-likelihood feature space

$$\boldsymbol{\psi}_{\boldsymbol{\lambda}}(\mathbf{O}) = \begin{bmatrix} \log p_{\boldsymbol{\lambda}}(\mathbf{O}|v_1) \\ \vdots \\ \log p_{\boldsymbol{\lambda}}(\mathbf{O}|v_{\mathrm{M}}) \end{bmatrix} \tag{3}$$

where $\boldsymbol{\lambda}$ denotes the generative model parameters, and $p_{\boldsymbol{\lambda}}(\mathbf{O}|v_k)$ is the likelihood for generative model v_k. This feature space concatenates the log-likelihoods from all models, including the correct model and competing ones, to yield additional information from the observations. More general feature-spaces, such as derivative ones [28], can relax the conditional independence assumption.

Using the above joint feature-spaces 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 [14]

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

where $\boldsymbol{\alpha}^{\mathsf{T}} = [{\boldsymbol{\alpha}^{(v_1)}}^{\mathsf{T}}, \ldots {\boldsymbol{\alpha}^{(v_k)}}^{\mathsf{T}} \ldots, {\boldsymbol{\alpha}^{(v_{\mathrm{M}})}}^{\mathsf{T}}, \alpha^{\mathtt{lm}}]$ and $w_i \in \{v_k\}_{k=1}^{M}$. One elegant property of this joint feature space is

that it allows the standard HMM baseline to be obtained by simply setting the value of $\boldsymbol{\alpha}$ associated with the correct model to be one and zero for all competing models (see (3)), i.e., the sparse vectors $\boldsymbol{\alpha}^{(v_1)} = [1\ 0 \ldots 0]^{\mathsf{T}}, \cdots, \boldsymbol{\alpha}^{(v_M)} = [0\ 0 \ldots 1]^{\mathsf{T}}$ and $\alpha^{\mathtt{lm}}$ is the language model scaling factor.

When medium to large vocabulary CSR are considered there is an issue with directly using this feature space with context dependent phones. The set of all possible models $\{v_k\}_{k=1}^{M}$ yields a very large *joint* feature space. Although in theory this could be used, the number of discriminative model parameters becomes large. Two approaches were proposed in [29] to address this problem, and are adopted in this work. One is to reduce the dimension of the feature space $\boldsymbol{\psi}(\cdot)$ by selecting a small set of "suitable" models. For example, instead of using the full feature space, the *matched-context* feature space of a segment $\mathbf{O}$ with the label $a{-}b{+}c$ can be used here as

$$\boldsymbol{\psi}_{\boldsymbol{\lambda}}(\mathbf{O}) = \begin{bmatrix} \log p_{\boldsymbol{\lambda}}(\mathbf{O}|a{-}a{+}c) \\ \vdots \\ \log p_{\boldsymbol{\lambda}}(\mathbf{O}|a{-}y{+}c) \\ \log p_{\boldsymbol{\lambda}}(\mathbf{O}|a{-}z{+}c) \end{bmatrix}_{M_1} . \tag{5}$$

This reduces the dimensionality of the feature space $\boldsymbol{\psi}_{\boldsymbol{\lambda}}(\cdot)$ from the number of context-dependent phones M to the number of monophones M_1. The second approach is to reduce the dimension by tying the discriminative model parameter $\boldsymbol{\alpha}$ using a phonetic decision tree [29]. For example, if v_i and v_j belong to the same leaf node in a decision tree, then $\boldsymbol{\alpha}^{(v_i)}$ and $\boldsymbol{\alpha}^{(v_j)}$ are tied. Thus the dimensionality of joint feature space reduced to $M_2 \times M_1 + 1$, where M_2 is the number of leaf nodes in decision tree. This is illustrated in Fig. 2. In this work, M_1 and M_2 are both set to 47, the number of monophones.

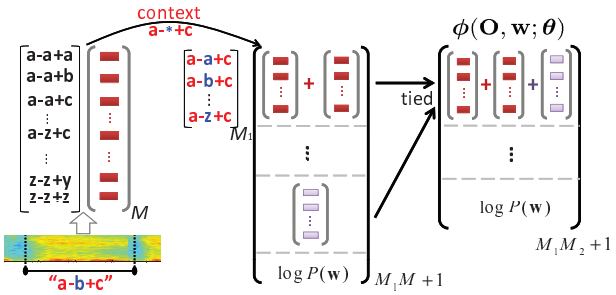

Fig. 2. Selecting matched context and discriminative parameter tying.

B. Inference with optimal segmentation

The joint feature space described above is based on a specific segmentation, $\boldsymbol{\theta}$. In SCRF [12], CAug [13] and LLM [14] based systems, the segmentations, $\boldsymbol{\theta}_{\boldsymbol{\lambda}}$, are typically generated using standard generative model HMMs. These segmentations are fixed for both decoding and training. For inference this yields

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

$$\text{where } \boldsymbol{\theta}_{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\theta}} \ P(\boldsymbol{\theta}|\mathbf{w}) p_{\boldsymbol{\lambda}}(\mathbf{O}|\boldsymbol{\theta}, \mathbf{w}). \tag{7}$$

Equation (7) can be solved when HMMs are used using the Viterbi algorithm. Although $\boldsymbol{\theta}_{\boldsymbol{\lambda}}$ is the optimal segmentation for the *generative* model, it may not be the best segmentation to characterize the dependencies on $(\mathbf{O}, \mathbf{w})$ pair for the *discriminative* model. There is thus a potential mismatch between (6) and (7).

The segmentation variable $\boldsymbol{\theta}$ should be optimised based on the discriminative models. For general feature-spaces it is not possible to define efficient algorithms to achieve this. However for the log-likelihood feature-spaces (3) it is possible. The decoding formula (6) now becomes

$$\left\{\mathbf{w}_{\boldsymbol{\alpha}}, \boldsymbol{\theta}_{\boldsymbol{\alpha}}\right\} = \arg\max_{\mathbf{w}, \boldsymbol{\theta}} \left\{\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \boldsymbol{\theta})\right\}. \tag{8}$$

Given the joint feature-space (2) a Viterbi-style algorithm to solve equation (8) can be found. Based on equation (4) it is possible to express the maximisation in equation (8) as

$$\left\{\mathbf{w}_{\boldsymbol{\alpha}}, \boldsymbol{\theta}_{\boldsymbol{\alpha}}\right\} = \tag{9}$$
$$\arg\max_{\mathbf{w}} \left\{ \max_{\boldsymbol{\theta}} \sum_{i=1}^{|\mathbf{w}|} \sum_{k=1}^{M} \alpha_k^{(w_i)} \log p_{\boldsymbol{\lambda}}(\mathbf{O}_{i|\boldsymbol{\theta}}|v_k) + \alpha^{\mathtt{lm}} \log P(\mathbf{w}) \right\}$$

For the form of feature-space in (3) this expression is related to factorial HMM inference [30]. The search process (9) can be split into two distinct terms. First given a segment the score for each model, the log-likelihood and the score for the language model need to be computed. This is the standard forward-backward algorithm for HMMs. The second is obtaining the optimal segmentation which requires a modified two-stage Viterbi search. This process is illustrated in Fig. 3. The M phone HMMs are shown in parallel with *synchronisation points* shown in black which are determined by the segment boundaries.

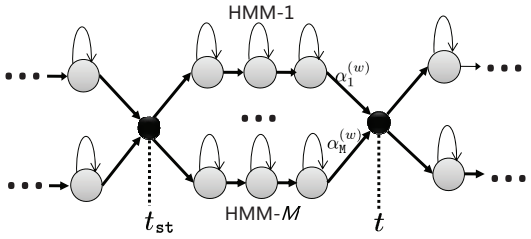

Fig. 3. Decoding procedure illustration. The black circles indicate the synchronisation points where the M HMM log likelihoods and language model score are merged.

This search process is similar to a semi-Markov search process [31]. The best score (and alignment history) for a label sequence $\mathbf{w}$ ending with w' at time $t_{\mathtt{st}}$ is stored as $\rho_{t_{\mathtt{st}}}^{(w')}$,

$$\rho_{t_{\mathtt{st}}}^{(w')} = \max_{\mathbf{w}, \boldsymbol{\theta}} \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}_{1:t_{\mathtt{st}}}, \mathbf{w}; \boldsymbol{\theta}). \tag{10}$$

Given this time $t_{\mathtt{st}}$ (the start time of current decoding segment), the forward score up-to time t for each model v_k, is computed at the end state of that model, $\log p_{\boldsymbol{\lambda}}(\mathbf{O}_{t_{\mathtt{st}}:t}|v_k)$. The best score for a label sequence ending with w at time t can then be expressed as

$$\rho_t^{(w)} = \max_{t_{\mathtt{st}}, w'} \left\{ \rho_{t_{\mathtt{st}}}^{(w')} + s_{(t_{\mathtt{st}}, w') \rightarrow (t, w)} \right\}, \tag{11}$$

where $s_{(t_{\mathtt{st}}, w') \rightarrow (t, w)} = \sum_{k=1}^{M} \alpha_k^{(w)} \log p_{\boldsymbol{\lambda}}(\mathbf{O}_{t_{\mathtt{st}}:t}|v_k) + \alpha^{\mathtt{lm}} \log P(w|w')$ is the score for segment $(t_{\mathtt{st}}, w') : (t, w)$ which can be computed using the forward-backward algorithm

and $t_{\mathtt{st}} \in [0, t-1]$. By running the above Viterbi-style search from time 1 to T the optimal sentence and segmentation can be obtained by tracing back the model and time index maximising $\rho_T^{(w_{|\mathbf{w}|})}$.

The complexity of the above process is $\mathcal{O}(MT^2)$. Pruning options are available, e.g., limiting w' in (11) to the top N models with highest scores and constraint the look-back time $t_{\mathtt{st}}$ to last $\mathcal{T}$ frames. Additionally, more efficient approximations, e.g., Gibbs sampling and variational methods [30], could be used to reduce the computation load. However these are not investigated in this work.

C. Large Margin Training

The previous section has shown that given $\boldsymbol{\alpha}$ the optimal alignment $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ can be inferred. However, during training both $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ are unknown and depend on one another. The optimal alignment may vary with $\boldsymbol{\alpha}$; and adjusting the alignments will affect the optimal value of $\boldsymbol{\alpha}$. In this section, the joint training of the structured SVM and the optimal alignment is described with a large margin criterion.

The training data consists of pairs $\left\{\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}\right\}_{r=1}^{R}$, where $\mathbf{O}^{(r)}$ is the r^{th} observation sequence and $\mathbf{w}_{\mathtt{ref}}^{(r)}$ is its reference label sequence. In a similar fashion to the latent SVM [32] and structured SVM [20], [33], the parameters and hidden variables can be jointly trained by solving the following optimisation problem:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \ \frac{1}{2}||\boldsymbol{\alpha}||^2 + \frac{C}{R}\sum_{r=1}^{R}\xi_r \tag{16}$$

s.t. For every utterance $r = 1, \ldots, R$,

For all competing hypothesis $\tilde{\mathbf{w}}^{(r)} \neq \mathbf{w}_{\mathtt{ref}}^{(r)}$:

$$\max_{\boldsymbol{\theta}^{(r)}}\left\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}; \boldsymbol{\theta}^{(r)})\right\} - \max_{\tilde{\boldsymbol{\theta}}^{(r)}}\left\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \tilde{\mathbf{w}}^{(r)}; \tilde{\boldsymbol{\theta}}^{(r)})\right\}$$
$$\geq \mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)}, \tilde{\mathbf{w}}^{(r)}) - \xi_r,$$

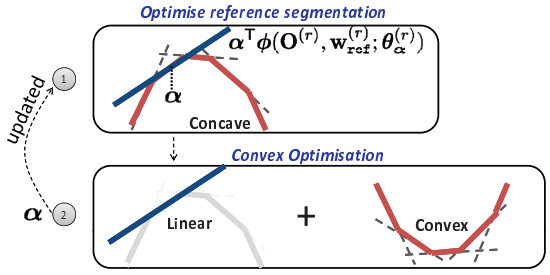

Fig. 4. The illustration of Step 1 and 2 in Algorithm 1 for joint learning the parameters $\boldsymbol{\alpha}$ of structured SVM and optimal segmentation $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ for CSR.

where $\xi_r \geq 0$ are the slack variables and $\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)}, \tilde{\mathbf{w}}^{(r)})$ is the loss function between the reference $\mathbf{w}_{\mathtt{ref}}^{(r)}$ and its competing hypothesis $\tilde{\mathbf{w}}^{(r)}$. The constraints in equation (16) can be explained as follows. For every training pair $(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)})$, the best score of the reference pair should be greater than all competing pairs $(\mathbf{O}^{(r)}, \tilde{\mathbf{w}}^{(r)})$ by a margin determined by the loss. However the number of possible competing hypotheses $\tilde{\mathbf{w}}^{(r)}$ is huge. Therefore, the challenge is to solve an optimisation problem with a large number of constraints, although the number of active constraints that affect the solution is limited.

Substituting the slack variable ξ_r from the constraints into the objective function, equation (16) can be reformulated as the *minimisation* of

$$\frac{1}{2}||\boldsymbol{\alpha}||_2^2 + \frac{C}{R}\sum_{r=1}^{R}\Big[\overbrace{-\max_{\boldsymbol{\theta}^{(r)}}\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}; \boldsymbol{\theta}^{(r)})\right)}^{\text{concave}} + \underbrace{\max_{\mathbf{w}\neq\mathbf{w}_{\mathtt{ref}}^{(r)},\boldsymbol{\theta}}\left\{\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)}, \mathbf{w}) + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta})\right\}}_{\text{convex}}\Big]_+ \tag{17}$$

where $[\,]_+$ is the hinge-loss function. Because of the $\max(\cdot)$, the objective function is non-differentiable and non-smooth. However, the maximum of a set of linear functions is convex. Therefore, the objective function in equation (17) comprises

Algorithm 1: Structured SVM learning algorithm for CSR.

0. Initial: $\boldsymbol{\alpha} = [1, 0, 0 \ldots]$ and $\boldsymbol{\theta}^{(r)} = \boldsymbol{\theta}_{\boldsymbol{\lambda}}^{(r)}$;

1. **Fixing** $\boldsymbol{\alpha}$, **optimise** the reference segmentation $\boldsymbol{\theta}^{(r)}$ for each training pair $(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)})$ using the Viterbi-style algorithm in Section II-B:

$$\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)} = \arg\max_{\boldsymbol{\theta}^{(r)}}\left\{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}; \boldsymbol{\theta}^{(r)})\right\}, \ \forall\ r \tag{12}$$

2. **Fixing** $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)}$, **optimise** $\boldsymbol{\alpha}$ by *minimizing* the following convex upper bound ((17) $\leq$ (13)), using the cutting plane algorithm in Algorithm 2:

$$\frac{1}{2}||\boldsymbol{\alpha}||_2^2 + \frac{C}{R}\sum_{r=1}^{R}\Big[\overbrace{-\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}; \boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)})}^{\text{linear}} + \max_{\mathbf{w}\neq\mathbf{w}_{\mathtt{ref}}^{(r)},\boldsymbol{\theta}}\left\{\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)}, \mathbf{w}) + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta})\right\}\Big]_+ \tag{13}$$

3. go back to Step 1 until converge;

return $\boldsymbol{\alpha}$;

Algorithm 2: 1-slack Cutting plane algorithm [20] for Eq. (13)

Input: $\{(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}; \boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)})\}_{r=1}^{R}$, C and precision ε;
Initial empty constraint set: $\mathscr{W} \leftarrow \emptyset$;
repeat
  / solving the 1-slack QP using current constraint set */*
  $$(\boldsymbol{\alpha}, \xi) \leftarrow \arg\min_{\boldsymbol{\alpha},\boldsymbol{\xi}\geq 0} \ \frac{1}{2}||\boldsymbol{\alpha}||_2^2 + \frac{C}{R}\xi \tag{14}$$
  $$\text{S.T. } \forall\ \mathscr{W} : \boldsymbol{\alpha}^{\mathsf{T}}\sum_{r=1}^{R}\Delta\boldsymbol{\phi}^{(r)} + \sum_{r=1}^{R}\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)}, \tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}) \leq \xi$$
  $$\text{where } \Delta\boldsymbol{\phi}^{(r)} = \boldsymbol{\phi}(\mathbf{O}^{(r)}, \tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}; \tilde{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}^{(r)}) - \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\mathtt{ref}}^{(r)}; \boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)})$$
  for $r = 1..R$ **do** */* generating best competing hypothesis: */*
    $$\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}, \tilde{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}^{(r)} \leftarrow \arg\max_{\mathbf{w},\boldsymbol{\theta}}\left\{\mathcal{L}(\mathbf{w}, \mathbf{w}_{\mathtt{ref}}^{(r)}) + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}; \boldsymbol{\theta})\right\} \tag{15}$$
  end
  $\mathscr{W} \leftarrow \mathscr{W} \cup \{\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}, \tilde{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}^{(r)}\}_{r=1}^{R}$; */* put it in constraint set */*
until */* no constraint can be found that is violated by more than ε */*
$$\boldsymbol{\alpha}^{\mathsf{T}}\sum_{r=1}^{R}\Delta\boldsymbol{\phi}^{(r)} + \sum_{r=1}^{R}\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)}, \tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}) \leq \xi + \varepsilon \ ;$$
return $\boldsymbol{\alpha}$

concave and convex parts. To solve this non-convex optimisation problem with respective to $\boldsymbol{\alpha}$ in equation (17), an algorithm based on the concave-convex procedure [32], [34] is proposed in Algorithm 1. It works similar to the iterative process of EM. First, the optimal reference segmentation $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)}$ for the current parameter $\boldsymbol{\alpha}$ are found in Step 1. This corresponds to finding the linear upper bound of the concave term of equation (25) as shown in Fig. 4. Second, with the current reference segmentation $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)}$, the optimal value of $\boldsymbol{\alpha}$ based on (13) is found. These two steps can then be repeated.

The objective function in equation (13) of Algorithm 1 is convex in $\boldsymbol{\alpha}$, however, solving this problem is not trivial because the criterion is non-differentiable. Existing algorithms for this problem fall into two groups. The first group of algorithms use generalized gradient-based approaches: subgradient methods [35] and exponentiated gradient methods [36]. The second group uses the cutting plane algorithm which does not take a single gradient step, but always takes an optimal step in the current constraint set [20], [37]. In this work, the 1-slack cutting plane algorithm summarized in Algorithm 2 is used to solve this convex optimization problem. Here the quadratic programming optimisation (14) only has 1-slack variable (see section IV-A for details). The algorithm iteratively constructs a working set $\mathscr{W}$ of constraints. In each iteration, it computes the solution over the current set $\mathscr{W}$ (14), finds the most violated constraint (15) ($\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}, \tilde{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}^{(r)}$ are the best competing hypothesis and its optimal segmentation), and adds it to the working set. This 1-slack algorithm stops when no constraint can be found that is violated by more than the desired precision ε.

D. Relationship with Log Linear Models

Just as SVMs can be interpreted as large margin logistic regressions [38], the SSVM can be viewed as large margin log linear models. To see this, the posterior of log linear model for hypothesized labels $\mathbf{w}$ given $\mathbf{O}$ can be written as,

$$P(\mathbf{w}|\mathbf{O};\boldsymbol{\theta}_{\boldsymbol{\alpha}},\boldsymbol{\alpha}) = \frac{\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta}_{\boldsymbol{\alpha}})\right)}{\sum_{\mathbf{w}'}\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w}';\boldsymbol{\theta}'_{\boldsymbol{\alpha}})\right)}, \tag{18}$$

where $\boldsymbol{\theta}_{\boldsymbol{\alpha}}$ is the best segmentation that maximises posterior probability $P(\mathbf{w}|\mathbf{O};\boldsymbol{\theta},\boldsymbol{\alpha})$, i.e., $\boldsymbol{\theta}_{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\theta}} \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta})$. Decoding with this log linear models can be simply expressed as

$$\begin{aligned}\mathbf{w}_{\boldsymbol{\alpha}} &= \arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{O};\boldsymbol{\theta}_{\boldsymbol{\alpha}},\boldsymbol{\alpha}) \\ &= \arg\max_{\mathbf{w}}\Big\{\max_{\boldsymbol{\theta}} \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta})\Big\}\end{aligned}$$

This yields both the optimal word sequence and alignment and is equivalent to structured SVM inference in equation (8).

In order to robustly train a model which has good generalization in a high-dimensional space with limited data, large margin based approaches can be applied [17], [39]. If the margin for the log linear models is defined as the log-posterior ratio of the reference $\mathbf{w}_{\mathtt{ref}}^{(r)}$ and best competing hypothesis $\mathbf{w}$, as illustrated in Fig. 5, large margin training for log linear model can be expressed as *minimising*

$$\mathcal{F}_{\mathtt{LLM}}(\boldsymbol{\alpha},\boldsymbol{\lambda}) = \frac{1}{R}\cdot\sum_{r=1}^{R} \tag{19}$$
$$\left[\max_{\mathbf{w}\neq\mathbf{w}_{\mathtt{ref}}^{(r)}}\left\{\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w}) - \log\left(\frac{P(\mathbf{w}_{\mathtt{ref}}^{(r)}|\mathbf{O}^{(r)};\boldsymbol{\theta}_{\boldsymbol{\alpha}},\boldsymbol{\alpha})}{P(\mathbf{w}|\mathbf{O}^{(r)};\boldsymbol{\theta}_{\boldsymbol{\alpha}},\boldsymbol{\alpha})}\right)\right\}\right]_+$$

where $\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w})$ is a loss function introduced to control the size of the margin and $[\ \cdot\ ]_+$ is the hinge-loss function. There are two sets of parameters for LLMs, discriminative parameters $\boldsymbol{\alpha}$ and generative model parameters $\boldsymbol{\lambda}$ to extract features. One general extension of this criterion is to incorporate priors $P(\boldsymbol{\alpha}), P(\boldsymbol{\lambda})$ and then *minimising*

$$\mathcal{F}(\boldsymbol{\alpha},\boldsymbol{\lambda}) = \mathcal{F}_{\mathtt{LLM}}(\boldsymbol{\alpha},\boldsymbol{\lambda}) - \log\left(P(\boldsymbol{\alpha})\right) - \log\left(P(\boldsymbol{\lambda})\right). \tag{20}$$

In this work the generative model parameters, $\boldsymbol{\lambda}$, are assumed to have been trained and fixed. (19) can then be expressed as

$$\begin{aligned}\mathcal{F}(\boldsymbol{\alpha}) = &-\log(P(\boldsymbol{\alpha})) + \frac{1}{R}\sum_{r=1}^{R}\Bigg[-\log P(\mathbf{w}_{\mathtt{ref}}^{(r)}|\mathbf{O}^{(r)};\boldsymbol{\theta}_{\boldsymbol{\alpha}},\boldsymbol{\alpha}) \\ &+ \max_{\mathbf{w}\neq\mathbf{w}_{\mathtt{ref}}^{(r)}}\Big\{\mathcal{L}(\mathbf{w},\mathbf{w}_{\mathtt{ref}}^{(r)}) + \log P(\mathbf{w}|\mathbf{O}^{(r)};\boldsymbol{\theta}_{\boldsymbol{\alpha}},\boldsymbol{\alpha})\Big\}\Bigg]_+\end{aligned} \tag{21}$$

The prior $P(\boldsymbol{\alpha})$ is assumed to be Gaussian with a zero mean and scaled identity covariance matrix $C\mathbf{I}$, thus $\log P(\boldsymbol{\alpha}) = \log\mathcal{N}(\mathbf{0},C\mathbf{I}) \propto -\frac{\mathbf{1}}{\mathbf{2C}}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\alpha}$. Substituting equation (18) into (21) with this prior assumption and *canceling out* the normalization terms in (18), yields the objective function (17). This suggests that the structured SVM used in this work can also be viewed as a large margin trained log linear model with an optimal segmentation.

Equation (21) is non-differentiable because of the $\max\{\cdot\}$ function. Thus gradient-based algorithm cannot be directly applied. One suitable algorithm to train the large margin LLM is described in Section II-C. It is also possible to apply a softmax approximation to this objective function, for example the large margin [9] and boosted MMI [40] training for HMMs. This yields a differentiable objective function. However this approximation minimises an upper bound of equations (21) rather than the criterion itself

$$\begin{aligned}\mathcal{F}(\boldsymbol{\alpha}) \leq &-\log(P(\boldsymbol{\alpha})) + \frac{1}{R}\sum_{r=1}^{R}\Bigg[-\log P(\mathbf{w}_{\mathtt{ref}}^{(r)}|\mathbf{O}^{(r)};\boldsymbol{\theta}_{\boldsymbol{\alpha}},\boldsymbol{\alpha}) \\ &+ \log\Big\{\sum_{\mathbf{w}}\exp\Big(\mathcal{L}(\mathbf{w},\mathbf{w}_{\mathtt{ref}}^{(r)})\Big)P(\mathbf{w}|\mathbf{O}^{(r)};\boldsymbol{\theta}_{\boldsymbol{\alpha}},\boldsymbol{\alpha})\Big\}\Bigg]_+\end{aligned} \tag{22}$$

Thus equation (22) allows gradient-based approach to be applied, but it is not the "real" maximum margin in (21).

The log linear model described above only uses the "most likely" segmentation. Alternatively, "all possible" segmentations could be considered. This turns the equation (18) to the SCRF, or CAug model [41],

$$P(\mathbf{w}|\mathbf{O};\boldsymbol{\alpha}) = \frac{\sum_{\boldsymbol{\theta}}\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta})\right)}{\sum_{\mathbf{w}'}\sum_{\boldsymbol{\theta}'}\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w}';\boldsymbol{\theta}')\right)}. \tag{23}$$

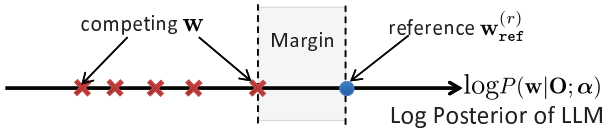

Fig. 5. The margin of log linear models is defined in log posterior domain between $\mathbf{w}_{\mathtt{ref}}$ and the best competing hypothesis $\mathbf{w}$.

Substituting (23) into the large margin criteria (21) will lead to a more complex concave-convex object, which can also be solved using the proposed Algorithms 1 and 2. This form of model is not investigated in this work.

III. Form of Prior

The previous section has shown that when training standard SSVMs, an implicit assumption is made that the prior distribution of $\boldsymbol{\alpha}$ is Gaussian, with zero mean and identity covariance matrix. However, for the feature space defined in equation (3), the prior mean, $\boldsymbol{\mu}$, should be non-zero. One appropriate form of prior mean is the one that yields the HMM baseline performance where

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

The value of $\boldsymbol{\mu}$ should thus be one for the correct class, zero otherwise, for example class v_1, $\boldsymbol{\mu}^{(v_1)} = [1,0,\ldots,0]^{\mathsf{T}}$. This motivates the need for a more general large margin training scheme that incorporate a general Gaussian prior $P(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\alpha};\boldsymbol{\mu},\boldsymbol{\Sigma})$ into SSVM training. Thus, the SSVMs training in equation (17) can be generalized to *minimise*

$$\frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu}) + \frac{C}{R}\sum_{r=1}^{R}\Big[-\max_{\boldsymbol{\theta}^{(r)}}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}_{\mathtt{ref}}^{(r)};\boldsymbol{\theta}^{(r)}) + \max_{\mathbf{w}\neq\mathbf{w}_{\mathtt{ref}},\boldsymbol{\theta}}\Big\{\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w}) + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w};\boldsymbol{\theta})\Big\}\Big]_{+} \quad (24)$$

This new expression is still concave-convex as long as the matrix $\boldsymbol{\Sigma}^{-1}$ is positive definite. Note the matrix $\boldsymbol{\Sigma}^{-1}$ can always be decomposed and merged into the feature space by using transformed features $\bar{\boldsymbol{\phi}}(\mathbf{O},\mathbf{w};\boldsymbol{\theta}) = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta})$. In this work, the log-likelihood features are assumed to be consistently scaled, so that $\boldsymbol{\Sigma} = C\boldsymbol{I}$ is a reasonable approximation. In order to utilize the training framework based on equation (17), it is necessary to transform the parameters $\bar{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}-\boldsymbol{\mu})$. Reformulate equation (24) in the form of (17)

$$\frac{1}{2}||\bar{\boldsymbol{\alpha}}||_2^2 + \frac{C}{R}\sum_{r=1}^{R}\Bigg[-\max_{\boldsymbol{\theta}^{(r)}}\Big\{(\bar{\boldsymbol{\alpha}}+\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}_{\mathtt{ref}}^{(r)};\boldsymbol{\theta}^{(r)})\Big\} + \max_{\mathbf{w}\neq\mathbf{w}_{\mathtt{ref}}^{(r)},\boldsymbol{\theta}}\Big\{\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w}) + (\bar{\boldsymbol{\alpha}}+\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w};\boldsymbol{\theta})\Big\}\Bigg]_{+} \quad (25)$$

Minimising equation (25) can be solved using Algorithm 3 and a modified version of Algorithm 1. Similar to Algorithm 1, once the optimal reference alignment $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)}$ is given, then equation (25) can be expressed as (29) which is exactly the same form of (13) with a new score-augmented loss function,

$$\bar{\mathcal{L}}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w}) = \underbrace{\boldsymbol{\mu}^{\mathsf{T}}\Delta\boldsymbol{\phi}^{(r)}}_{\text{score loss}} + \underbrace{\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w})}_{\text{transcription loss}} \quad (26)$$

$\boldsymbol{\mu}^{\mathsf{T}}\Delta\boldsymbol{\phi}^{(r)}$ can be viewed as an acoustic and language score loss, where $\Delta\boldsymbol{\phi}^{(r)} = \boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w};\boldsymbol{\theta}) - \boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}_{\mathtt{ref}}^{(r)};\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)})$. Inference with SSVMs based on $\bar{\boldsymbol{\alpha}}$ can be written as

$$\{\mathbf{w}_{\boldsymbol{\alpha}},\boldsymbol{\theta}_{\boldsymbol{\alpha}}\} = \arg\max_{\mathbf{w},\boldsymbol{\theta}}\Big((\bar{\boldsymbol{\alpha}}+\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w};\boldsymbol{\theta})\Big). \quad (27)$$

[1]Raising a fractional power $\frac{1}{\alpha_{\mathtt{lm}}}$ on HMM likelihoods known as acoustic deweighting [6].

Algorithm 3: Structured SVM training with Gaussian prior

0. Initial: $\bar{\boldsymbol{\alpha}} = [0,0,0\ldots]$, $\boldsymbol{\mu} = [1,0,0\ldots]$;
1. **Fixing** $\bar{\boldsymbol{\alpha}}$, **optimise** the reference alignment $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)},\ \forall\ r$,
$$\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)} = \arg\max_{\boldsymbol{\theta}^{(r)}}\Big\{(\bar{\boldsymbol{\alpha}}+\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}_{\mathtt{ref}}^{(r)};\boldsymbol{\theta}^{(r)})\Big\}, \quad (28)$$
2. **Fixing** $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)}$, **optimise** $\bar{\boldsymbol{\alpha}}$ by *minimizing*:
$$\frac{1}{2}||\bar{\boldsymbol{\alpha}}||_2^2 + \frac{C}{R}\sum_{r=1}^{R}\Big[-\bar{\boldsymbol{\alpha}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}_{\mathtt{ref}}^{(r)};\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)}) + \max_{\mathbf{w}\neq\mathbf{w}_{\mathtt{ref}}^{(r)},\boldsymbol{\theta}}\Big\{\bar{\mathcal{L}}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w}) + \bar{\boldsymbol{\alpha}}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w};\boldsymbol{\theta})\Big\}\Big]_{+} \quad (29)$$
where $\bar{\mathcal{L}}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w}) = \boldsymbol{\mu}^{\mathsf{T}}\Delta\boldsymbol{\phi}^{(r)} + \mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w})$.
$\bar{\boldsymbol{\alpha}}$ in problem (29) can be learned using Algorithm 2.
3. go back to Step 1, until converge **return** $\boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}} + \boldsymbol{\mu}$;

One interesting property of (25) is that even if $\bar{\boldsymbol{\alpha}}$ is not well trained, e.g., in the early training interation, with a proper $\boldsymbol{\mu}$ the algorithm will still generate sensible competing hypothesis and segmentation using equation (27). This is particularly helpful in reducing the convergence time in medium to large vocabulary CSR (see Section VI-B for more details).

IV. Implementation Issues

An efficient implementation of the training algorithm is important for medium and large vocabulary speech recognition systems. In Section III, a prior was introduced that helped reduce the number of training iterations. In this section several design options are described that have a substantial influence on computational efficiency. To reduce the memory cost, 1-slack optimisation is used as an alternative to n-slack optimisation. To reduce the training and decoding time, a lattice-based efficient search is proposed.

A. 1-slack optimisation

There are two forms of cutting plane algorithms [20], n-slack and 1-slack algorithms. The algorithm used in this work, and described in Algorithm 2, is the 1-slack version. An advantage of the 1-slack algorithm is that the number of constraints and support vectors generated is much smaller than for the n-slack case [20]. In theory, the n-slack algorithm can add R constraints at every iteration, where R is the size of training set. Conversely, the 1-slack algorithm adds at most a single constraint, $\bar{\boldsymbol{\alpha}}^{\mathsf{T}}\sum_{r=1}^{R}\Delta\boldsymbol{\phi}^{(r)} + \sum_{r=1}^{R}\bar{\mathcal{L}}(\mathbf{w}_{\mathtt{ref}}^{(r)},\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}) \leq \xi$, per iteration (as shown in Algorithm 2). For example, in AURORA 4 experiments, 1-slack algorithms produced less than 300 active constraints, whereas n-slack algorithms produced more than $50,000$ constraints after 20 iterations (still far from the converge). This makes n-slack algorithms impractical for large vocabulary CSR, since each constraint includes a 2210 dimensional joint feature vector. 20 iterations of n-slack optimisation required more around 18G of memory for AURORA 4. This rapidly becomes impractical using the current computer infrastructure. Thus for AURORA 4 experiments, only the result of 1-slack algorithm (with proper prior) is shown in Section VI-B.

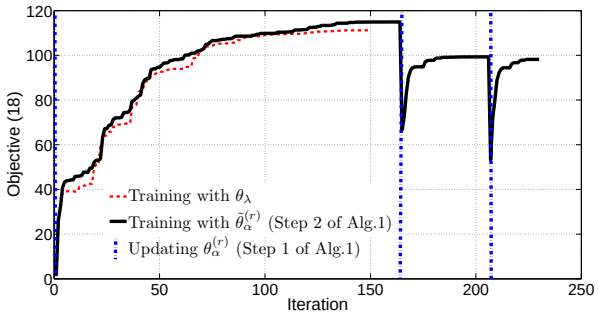

Fig. 6. Learning curves of SSVMs. Dashed curve: training with HMM segmentation $\boldsymbol{\theta}_{\boldsymbol{\lambda}}$. Vertical dashdotted lines: optimising reference segmentation $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)}$ (12). Solid curve: training with optimal competing segmentation $\tilde{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}^{(r)}$ (15).

Caching Another interesting property about 1-slack algorithm (Algorithm 2) is that constraints depend on $\sum_{r=1}^{R}\Delta\phi^{(r)}$ instead of individual $\phi(\mathbf{O}^{(r)},\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)};\tilde{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}^{(r)})$. Thus, the competing hypotheses $\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}$ can be involved in the set of active constraints many times. To avoid the computational cost of repeatedly searching for (the same) $\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)}$ in the large space (or lattice), the 10 most recently used $\phi(\mathbf{O}^{(r)},\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)};\tilde{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}^{(r)})$ for each training sample are cached. Therefore the search process for the best competing hypothesis (15) becomes

for $r=1,\ldots,R$ **do**
  $\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)} \leftarrow$ search equation (15) in the caches
end for
if $\sum_{r=1}^{R}\Delta\phi^{(r)}$ remains **then**
  $\tilde{\mathbf{w}}_{\boldsymbol{\alpha}}^{(r)} \leftarrow$ search equation (15) in decoding space, $\forall r$
end if

The aim of the caching strategy is to reduce the number of calls to search in the decoding space (or lattice).

Pruning For both the n-slack and 1-slack algorithms, constraints added to the working set in early iterations often become inactive later. These constraints can be removed without affecting the final solution. This is practically useful since it leads to a relatively smaller QP problem to be solved in later iterations. In this work constraints that have not been active for 50 iterations are pruned to reduce the memory cost.

Convergence According to Theorem 2 in [34], iterating step 1 and step 2 of Algorithm 1 is guaranteed to monotonically decrease the objective function (17) and will converge to a minimum or saddle point. For the AURORA 2 task, the criterion value for this algorithm against iteration is shown in Fig. 6. Every point in Fig. 6 is a minimum solution of the QP problem (14) in Algorithm 2 under a set of constraints. The criterion is increased because the cutting plane algorithm keeps adding more constraints to the QP [20] to get closer to the "real" minimum. However when $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(r)}$ is updated the objective function drops because the linear part of (13) decreases[2]. The gap between the solid curve and dashed curve indicates the differences from optimising the segmentation, $\boldsymbol{\theta}$ in (13), compared to the one obtained from the generative model, $\boldsymbol{\theta}_{\boldsymbol{\lambda}}$ in (7).

[2]The object drops also because the set of previous constraints discarded. Although in theory the previous constraints could be kept, for implementation simplicity this was not performed.

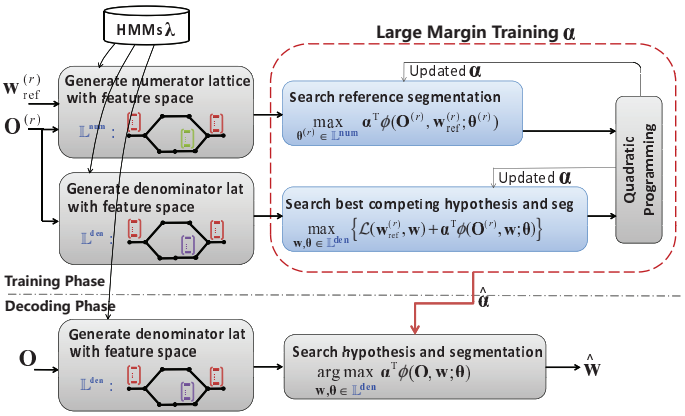

Fig. 7. The diagram of training and decoding for SSVMs.

B. Efficient search

Theoretically, the large margin training criterion discussed in the previous section can be directly applied to train the model parameter. In practice, to make the algorithms applicable to larger vocabulary systems additional speed improvements are required. There are two search sub-problems that must be solved efficiently (see Fig. 7): the best reference segmentation equation (12) in Algorithm 1 (or (28) in Algorithm 3); and the best competing hypothesis/segmentation, equation (15) in Algorithm 2. By using the approximate MPE loss [19] it is possible to approximate the loss $\mathcal{L}(\mathbf{w}_{\mathtt{ref}}^{(r)},\mathbf{w})$ at the segment level and incorporate it into (11). These two search problems can thus be solved using the Viterbi-style inference algorithm described in the Section II-B.

Lattices constrained search. In small vocabulary speech recognition task, it is feasible to search all possible segmentations and competing hypothesis in the two search problems. However, it is not practical for larger tasks because of the large search space of all possible $\mathbf{w}$ and $\boldsymbol{\theta}$. Similar to discriminative training in [19], numerator and denominator lattices $\mathbb{L}^{\mathtt{num}}$ and $\mathbb{L}^{\mathtt{den}}$ are generated to restrict the search space. Then a lattice-based search algorithm is used to find the best competing path (hypothesis) among the lattices. The MPE approximate loss [19] is computed at the arc level. Fig. 8 shows a lattice search where n is a node in the $\mathbb{L}^{\mathtt{den}}$, n' is one of its previous nodes, and $\boldsymbol{\rho}_n$ is the best path score at node n. Thus, the best competing path (hypothesis) in equation (15) can then be found using the following arc-level recursion

$$\boldsymbol{\rho}_n = \max_{n'\in\mathbb{L}^{\mathtt{den}}}\{\boldsymbol{\rho}_{n'} + s_{n'\rightarrow n}\} \tag{30}$$

where $s_{n'\rightarrow n}$ is the segmental score for the arc between n' and n. This lattice based arc-level Viterbi search is a degenerated version of (11). Similarly, equation (12) can also be efficiently searched in the numerator lattice $\mathbb{L}^{\mathtt{num}}$.

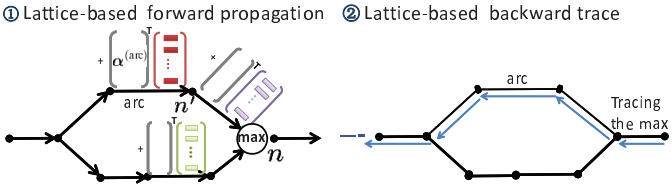

Fig. 8. Inference based on the lattices using arc-level forward-backward algorithm.

Parallelization. For large scale applications, the computational load during training is dominated by finding the best competing hypothesis/segmentation. For the n-slack algorithm, in order to run 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 the n-slack algorithm, this parallelization will decrease the performance slightly [26][3]. However the 1-slack algorithm used in this work can be easily parallelized without any degradation in performance. Paralleling the loop for equation (15) will lead to a substantial speed-up in the training process.

V. Adaptation

In speech recognition, the acoustic conditions during training and testing are seldom matched (due to inter-speaker variability, intra-speaker variability, background noise and channel distortions). For HMMs, a range of model adaptation researches have been devoted to handling this problem, such as maximum likelihood linear regression (MLLR) [42] and Vector Taylor Series (VTS) [43]. When applying these concepts to SSVMs there are two options. First, the discriminative model parameters, $\boldsymbol{\alpha}^{\mathsf{T}} = [\boldsymbol{\alpha}^{(v_1)^{\mathsf{T}}}, \ldots, \boldsymbol{\alpha}^{(v_{\mathrm{M}})^{\mathsf{T}}}]$, can be adapted. However with very limited data in the target domain, in these experiments a single utterance, this is very difficult.

Alternatively, the HMM parameters $\boldsymbol{\lambda}$ associated with the joint feature space can be adapted [16]. In this work VTS is used to handle background noise. Considering only the static elements of the acoustic models, the compensated mean vector and covariance matrix of component m of the generative model $\boldsymbol{\lambda}$ 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 consisting of partial derivatives of noise-corrupted speech with respect to clean speech [43]. $\mathbf{exp}()$ and $\mathbf{log}()$ are element-wise exponential and logarithm respectively. The noise model parameters are estimated using maximum likelihood estimation [44]. Thus in the target condition the parameters of proposed SSVMs can be assumed to be speaker and noise-independent, whereas the HMM parameters and joint feature spaces are speaker and noise-dependent.

VI. Experiments

This section describes the evaluation of the proposed SSVM algorithms described in previous sections. To illustrate that the proposed SSVMs can be adapted to the mismatched acoustic condition, the noise-corrupted corpus AURORA 2 and 4 were used. The AURORA 2 corpus is used to contrast between the performance of 1-slack and n-slack algorithms, and demonstrate the gains from optimising the segmentation and modeling the prior. The AURORA 4 experiments are used to illustrate the performance of the proposed SSVM framework for medium vocabulary speech recognition. The 5K Wall Street Journal (WSJ0) data, the clean part of AURORA 4, were used to evaluate the performance of SSVMs excluding the noise affects.

[3]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 II
AURORA 2 results (WER %) of VTS based HMM, Multi-class SVMs (MSVM) [14], log linear model (LLM) [29], and SSVMs using n-slack algorithms.

Model	Param.	Criteria	Set A	Set B	Set C	Avg.
HMM-VTS	46,732	ML	9.8	9.1	9.5	9.5
MSVM	+144	LM	8.3	8.1	8.6	8.3
LLM	+144	CML	8.1	7.7	8.3	8.1
SSVM	+144	LM (n-slack)	7.8	7.3	8.0	7.6

TABLE III
AURORA 2 results (WER %) of VTS based HMM, MSVM, LLM and SSVM (n-slack) in different SNRs conditions.

SNR	Set A				Avg. of Set A, B and C			
(dB)	HMM	MSVM	LLM	SSVM	HMM	MSVM	LLM	SSVM
20	1.7	1.5	1.4	1.3	1.6	1.4	1.4	1.2
15	2.4	2.0	1.9	1.8	2.4	2.0	2.0	1.8
10	4.4	3.6	3.5	3.3	4.3	3.6	3.6	3.4
05	11.2	9.2	8.9	8.7	10.7	9.1	8.8	8.5
00	29.6	25.1	24.9	23.9	28.5	25.4	24.5	23.5
Avg	9.8	8.3	8.1	7.8	9.5	8.3	8.1	7.6

A. AURORA 2

AURORA 2 is a small vocabulary continuous digit 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 8440 clean mixed-gender training utterances were used to train the acoustic generative models (HMMs). 39 dimensional observations consisting of 12 MFCCs appended with the zeroth cepstrum, delta and delta-delta coefficients were used. The generative models, HMMs, were 16 emitting states whole word digit models, with 3 mixtures per state. All three test sets, A, B and C, were used. For sets A and B, there were a total of 8 noise conditions (4 in each) at 5 different SNRs, 0dB to 20dB. For test set C noise convolutional distortion was also added. Set A was used as the development set for tuning parameters for all systems, such as the penalty factor C in SSVMs. The parameters of SSVMs were trained using the same subset of the multi-condition training data as [16]: three of the four subsets (N2-N4) and three of five SNRs (10dB, 15dB, 20dB).[4]

To evaluate the benefit of the proposed SSVMs framework, a range of configuration were compared. The baseline generative system was HMM based with VTS compensation. These compensated HMMs were also used to derive: the noise robust joint feature space; the word-level segmentation for the multi-class SVMs; and the lattices for the structured SVM training and inference. For all configurations the joint feature space was based on appended-all likelihood features (3). For this task no language model was used. The performances of VTS-compensated HMM, the log-linear models proposed in [29],

[4]This allows the generalisation of the SVMs, LLMs and SSVMs to unseen noise conditions to be evaluated on test set A as well as the test sets B and C, as no data from noise condition N1 and SNRs 5dB and 0dB were used.

TABLE IV
AURORA 2 RESULTS (WER %) OF SSVMS TRAINED USING n-SLACK ALGORITHM WITHOUT/WITH OPTIMISING $\boldsymbol{\theta}$ AND 1-SLACK ALGORITHMS WITHOUT/WITH GAUSSIAN PRIOR (ALG. 3+2). SV IS SHORT FOR SUPPORT VECTORS.

Training Algorithm	$\boldsymbol{\theta}$	# SV	**Set A**	**Set B**	**Set C**	**Avg.**
n-slack [14]	$\boldsymbol{\theta}_{\boldsymbol{\lambda}}$	629	7.8	7.3	8.0	7.6
n-slack [26]	$\boldsymbol{\theta}_{\boldsymbol{\alpha}}$	642	7.6	7.1	7.8	7.4
1-slack (Alg. 1+2)	$\boldsymbol{\theta}_{\boldsymbol{\alpha}}$	29	7.6	7.3	7.9	7.5
1-slack-$\boldsymbol{\mu}$ (Alg. 3+2)	$\boldsymbol{\theta}_{\boldsymbol{\alpha}}$	30	7.5	7.1	7.9	7.4

multi-class SVMs [14] and SSVMs with different training algorithms and criteria are shown in Table II and IV. The log-linear model (LLM) was trained with L_2 regularization. For reference, the detailed results on different SNR of Set A are also shown in Table III.

Examining the results in Table II shows that the structured SVM achieved the best results among all the systems. The multi-class SVM is an unstructured model where the observation sequence is first segmented into words based on HMMs and individual "segmented" words classified independently. The difference in performance between the structured SVM and multi-class SVM systems shows the impact of fixing the segmentation rather than including structures in the model. The overall gain from using SSVMs over the VTS-compensated HMM system was over 20%.[5] Note that the number of parameters in HMM and proposed SSVM system are in the same range—more than $45,000$ for HMM and 144 more for SSVM. Thus, the improvement obtained were not just the result of increasing parameters.

Examining the results in Table IV, the first two lines show that optimising the segmentation yields small, but consistent, gains, in performance over using the HMM-based alignment $\boldsymbol{\theta}_{\boldsymbol{\lambda}}$, about 3% relative reduction on average. The results also show the benefit of using the 1-slack algorithm as the WER are almost the same with n-slack algorithm but with far fewer support vectors (29 compared with 629) and less memory cost (83M compared with 922M). Small gains are also observed when training SSVMs 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).

B. AURORA 4

AURORA 4 is a medium vocabulary task based on the Wall Street Journal (WSJ) data. Four configurations of canonical HMMs were considered. The first repeats the previous setup where the HMMs were trained using clean data (SI-84 WSJ0 part, 14 hours) and used with VTS compensation. In the second more advanced systems VTS-adaptive training (VAT) was used to obtain the canonical HMM [45]. In the third MPE and VAT training was used to obtain the canonical HMM [46]. In the final experiment clean trained HMMs without VTS are applied to demonstrate the performance of SSVMs excluding the noise affect. For all setups the HMMs were state-clustered triphones (3140 states) with 16 components/mixture. Four iterations of VTS compensation were performed for the training and test data. To compare with the log linear model proposed in [29], the joint feature space of the SSVMs follows the same setup as in [29], 47×47 dimensions. In the first three configurations, both the log linear models and SSVMs are trained on the multi-style data. Evaluation was performed using the standard 5000- word WSJ0 bigram model on four noise-corrupted test sets based on NIST Nov'92 WSJ0 test set. Test set A is clean, set B has 6 types of noise added, set C has the channel distortion introduced (desk-mounted secondary microphones recorded) and set D has both additive noise and channel distortion. The average SNR in noise-corrupted data is 10 dB. Set B is used as the development set for tuning parameters of all systems.

[5]The proposed SSVM based on the "uncompensated" HMMs is also evaluated. The performance 37.8 on set A. Comparing with "uncompensated" HMMs baseline 43.8, a consistent improvement is achieved.

TABLE V
AURORA 4 RESULTS BASED ON VTS-COMPENSATED HMMS. FOR SSVM, $\boldsymbol{\mu}$ MEANS LARGE MARGIN TRAINING $\boldsymbol{\alpha}$ WITH GAUSSIAN PRIOR (ALG. 3+2). ALL POSSIBLE HYPOTHESIS $\mathbf{w}$ AND SEGMENTATIONS $\boldsymbol{\theta}$ ARE RESTRICTED BY LATTICES GENERATED BY HMM-VTS.

Model	Param.	Criterion	Test Set WER (%)				Avg
			A	B	C	D	
HMM-VTS	3.98M	ML	7.1	15.3	12.2	23.1	17.8
LLM	+2210	CML	7.2	14.7	11.1	22.8	17.4
		MPE	7.3	14.7	11.2	22.7	17.4
SSVM	+2210	LM (1-slack-$\boldsymbol{\mu}$)	7.4	14.2	11.3	21.9	16.8

TABLE VI
AURORA 4 RESULTS BASED ON VTS ADAPTIVELY TRAINED HMMS. ALL POSSIBLE HYPOTHESIS $\mathbf{w}$ AND SEGMENTATIONS $\boldsymbol{\theta}$ ARE RESTRICTED BY LATTICES GENERATED BY HMM-VAT.

Model	Criterion	Test Set WER (%)				Avg
		A	B	C	D	
HMM-VAT	ML	8.6	13.8	12.0	20.1	16.0
LLM	CML	7.8	13.6	11.3	20.2	15.8
	MPE	7.7	13.5	11.2	20.0	15.7
SSVM	LM (1-slack-$\boldsymbol{\mu}$)	7.5	13.3	11.1	19.6	15.4

The first configuration used clean trained HMMs with VTS compensation. Table V shows the AURORA4 results of SSVMs trained with a general Gaussian prior (Algorithm 3). The mean of the prior was set as the parameters of CML trained log-linear model. The log linear models [29] and SSVMs based on the same 2210 $(47 \times 47 + 1)$ dimensional joint features. Compared to the CML trained LLM, SSVMs yielded a 3.4% relative reduction in WER. For this task, the n-slack algorithm cannot be applied due to memory issues (more than 18G required) described in Section IV-A. The 1-slack algorithm without a proper prior is also impractical as the number of iterations becomes very large for this size of feature space (over 1000 iterations were run without convergence). The only algorithm that can be applied is the proposed 1-slack-$\boldsymbol{\mu}$ algorithm (it converged in 258 iterations), as the prior yields sensible model parameters when there are few constraints as described in Section III. For small vocabulary tasks, searching for the best competing hypothesis and segmentation is feasible. However, it is not practical to do this in AURORA4. Therefore the search space of all possible $\mathbf{w}, \boldsymbol{\theta}$ here are restricted by the lattices. Given the lattices, the decoding complexities of SSVMs and HMMs are in the same order of magnitude.

The second configuration used a VTS adaptively trained (VAT) HMM system. Note in this configuration both the generative and discriminative models were trained on multistyle

TABLE VII
AURORA 4 RESULTS BASED ON MPE-VAT TRAINED HMMS. ALL POSSIBLE HYPOTHESIS $\mathbf{w}$ AND SEGMENTATIONS $\boldsymbol{\theta}$ ARE RESTRICTED BY LATTICES GENERATED BY MPE-VAT HMMS.

Model	Criterion	Test Set WER (%)				Avg
		A	B	C	D	
HMM-MPE-VAT	MPE	7.2	12.8	11.5	19.7	15.3
SSVM	LM (1-slack-$\boldsymbol{\mu}$)	6.9	12.7	11.2	19.4	15.0

TABLE VIII
WSJ0 RESULTS BASED ON MAXIMUM LIKELIHOOD TRAINED HMMS. ALL POSSIBLE HYPOTHESIS $\mathbf{w}$ AND SEGMENTATIONS $\boldsymbol{\theta}$ ARE RESTRICTED BY LATTICES GENERATED BY HMMS.

Model	Criterion	Test Set WER (%)
HMM	ML	7.3
SSVM	LM (1-slack-$\boldsymbol{\mu}$)	6.8

data. Table VI shows the performance of the baseline VAT system, the log-linear models [29] and SSVMs based on the same 2210 dimensional joint features. These features were extracted using the likelihoods of VTS adaptively trained HMMs. Comparing the VAT in Table VI (line 1) and the VTS in Table V (line 1) shows gains of about 2% absolute. In comparison with the HMM-VAT system and LLMs, the SSVMs gain on average about 4% and 2% relative improvements.

The third configuration used an MPE and VAT trained HMM system. In this configuration the generative and discriminative models were both discriminatively trained. Table VII shows the performance of baseline MPE-VAT HMMs, the log-linear models [29] and SSVMs based on the same 2210 dimensional joint features. These features were extracted using likelihoods of MPE-VAT HMMs. In comparison with the MPE-VAT HMMs, the proposed SSVMs on average yield 2% relative improvement.

In the fourth configuration both generative and discriminative models were trained and evaluated on the clean part of AURORA4 (the standard 5K WSJ0 setup). SSVMs were based on the same 2210 dimensional joint features. These features were extracted using likelihoods of clean HMMs. The WER ($\%$) of the clean HMMs and proposed SSVMs are shown in Table VIII. [6] The relative improvement is 7%.

VII. CONCLUSION

This paper has described a structured SVM framework suitable for medium to large vocabulary speech recognition. Several theoretical and practical extensions to previous work on small vocabulary task [14] are reported. First, the joint feature space based on word models is extended to allow context-dependent triphone models to be used. Second, since the joint feature space requires the segmentation of frames into words/subwords to be specified, an optimal segmentation approach is described. Third, by interpreting the structured SVM as a large margin log linear model, illustrates 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. This assumption is relaxed by incorporating a more general Gaussian prior into the large margin training criterion in a form that allows the cutting plane algorithm to be directly applied. Finally, to speed up the training process, strategies such as 1-slack algorithm, caching competing hypothesis and parallelization are proposed.

The performance of SSVMs was evaluated on AURORA 2 and 4. Significant gains are observed over both HMMs and log linear models. In this work the generative model parameters $\boldsymbol{\lambda}$, are assumed to have been trained. Joint learning $\{\boldsymbol{\lambda}, \boldsymbol{\alpha}\}$ in the large margin framework will be investigated in the future. Future work will also involve the kernelization of the proposed structured SVM to support high dimensional feature spaces such as the derivative feature space [28].

[6]Note the HMM performance 7.3% in Table VIII is slightly worse than 7.1%, the VTS set A result in Table V, because VTS on clean data is actually performing utterance-dependent normalisation.

ACKNOWLEDGMENT

The authors would like to thank Toshiba Research Europe Ltd, Cambridge Research Lab, for partly funding this work and the valuable comments given by the reviewers.

REFERENCES

[1] S.-X. Zhang and M. J. F. Gales, "Extending noise robust structured support vector machines to larger vocabulary tasks," in *Proc. ASRU*, Hawaii, USA, 2011.
[2] M. J. F. Gales, D. Kim, P. Woodland, H. Chan, D. Mrva, R. Sinha, and S. Tranter, "Progress in the CU-HTK broadcast news transcription system," *IEEE Transactions Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1513–1525, 2006.
[3] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin, "Learning structured prediction models: a large margin approach," in *Proceedings of Int. Conf. on Machine Learning*, 2005, pp. 896–903.
[4] G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan, *Predicting Structured Data (Neural Information Processing)*. The MIT Press, 2007.
[5] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," in *Foundations and Trends in Signal Processing*, p. 2007.
[6] P. Woodland and D. Povey, "Large scale discriminative training of hidden Markov Models in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 25–48, Jan 2002.
[7] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing*, vol. 40, no. 12, p. 3043, 1992.
[8] W. Byrne, "Minimum Bayes risk estimation and decoding in large vocabulary continuous speech recognition," *IEICE Transactions*, vol. 89-D, no. 3, pp. 900–907, 2006.
[9] F. Sha and L. K. Saul, "Large margin hidden Markov models for automatic speech recognition," in *Neural Information Processing Systems*, 2007, pp. 1249–1256.
[10] J. Keshet, C.-C. Cheng, M. Stoehr, and D. A. McAllester, "Direct error rate minimization of hidden Markov models," in *Proc. Interspeech*, 2011, pp. 449–452.
[11] G. Heigold, G. Zweig, X. Li, and P. Nguyen, "A flat direct model for speech recognition," in *ICASSP*, 2009, pp. 3861–3864.
[12] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *ASRU*, 2009.
[13] M. Layton and M. Gales, "Augmented statistical models for speech recognition," in *Proc. ICASSP*, Toulouse, 2006.
[14] 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.
[15] G. Heigold, "A log-linear discriminative modeling framework for speech recognition," Ph.D. dissertation, RWTH Aachen University, Aachen, Germany, 2010.
[16] 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.

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

[18] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," in *NIPS*, 2004.

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

[20] 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.

[21] O. Birkenes, T. Matsui, and K. Tanabe, "Isolated-word recognition with penalized logistic regression machines," in *ICASSP 2006*, vol. 1, 2006.

[22] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of Int. Conf. on Machine Learning*, 2001.

[23] G. Zweig and P. Nguyen, "From flat direct models to segmental CRF models," in *ICASSP*, 2010, pp. 5530–5533.

[24] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, 2002.

[25] V. Venkataramani, S. Chakrabartty, and W. Byrne, "Support vector machines for segmental minimum Bayes risk decoding of continuous speech," in *ASRU*, 2003.

[26] 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.

[27] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proceedings of the 1998 conference on Advances in neural information processing systems II*. Cambridge, MA, USA: MIT Press, 1999, pp. 487–493.

[28] A. Ragni and M. J. F. Gales, "Derivative kernels for noise robust ASR," in *Proc. ASRU*, Hawaii, USA, 2011.

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

[30] Z. Ghahramani and M. I. Jordan, "Factorial Hidden Markov Models," *Machine Learning*, vol. 29, pp. 245–273, 1997.

[31] S. Sarawagi and W. W. Cohen, "Semi-Markov conditional random fields for information extraction," in *NIPS*, 2005.

[32] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, pp. 1627–1645, 2010.

[33] C.-N. Yu and T. Joachims, "Learning structural SVMs with latent variables," in *Proceedings of ICML*, 2009.

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

[35] Y. Singer and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for SVM," in *In ICML*, 2007, pp. 807–814.

[36] A. Globerson, T. Y. Koo, X. Carreras, and M. Collins, "Exponentiated gradient algorithms for log-linear structured prediction," in *In Proc. ICML*, 2007, pp. 305–312.

[37] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, 2005.

[38] T. Jebara, "Discriminative, generative and imitative learning," Ph.D. dissertation, USA, 2001.

[39] J. Li, M. Yuan, and C.-H. Lee, "Approximate test risk bound minimization through soft margin estimation," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 8, pp. 2393–2404, 2007.

[40] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *ICASSP*. IEEE, 2008, pp. 4057–4060.

[41] M. Layton, "Augmentd statistical models for classifying sequence data," *Ph.D. thesis*, 2006.

[42] M. J. F. Gales, D. Pye, and P. Woodland, "Variance compensation within the MLLR framework for robust speech recognition and speaker adaptation," in *Proc. ICSLP*, vol. 3, 1996, pp. 1832–1835.

[43] 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.

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

[45] O. Kalinli, M. L. Seltzer, and A. Acero, "Noise adaptive training using a Vector Taylor Series approach for noise robust automatic speech recognition," in *ICASSP*, 2009, pp. 3825–3828.

[46] F. Flego and M. Gales, "Factor analysis based VTS and JUD noise estimation and compensation," Cambridge University, Tech. Rep. CUED/F-INFENG/TR653, 2011, available from: http://mi.eng.cam.ac.uk/∼mjfg.