Structured Support Vector Machines for Noise Robust Continuous Speech Recognition

Shi-Xiong Zhang and Mark Gales

Department of Engineering, University of Cambridge, Cambridge, UK
{sxz20, mjfg}@eng.cam.ac.uk

Abstract

The use of discriminative models is an interesting alternative to generative models for speech recognition. This paper examines one form of these models, structured support vector machines (SVMs), for noise robust speech recognition. One important aspect of structured SVMs is the form of the joint feature space. In this work features based on generative models are used, which allows model-based compensation schemes to be applied to yield robust joint features. However, these features require the segmentation of frames into words, or subwords, to be specified. In previous work this segmentation was obtained using generative models. Here the segmentations are refined using the parameters of the structured SVM. A Viterbi-like scheme for obtaining "optimal" segmentations, and modifications to the training algorithm to allow them to be efficiently used, are described. The performance of the approach is evaluated on a noise corrupted continuous digit task: AURORA 2.

Index Terms: speech recognition, structural SVMs, optimal alignment, large margin, log linear model

1. Introduction

Discriminative training [1] of Hidden Markov Models (HMMs) has been shown to yield performance gains for automatic speech recognition (ASR). However the underlying models are still generative, with the standard HMM conditional independence assumptions, and sentence posteriors obtained using Bayes' rule. This has led to interest in discriminative models, e.g., Structured Conditional Random Fields (SCRF) [2], and structured Log Linear Model (LLM) [3, 4], 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 [2,4]. 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 to handle speaker and noise condition changes [5]. Discriminative models are often trained using Conditional Maximum Likelihood (CML) [2, 3]. Alternatively, there has been interest in large margin [4, 6] and minimum Bayes' risk [1] criteria. It has been shown that the large margin trained log linear models can be viewed as structured SVMs [4]. To handle continuous speech, structured discriminative models often require a segmentation of the frames into word, or sub-word units. For approaches such as SCRFs, where word-level features are used, these segmentations are defined by standard HMM acoustic models. However for approaches where the underlying acoustic models are altered [3, 4], the segmentation should be a function of the discriminative model parameters. This paper extends the previous work with structured SVMs (SSVM) [4] to enable optimal segmentations, based on the current discriminative model, to be used for both training and inference.

Previously, the segmentation for both training and inference were based on the generative models (used to obtain the features). This paper shows that a Viterbi-like algorithm can be defined to obtain the segmentation with the discriminative model parameters for a particular form of feature-space. This scheme is related to inference with factorial HMMs [7]. If the segmentation is optimised during training, then it is necessary to modify the structured SVM training algorithm. The convex problem of standard structured SVM training becomes a difference of convex programming problem. The optimization can be solved using the concave-convex procedure [8] and cutting-plane algorithm [9]. An additional issue for training is the computational cost of obtaining the segmentation. To handle this a "batch"-mode update of the structured SVM parameters is proposed, where the discriminative model parameters are updated after seeing blocks of training data, rather than sequentially.

The impact of the segmentation on speech recognition performance is evaluated on a standard continuous digit noise-corrupted speech recognition task, AURORA 2.

2. Structured Support Vector Machines

One of the key issues for using structured discriminative models is to derive an appropriate *joint* feature space, $\boldsymbol{\phi}(\mathbf{O}, \mathbf{w})$, for a given task. This represents the structured relationship between the observation sequence, $\mathbf{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$, and the corresponding label sequence, $\mathbf{w} = \{w_1, \ldots, w_L\}$. For some applications there is a direct mapping between the observations and the labels, the relationship between $(\mathbf{O}, \mathbf{w})$ can be fully described by the pair itself. However, for continuous speech recognition, the relationship between the observations, the frames, and the labels, words, is normally not known and must be inferred given some model. This requires an additional level of latent variable $\boldsymbol{\theta}$ that represents this alignment.

In previous work on structured SVMs for ASR [4], the model used to infer the alignment was the standard generative model HMMs. This alignment was fixed for both inference and throughout training. Thus the joint feature space $\boldsymbol{\phi}(\mathbf{O}, \mathbf{w}; \hat{\boldsymbol{\theta}}_{\text{hmm}}, \boldsymbol{\lambda})$ was based on the pre-fixed alignments $\hat{\boldsymbol{\theta}}_{\text{hmm}}$. For inference this yields

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

$$\hat{\boldsymbol{\theta}}_{\text{hmm}} = \arg\max_{\boldsymbol{\theta}} \ \log P(\boldsymbol{\theta} | \mathbf{O}, \mathbf{w}; \boldsymbol{\lambda}), \tag{2}$$

where $\boldsymbol{\lambda}$ are the HMM parameters and $\boldsymbol{\alpha}$ are the structured SVM parameters. Although $\hat{\boldsymbol{\theta}}_{\text{hmm}}$ is the most likely alignment for the generative model, it may not be the best alignment to describe the relationship between $(\mathbf{O},\mathbf{w})$ for the discriminative models. There may be a mismatch between (1) and (2).

Instead of using pre-fixed values, the alignment variable $\boldsymbol{\theta}$ can be optimised for both decoding and training. For general feature-spaces it is not possible to define efficient algorithms for this task. However for the log-likelihood feature-spaces this is possible. The alignment $\boldsymbol{\theta}$ partitions the observation sequence into L segments $\mathbf{O} = \{\mathbf{O}_{t(w_1,\boldsymbol{\theta})},\dots,\mathbf{O}_{t(w_i,\boldsymbol{\theta})},\dots,\mathbf{O}_{t(w_L,\boldsymbol{\theta})}\}$. The resulting joint feature space is defined as

$$\boldsymbol{\phi}(\mathbf{O},\mathbf{w},\boldsymbol{\theta};\boldsymbol{\lambda}) \triangleq \frac{1}{T}\left[\sum_{i=1}^{L}\boldsymbol{\delta}(w_i)\otimes\boldsymbol{\varphi}^{\text{LL}}(\mathbf{O}_{t(w_i,\boldsymbol{\theta})};\boldsymbol{\lambda})\right] \tag{3}$$

where $\otimes$ 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 $\boldsymbol{\varphi}^{\text{LL}}(\mathbf{O}_{t(w_i,\boldsymbol{\theta})};\boldsymbol{\lambda})$ is the generative model based log likelihood feature space for segment $\mathbf{O}_{t(w_i,\boldsymbol{\theta})}$

$$\boldsymbol{\delta}(w)=\begin{bmatrix}\delta(w-v_1)\\ \vdots\\ \delta(w-v_\text{M})\end{bmatrix},\boldsymbol{\varphi}^{\text{LL}}(\mathbf{O};\boldsymbol{\lambda})=\begin{bmatrix}\log(p(\mathbf{O};\boldsymbol{\lambda}^{(v_1)}))\\ \vdots\\ \log(p(\mathbf{O};\boldsymbol{\lambda}^{(v_\text{M})}))\end{bmatrix}. \tag{4}$$

One interesting property of this joint feature space is the dot-product of the $\boldsymbol{\phi}(\mathbf{O},\mathbf{w},\boldsymbol{\theta};\boldsymbol{\lambda})$ and structured SVM parameter $\boldsymbol{\alpha}$ can be evaluated by accumulating every segment score [4]

$$\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w},\boldsymbol{\theta};\boldsymbol{\lambda})=\frac{1}{T}\sum_{i=1}^{L}\boldsymbol{\alpha}^{(w_i)}{}^{\mathsf{T}}\boldsymbol{\varphi}^{\text{LL}}(\mathbf{O}_{t(w_i,\boldsymbol{\theta})};\boldsymbol{\lambda}), \tag{5}$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}^{(v_1)}{}^{\mathsf{T}},\dots\boldsymbol{\alpha}^{(v_k)}{}^{\mathsf{T}}\dots,\boldsymbol{\alpha}^{(v_\text{M})}{}^{\mathsf{T}}]^{\mathsf{T}}_{\text{M}^2}$ in which $\boldsymbol{\alpha}^{(w)}=[\alpha_1^{(w)},\dots\alpha_k^{(w)}\dots,\alpha_\text{M}^{(w)}]^{\mathsf{T}}$.

If the segmentation of the observation sequence is allowed to vary, the decoding formula (1) becomes

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

Given the log-likelihood joint feature-space (3) a Viterbi-style algorithm to solve (6) can be found. Based on (5) it is possible to express the maximisation in (6) as

$$\arg\max_{\mathbf{w}}\left\{\max_{\boldsymbol{\theta}}\sum_{i=1}^{|\mathbf{w}|}\sum_{k=1}^{M}\alpha_k^{(w_i)}\log p(\mathbf{O}_{t(w_i,\boldsymbol{\theta})};\boldsymbol{\lambda}^{(v_k)})\right\} \tag{7}$$

This expression is related to forms of factorial HMM inference [7]. The search process (7) involves two distinct terms. The first is, given the alignment the score for each model, the log-likelihood, needs to be computed for each segment. This is the standard forward-backward algorithm for HMMs. The second is deriving the segmentation which requires a modified Viterbi search. This two stage process is illustrated in Fig. 1. The M HMMs are shown in parallel with *synchronisation points* shown in black which are determined by the segment boundaries.

The search process to find the optimal segmentation is similar to a semi-Markov search process. The best score (and alignment history) for the start of the segment is stored, $\phi(t_{\texttt{st}})$,

$$\phi(t_{\texttt{st}})=\max_{\mathbf{w},\boldsymbol{\theta}}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{1:t_{\texttt{st}}},\mathbf{w},\boldsymbol{\theta};\boldsymbol{\lambda}) \tag{8}$$

Given this start time, $t_{\texttt{st}}$, the forward score for the model is computed at the end state of each model, v_k, up-to time t, $\log(p(\mathbf{O}_{t_{\texttt{st}}:t};\boldsymbol{\lambda}^{(v_k)}))$. The best score for the start of the next segment (the end of the current segment) can be expressed as

$$\phi(t)=\max_{t_{\texttt{st}},w}\left\{\phi(t_{\texttt{st}})+\sum_{k=1}^{M}\alpha_k^{(w)}\log(p(\mathbf{O}_{t_{\texttt{st}}:t};\boldsymbol{\lambda}^{(v_k)}))\right\} \tag{9}$$

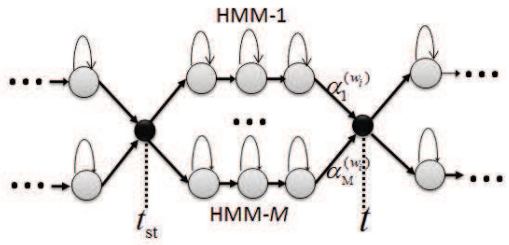

Figure 1: Decoding procedure illustration. The black circles indicate the synchronisation points where the M HMM log likelihoods are merged.

The above process is based on Viterbi-style search. Alternative, more efficient approximations, e.g., Gibbs sampling and variational methods [7], could be used to reduce the computation load, but is not investigated in this work.

3. Large Margin Training

The previous section has shown that given $\boldsymbol{\alpha}$ the optimal alignment $\boldsymbol{\theta}$ can be inferred. However, during training both $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ are unknown and dependent 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.

Given training data pairs $(\mathbf{O}^{(1)},\mathbf{w}^{(1)}_{\texttt{ref}}),\dots,(\mathbf{O}^{(R)},\mathbf{w}^{(R)}_{\texttt{ref}})$, similar to the latent SVM [10] and structured SVM [9, 11], the parameters of structured SVM can be trained by solving the following optimization problem:

$$\begin{aligned}&\min_{\boldsymbol{\alpha},\boldsymbol{\xi}}\ \frac{1}{2}||\boldsymbol{\alpha}||^2+\frac{C}{R}\sum_{r=1}^{R}\xi_r \qquad (10)\\ &\text{s.t.}\ \max_{\boldsymbol{\theta}^{(r)}}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}^{(r)}_{\texttt{ref}},\boldsymbol{\theta}^{(r)};\boldsymbol{\lambda})-\max_{\boldsymbol{\theta}}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w},\boldsymbol{\theta};\boldsymbol{\lambda})\\ &\qquad\geq\mathcal{L}(\mathbf{w}^{(r)}_{\texttt{ref}},\mathbf{w})-\xi_r,\quad 1\leq r\leq R,\ \forall\,\mathbf{w}\neq\mathbf{w}^{(r)}_{\texttt{ref}},\end{aligned}$$

where $\xi_r>0$ are the slack variables and $\mathcal{L}(\mathbf{w}^{(r)}_{\texttt{ref}},\mathbf{w})$ is the loss function. The constraints in (10) can be explained as follows. For every training pair $(\mathbf{O}^{(r)},\mathbf{w}^{(r)}_{\texttt{ref}})$, the best score of the correct pair should be greater than all competing pairs by a margin determined by the loss. The difference this form and the criterion used in previous work is that the reference alignment, $\boldsymbol{\theta}^{(r)}$, and best competing path alignment, $\boldsymbol{\theta}$, are optimised.

Substituting the slack variable in the constraints to the objective function, (10) can be reformulated as *minimizing*

$$\begin{aligned}&\frac{1}{2}||\boldsymbol{\alpha}||_2^2+\frac{C}{R}\sum_{r=1}^{R}\Big[\overbrace{-\max_{\boldsymbol{\theta}^{(r)}}\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}^{(r)}_{\texttt{ref}},\boldsymbol{\theta}^{(r)};\boldsymbol{\lambda})\right)}^{\text{concave}}\\ &\underbrace{+\max_{\mathbf{w}\neq\mathbf{w}_{\texttt{ref}},\boldsymbol{\theta}}\left\{\mathcal{L}(\mathbf{w},\mathbf{w}^{(r)}_{\texttt{ref}})+\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w},\boldsymbol{\theta};\boldsymbol{\lambda})\right\}}_{\text{convex}}\Big]_+ \qquad (11)\end{aligned}$$

where $[\ ]_+$ is the hinge-loss function. The constraints in (10) specify a set of linear functions. They are convex with respect to $\boldsymbol{\alpha}$. However, the objective function in (10), as also shown in (11), is non-convex. To solve this non-convex optimization problem, an approach similar to the concave-convex procedure [8, 10, 11] can be applied. The process is shown in Algorithm 1.

There are two search sub-problems in Algorithm 1, the best reference alignment (12) and the best competing hypothesis/alignment (13) in Step 2. Note by using the approximate

```
Algorithm 1: Joint learning algorithm for $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$.
0. Initial: $\tau = 0$, $\boldsymbol{\alpha}^{[0]} = [1, 0, 0 \ldots]$, $\hat{\boldsymbol{\theta}}^{(r)}[0] = \hat{\boldsymbol{\theta}}^{(r)}_{\text{hmm}}$ ;
1. Fixing $\boldsymbol{\alpha}$, optimise variable alignment $\boldsymbol{\theta}$ for each
   training pair $(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}})$ using Viterbi algorithm:
   $\hat{\boldsymbol{\theta}}^{(r)}[\tau] = \arg\max_{\boldsymbol{\theta}} \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}, \boldsymbol{\theta}; \boldsymbol{\lambda})$   (12)
2. Fixing $\hat{\boldsymbol{\theta}}^{(r)}[\tau] \; \forall \; r$, optimise $\boldsymbol{\alpha}$ by minimizing the
   following convex upper bound using cutting plane
   algorithm in [9] ( (11) $\leq$ (13)):
   $\frac{1}{2}||\boldsymbol{\alpha}||^2_2 + \frac{C}{R}\sum_{r=1}^{R}\Big[\overbrace{-\Big(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}^{(r)}_{\mathtt{ref}}, \hat{\boldsymbol{\theta}}^{(r)}[\tau]; \boldsymbol{\lambda})\Big)}^{\text{linear}}$   (13)
   $+ \max_{\mathbf{w} \neq \mathbf{w}^{(r)}_{\mathtt{ref}}, \boldsymbol{\theta}} \Big\{\mathcal{L}(\mathbf{w}, \mathbf{w}^{(r)}_{\mathtt{ref}}) + \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}, \boldsymbol{\theta}; \boldsymbol{\lambda})\Big\}\Big]_{+}$
3. $\tau = \tau + 1$, go back to Step 1 until converge;
```

MPE loss [1] it is possible to approximate the loss $\mathcal{L}(\mathbf{w}^{(r)}_{\mathtt{ref}}, \mathbf{w})$ at the segment level and incorporate it into (9).

These two search problems can both be solved using the Viterbi algorithm described in the previous section. However the computational load during training is dominated by searching for the best competing hypothesis/alignment. To enable this form of approach to be applied to reasonable size speech tasks, the sequential update mode of the standard cutting plane algorithm is modified to a batch-mode update. This allows Step 2 of Algorithm 1 to be run *in parallel* on many machines. This yields a substantial speed-up in the training process.

According to Theorem 2 in [8], iterating steps 1 and 2 of Algorithm 1 is guaranteed to monotonically decrease the objective function (11) 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. 2. Every point in Fig. 2 is a minimum solution of the QP problem (Step 2) under a certain set of constraints. The objective is increasing because the cutting plane algorithm keeps adding constraints. When updating $\boldsymbol{\theta}^{(r)}$ the objective function drops because the linear part of (13) decreases, and the set of previous constraints discarded [1]. The gap between the solid curve and dashed curve indicates the differences from incorporating the optimal competing hypothesis alignment, $\boldsymbol{\theta}$ in (13), compared to the one obtained from the generative model, $\hat{\boldsymbol{\theta}}_{\text{hmm}}$ [4].

The training criterion in Eq. 11 can be also viewed as large margin training of log linear models. If the margin for log linear models is defined as the log posterior probability ratio of the best alignment of $\mathbf{w}^{(r)}_{\mathtt{ref}}$, $\boldsymbol{\theta}^{(r)}$, and best competing hypothesis/alignment, $\{\mathbf{w}, \hat{\boldsymbol{\theta}}\}$, the large margin training for log linear model can be expressed as minimising [4, 6] (considering one utterance r only)

$$\left[\max_{\mathbf{w} \neq \mathbf{w}^{(r)}_{\mathtt{ref}}} \left\{\mathcal{L}(\mathbf{w}, \mathbf{w}^{(r)}_{\mathtt{ref}}) - \log\left(\frac{P(\mathbf{w}^{(r)}_{\mathtt{ref}}|\mathbf{O}^{(r)}; \hat{\boldsymbol{\theta}}^{(r)}, \boldsymbol{\lambda}, \boldsymbol{\alpha})}{P(\mathbf{w}|\mathbf{O}^{(r)}; \hat{\boldsymbol{\theta}}, \boldsymbol{\lambda}, \boldsymbol{\alpha})}\right)\right\}\right]_{+}$$

where the best alignment $\hat{\boldsymbol{\theta}}^{(r)}$ and $\hat{\boldsymbol{\theta}}$ are the ones that maximise the reference and competing path posterior probabilities. As discussed in [4], introducing a Gaussian prior $P(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\alpha}; 0, C\mathbf{I})$, and substituting the log linear model into the above object function, yields (11). Therefore the SSVM used in this work can also be viewed as a large margin trained log linear model with the "most discriminative" alignment.

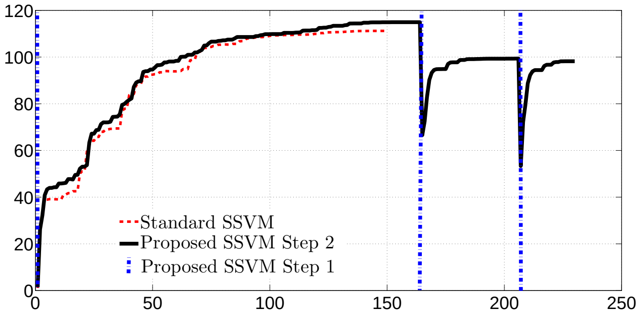

Figure 2: Learning curves for structured SVMs. Dashed curve: standard SSVM, fixed alignments [4]. Vertical dashdotted lines: optimising reference alignments. Solid curve: optimising competing alignments.

4. Noise Robustness

As previously discussed, one of the advantages of using generative models to define the features for the structured SVM is that it is possible to use state-of-the-art model-based noise robustness and speaker adaptation approaches. In this work only noise-robustness is considered. For standard generative models, model-based compensation schemes such as Vector Taylor Series (VTS) compensation [12] are a successful approach to handling this problem. Here, the parameters $\boldsymbol{\lambda}$ associated with the generative model for joint feature space are modified to represent the target acoustic environment [5]. Considering just the static feature vector parameters, the compensated mean and covariance for component m using VTS, are given by

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

where the additive noise mean $\boldsymbol{\mu}_{\text{n}}$ and covariance$\boldsymbol{\Sigma}_{\text{n}}$ are the parameters of the noise model estimated from the data using maximum likelihood estimation [13]. Other terms in above equations include the DCT matrix $\mathbf{C}$ and Jacobian matrix $\mathbf{J}^{(m)}$ are fully described in [12]. Thus in this work discriminative model parameters are noise-independent, whereas the generative model parameters are noise-dependent.

5. Experiments

The performance of the proposed structured SVM was evaluated on the AURORA 2 task. AURORA 2 is a standard small vocabulary digit string recognition task. The vocabulary size M is only 12 (one to nine, plus zero, oh and silence). The utterances in this task are one to seven digits long based on the TIDIGITS database with noise artificially added. The 8440 clean mix-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 in this work. The "simple" back-end was used, thus the HMMs were 16 emitting states whole word digit models, with 3 mixtures per state and silence and inter-word pause models. Test set A was used as the development set for tuning parameters for all systems, such as the penalty factor C for the structured SVMs. All three test sets, A, B and C, were used for final evaluation. The pa-

[1] In theory the previous constraints could be kept, however for implementation simplicity this was not performed.

rameters of SSVM were trained using the same subset of the multi-condition training data as [5]: three of the four subsets (N2-N4) and three of five SNRs (10dB, 15dB, 20dB). This allows direct comparison with the previously published results.

To evaluate the benefit of structured SVMs and optimising the alignment in decoding and training, a range of setups were compared. For all configurations the 12 dimensional feature-space $\boldsymbol{\varphi}^{\text{LL}}$ in (4) was used. 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 binary SVM and multi-class SVMs, and producing the lattices for the structured SVM training and inference [4].

Model	Set A	Set B	Set C	Avg.
HMM	9.84	9.11	9.53	9.49
SVM	9.10	8.68	9.25	8.96
Multi-class SVM	8.27	8.06	8.64	8.26
SSVM ($\hat{\boldsymbol{\theta}}_{\text{hmm}}/\hat{\boldsymbol{\theta}}_{\text{hmm}}$)	7.78	7.31	8.02	7.64
SSVM ($\hat{\boldsymbol{\theta}}_{\text{hmm}}/\boldsymbol{\theta}$)	7.55	7.15	8.01	7.49

Table 1: Average WER (%) in all noise conditions of VTS based HMM, SVM, Multi-class SVM and Structured SVM. For the SSVM $\boldsymbol{\theta}$ indicates optimised alignments, $\hat{\boldsymbol{\theta}}_{\text{hmm}}$ indicates the alignments derived from the HMMs.

Examining the results in Table 1, shows the benefit of using structured SVM over SVM approaches where the observation sequence is segmented into words and individual "segmented" words classified with the SVM (these results are repeated from [4]). The last line shows the performance of optimising the alignment during inference ($\hat{\boldsymbol{\theta}}_{\text{hmm}}/\boldsymbol{\theta}$). Optimising the alignment yields a small gain in performance over using the original alignments ($\hat{\boldsymbol{\theta}}_{\text{hmm}}/\hat{\boldsymbol{\theta}}_{\text{hmm}}$), about 2.0% relative reduction on average.

Model	Train/Test	Set A	Set B	Set C	Avg.
HMM	—	9.84	9.11	9.53	9.49
SSVM (batch)	($\hat{\boldsymbol{\theta}}_{\text{hmm}}/\hat{\boldsymbol{\theta}}_{\text{hmm}}$)	7.89	7.42	8.19	7.76
	($\hat{\boldsymbol{\theta}}_{\text{hmm}}/\boldsymbol{\theta}$)	7.75	7.22	8.02	7.59
	($\boldsymbol{\theta}/\boldsymbol{\theta}$)	**7.56**	**7.14**	**7.77**	**7.43**

Table 2: Average WER (%) among all noise conditions of VTS based HMM and parallel mode Structured SVM, $\boldsymbol{\theta}$ indicates optimised alignments, $\hat{\boldsymbol{\theta}}_{\text{hmm}}$ alignments derived from the HMMs.

To evaluate the impact of optimising the alignment during training, batch-mode training of the SSVM was required due to the computational load. Table 2 shows the performance of these batch-mode systems. The first SSVM system used the HMM alignments for both training and test ($\hat{\boldsymbol{\theta}}_{\text{hmm}}/\hat{\boldsymbol{\theta}}_{\text{hmm}}$). Compared to the equivalent sequential mode update in Table 1 a slight degradation in average performance from 7.64% to 7.76% WER can be seen. Using batch-mode updates allows the joint training of both the alignments and discriminative model parameters. Optimising both the training and inference alignments, ($\boldsymbol{\theta}/\boldsymbol{\theta}$), yielded a 4.3% relative reduction in WER. Just optimising the inference alignment gave 2.1% relative reduction. The overall gain from using the SSVM over the VTS-compensated HMM system was over 20%, though it should be noted that the SVM and SSVM systems made use of a subset of the multi-style training data.

6. Conclusion

This paper has examined the use of structured SVMs for noise robust ASR. One key part of this work, compared to previous work, is that the alignment of frames to labels in the joint feature-space is not fixed. Here the alignment is optimised jointly with the discriminative model parameters. To perform this joint training a number of modifications to the previous published work have been made. First, a Viterbi-style algorithm is described for optimising the alignment based on the SSVM parameters. This algorithm is related to the inference of factorial HMMs. Second, to incorporate the optimal alignments into the training process, the training algorithm is modified making use of the concave-convex optimisation procedure. Finally to reduce the time for jointly training the alignment and discriminative model parameters, a batch-mode training algorithm, where the optimal alignment is optimised using multiple machines, is described. Results on the AURORA 2 task demonstrate that optimising the alignment yields performance gains for both inference and training.

Currently the performance gains from optimising the alignments are small. However this is felt to be due to the use of whole-word models for the AURORA 2 task. Thus the alignment is only defined at the word-level. For medium-to-large vocabulary tasks where the alignment between frames and labels is required at the phone-level, it is expected that optimising the alignment will have a larger impact.

7. References

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

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

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

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

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

[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] 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.

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

[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.