INFINITE STRUCTURED SUPPORT VECTOR MACHINES FOR SPEECH RECOGNITION

J. Yang, R. C. van Dalen, S.-X. Zhang and M. J. F. Gales

Department of Engineering, University of Cambridge, Cambridge, UK

{jy308,rcv25,sxz20,mjfg}@eng.cam.ac.uk

ABSTRACT

Discriminative models, like support vector machines (SVMs), have been successfully applied to speech recognition and improved performance. A Bayesian non-parametric version of the SVM, the infinite SVM, improves on the SVM by allowing more flexible decision boundaries. However, like SVMs, infinite SVMs model each class separately, which restricts them to classifying one word at a time. A generalisation of the SVM is the structured SVM, whose classes can be sequences of words that share parameters. This paper studies a combination of Bayesian non-parametrics and structured models. One specific instance called infinite structured SVM is discussed in detail, which brings the advantages of the infinite SVM to continuous speech recognition.

Index Terms— Bayesian non-parametrics, Dirichlet process, mixture of experts, infinite structured SVM

1. INTRODUCTION

Discriminative models, like support vector machines (SVMs), have been successfully applied to speech recognition. By introducing features in a generative feature space [1] extracted using HMMs, state-of-art speaker adaptation and noise robustness techniques [2] can be used to generate the features.

An SVM by itself uses a linear decision boundary, which may be inappropriate for the feature space. The standard approach is to apply a kernel function. An alternative that does not require choosing a kernel is to use a mixture-of-experts model [3, 4], which employs different classifiers for different regions of space. Normally the optimal number of experts is unknown. In order to sidestep the problem of setting the number of experts, a Bayesian non-parametric framework can be used. By using the infinite SVM (iSVM) [5, 6], better performance is achieved compared with using the SVM, because the iSVM allows SVMs to focus on regions of feature space. Since the interpolation weights for the SVM outputs depends on the location of the data in the feature space, the ensemble decision effectively uses a non-linear decision boundary.

This work was partially supported by EPSRC Project EP/I006583/1 within the Global Uncertainties Programme and DARPA under the RATS program. The paper does not necessarily reflect the position or the policy of US Government and no official endorsement should be inferred. The authors would like to thank Eric Wang and Jeff Chen for their help, and J. Yang would like to thank Cambridge Overseas Trust for partially funding this work.

The SVM and iSVM are unstructured models in that they model each class separately. They have therefore been applied to digit recognition in an acoustic code-breaking [7] setting: first continuous speech is segmented into segments, and then each segment from an HMM recognition pass is rescored independently [6]. In continuous speech recognition (CSR), however, each class is a sentence, and the number of possible classes is unlimited. For example, the possible number of classes for a 6-digit length utterance is 10^6. However, these classes have structure: they share words or phones. The structured SVM (SSVM) [8] was introduced to classify data with structured labels. In [9], the SSVM was successfully used in medium to large vocabulary CSR tasks. In order to apply the mixture-of-experts framework to large vocabulary CSR, the structure must be incorporated into the model.

This paper discusses Bayesian non-parametrics for structured SVMs and introduces the infinite structured SVM (iSSVM) in particular. Rather than using a kernel function in the SSVM [10], which requires a kernel to be chosen, the iSSVM deploys multiple SSVMs to yield a non-linear boundary. The generative feature space already implies a sequence kernel [1]. Though an additional kernel could be used by each expert in the iSSVM, the kernel trick is not considered here.

This paper is organised as follows. The mixture of experts and its infinite version are discussed in Section 2. The SSVM is detailed in Section 3, and the iSSVM is introduced in Section 4. Classification and corresponding issues are discussed in Section 5. Finally, the experimental results and conclusions are given in Section 6.

2. MIXTURE OF EXPERTS

The mixture of experts combines the posteriors from multiple experts focusing on different regions of the feature space, producing a model that can perform more complicated classifications than a single expert can. The framework of the mixture of experts with M experts is illustrated in Fig. 1. The gating network, e.g. a GMM with parameters Θ, uses the input to determine the mixture weights $P(z = m|\boldsymbol{x}, \Theta)$ for each expert m. z is a random variable, the indicator variable, which denotes which expert the input $\boldsymbol{x}$ is associated with. The overall class posteriors are computed with

$$P(w|\boldsymbol{x}, \Theta, H) = \sum_{z \in \mathbb{Z}} P(z|\boldsymbol{x}, \Theta) P(w|\boldsymbol{x}, \boldsymbol{\eta}_z) \qquad (1)$$

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

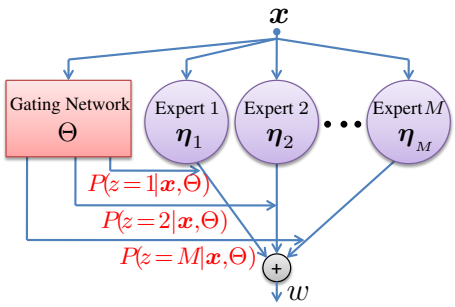

Fig. 1. *The framework of the mixture-of-experts model*

where $\mathbb{Z}$ is the indicator set: $\mathbb{Z} = \{1, \ldots, M\}$. The second term $P(w|\boldsymbol{x}, \boldsymbol{\eta}_z)$ is the zth *expert* with parameter $\boldsymbol{\eta}_z$. H is the parameter set for all the experts, and w is the class label.

If the number of experts in the mixture of expert is set to infinity, $M \to \infty$, and the gating network is given by a Dirichlet process (DP) mixture model [11, 12], the DP mixture of experts [6] can be derived. It has the same form as equation (1), but the indicator set $\mathbb{Z}$ has infinite size: $\mathbb{Z} = \{1, 2, \ldots, \infty\}$. Section 4 discusses a Monte Carlo method to deal with this.

3. STRUCTURED SVM

The structured SVM can be considered as a log-linear model with large-margin training [9]. The log-linear model gives the distribution of word sequence $\mathcal{W}$ and the segmentation ρ given the utterance $\mathcal{O}$:

$$P(\mathcal{W}, \rho|\mathcal{O}, \boldsymbol{\lambda}, \boldsymbol{\eta}) = \frac{\exp\left(\boldsymbol{\eta}^{\mathsf{T}} \Phi(\mathcal{O}, \mathcal{W}; \boldsymbol{\lambda}, \rho)\right)}{\sum_{\mathcal{W}', \rho'} \exp\left(\boldsymbol{\eta}^{\mathsf{T}} \Phi(\mathcal{O}, \mathcal{W}'; \boldsymbol{\lambda}, \rho')\right)} \quad (2)$$

where $\boldsymbol{\lambda}$ indicates the parameters of the generative model, and $\boldsymbol{\eta}$ those of the log-linear model. Given ρ, which is a segmentation into words, the utterance and word sequence can be further described as $\mathcal{O} = \{\boldsymbol{O}_1, \ldots, \boldsymbol{O}_{I_\rho}\}$ and $\mathcal{W} = \{w_1, \ldots, w_{I_\rho}\}$, where $\boldsymbol{O}_i$ is a segment of audio, w_i is the corresponding word of the segment, and I_ρ is the number of segments. $\Phi(\mathcal{O}, \mathcal{W}; \boldsymbol{\lambda}, \rho)$ is the *joint feature space*, which can be set to a sum over segments [13]:

$$\Phi(\mathcal{O}, \mathcal{W}; \boldsymbol{\lambda}, \rho) = \frac{1}{T} \begin{bmatrix} \sum_{i=1}^{I_\rho} \delta(w_i, \tilde{w}_1) \varphi(\boldsymbol{O}_i; \boldsymbol{\lambda}) \\ \vdots \\ \sum_{i=1}^{I_\rho} \delta(w_i, \tilde{w}_L) \varphi(\boldsymbol{O}_i; \boldsymbol{\lambda}) \\ \log\left(P(\mathcal{W})\right) \end{bmatrix} \quad (3)$$

where $\{\tilde{w}_1, \ldots, \tilde{w}_L\}$ are all the unique words, $P(\mathcal{W})$ is given by language model, T is the number of frames in utterance $\mathcal{O}$, which is utilised to normalise the feature space corresponding the utterances with various lengths, and $\varphi(\boldsymbol{O}_i; \boldsymbol{\lambda})$ is the log-likelihood feature vector, which can be described as follows:

$$\varphi(\boldsymbol{O}_i; \boldsymbol{\lambda}) = \begin{bmatrix} \log\left(p(\boldsymbol{O}_i|\boldsymbol{\lambda}_{\tilde{w}_1})\right) \\ \vdots \\ \log\left(p(\boldsymbol{O}_i|\boldsymbol{\lambda}_{\tilde{w}_L})\right) \end{bmatrix}_{L \times 1} \quad (4)$$

In equation (4), $p(\boldsymbol{O}_i|\boldsymbol{\lambda}_{\tilde{w}_l})$ is the likelihood of the HMM corresponding to label $\tilde{w}_l$ given the segment $\boldsymbol{O}_i$.

For large-margin training of the log-linear model, the margin is defined as the log-posterior ratio between the reference $\mathcal{W}_n$ and the most competing hypothesis $\mathcal{W}$. There are N training instances. By introducing the prior $P(\boldsymbol{\eta})$, the training criterion to be minimised can be described as:

$$\sum_{n=1}^{N} \left[\max_{\mathcal{W}, \rho \neq \mathcal{W}_n, \rho_n} \left\{ \mathcal{L}(\mathcal{W}, \mathcal{W}_n) - \log\left(\frac{P(\mathcal{W}_n, \rho_n|\mathcal{O}_n, \boldsymbol{\eta}, \boldsymbol{\lambda})}{P(\mathcal{W}, \rho|\mathcal{O}_n, \boldsymbol{\eta}, \boldsymbol{\lambda})} \right) \right\} \right]_+ - \log P(\boldsymbol{\eta}) \quad (5)$$

where $\mathcal{L}(\mathcal{W}, \mathcal{W}_n)$ is the loss between the hypothesis $\mathcal{W}$ and the reference $\mathcal{W}_n$. The set of possible $\mathcal{W}$ and ρ are obtained from a lattice. When the prior $P(\boldsymbol{\eta})$ is given a Gaussian distribution $P(\boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\eta}; \boldsymbol{\mu}_\eta, \boldsymbol{\Sigma}_\eta)$ with mean $\boldsymbol{\mu}_\eta$ and scaled identity covariance matrix $\boldsymbol{\Sigma}_\eta = C\boldsymbol{I}$, and substituting equation (2) into equation (5), the large-margin training criterion can be further described as follows [9]:

$$\frac{1}{2}||\boldsymbol{\eta} - \boldsymbol{\mu}_{\boldsymbol{\eta}}||^2 + C \sum_{n=1}^{N} \left[\max_{\mathcal{W}, \rho \neq \mathcal{W}_n, \rho_n} \left\{ \boldsymbol{\eta}^{\mathsf{T}} \Phi(\mathcal{O}_n, \mathcal{W}; \boldsymbol{\lambda}, \rho) + \mathcal{L}(\mathcal{W}, \mathcal{W}_n) \right\} - \boldsymbol{\eta}^{\mathsf{T}} \Phi(\mathcal{O}_n, \mathcal{W}_n; \boldsymbol{\lambda}, \rho_n) \right]_+ \quad (6)$$

4. INFINITE STRUCTURED SVM

The Dirichlet process mixture of experts is given in equation (1) with infinite-sized indicator set $\mathbb{Z}$. In order to apply this type of model to continuous speech recognition, structure needs to be incorporated in the model. The direct way to do this is to incorporate the structure into the experts. If each expert is an SSVM described in equation (2), then the DP mixture of experts given in equation (1) becomes:

$$P(\mathcal{W}, \rho|\mathcal{O}, \boldsymbol{\lambda}, \Theta, H) = \sum_{z \in \mathbb{Z}} P(z|\mathcal{O}, \boldsymbol{\lambda}, \Theta) P(\mathcal{W}, \rho|\mathcal{O}, \boldsymbol{\lambda}, \boldsymbol{\eta}_z) \quad (7)$$

Here, the indicator variable z corresponds to the whole utterance $\mathcal{O}$, and the utterance indicator z is a scalar. The resulting model consists of infinite number of SSVMs, and is called the infinite structured SVM (iSSVM). An alternative is to model each word as a mixture of SVMs. The utterance indicator z then becomes a vector, and the resulting model can be called a structured infinite SVM (SiSVM). However, this paper focuses on the infinite structured SVM, as given in (7) with scalar indicator z.

Suppose the training data are $\mathcal{D} = \{\mathcal{O}_1, \ldots, \mathcal{O}_N; \mathcal{W}_1, \ldots, \mathcal{W}_N; \rho_1, \ldots, \rho_N\}$, then classification of the iSSVM can be described as follows:

$$P(\mathcal{W}, \rho|\mathcal{O}, \boldsymbol{\lambda}, \mathcal{D}) = \int P(\mathcal{W}, \rho|\mathcal{O}, \boldsymbol{\lambda}, \mathcal{A}) p(\mathcal{A}|\mathcal{D}) d\mathcal{A} \quad (8)$$

where $\mathcal{A} = \{\Theta, H\}$ are all the parameters of the iSSVM. Since the integral in equation (8) is intractable to compute, a

Monte Carlo method can be applied to approximate this integral. Classification can then be described as:

$$P(\mathcal{W},\rho|\mathcal{O},\boldsymbol{\lambda},\mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K} P(\mathcal{W},\rho|\mathcal{O},\boldsymbol{\lambda},\mathcal{A}^{(k)}) \quad (9)$$

$$= \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M_k} P(z=m|\mathcal{O},\boldsymbol{\lambda},\Theta^{(k)})P(\mathcal{W},\rho|\mathcal{O},\boldsymbol{\lambda},\boldsymbol{\eta}_m^{(k)})$$

where $\mathcal{A}^{(k)}$ are sampled from the model posterior distribution $p(\mathcal{A}|\mathcal{D})$. Here, K samples are used to approximate this intractable integral. Since $\mathcal{A}$ is the whole parameter set of the iSSVM, the joint posterior distribution $p(\mathcal{A}|\mathcal{D})$ does not have a closed form. Thus, Gibbs sampling [14] is used to obtain samples from this joint posterior distribution. In sampling, the auxiliary variables $\boldsymbol{z} = \{z_1, \ldots, z_N\}$ (which are the indicator variables for the training data) are introduced. The samples $\mathcal{A}^{(k)}$ are obtained by sampling from $p(\mathcal{A}, \boldsymbol{z}|\boldsymbol{\lambda}, \mathcal{D})$, yielding $\{\mathcal{A}^{(k)}, \boldsymbol{z}^{(k)}\}$. $\mathcal{A}^{(k)}$ can be considered as being sampled from $p(\mathcal{A}|\boldsymbol{\lambda}, \mathcal{D})$ [15]. M_k is the number of unique values of the sampled indicators $\boldsymbol{z}^{(k)}$.

Θ is the parameter set of the gating network which here is a DP mixture model. The conditional posterior distribution of the parameter set can be described as follows:

$$p(\Theta|H^{(k)}, \boldsymbol{z}^{(k)}, \boldsymbol{\lambda}, \mathcal{D}) = p(\Theta|\{\phi(\mathcal{O}_n;\boldsymbol{\lambda})\}_{n=1}^{N}, \boldsymbol{z}^{(k)}) \quad (10)$$

where $\phi(\mathcal{O}_n, \boldsymbol{\lambda})$ is the feature space for the utterance $\mathcal{O}_n$, which maps the observation $\mathcal{O}_n$ to a space with fixed dimension. The feature is the log-likelihood feature of the whole utterance. Here, the normalised features based on segments are used: $\phi(\mathcal{O}_n, \boldsymbol{\lambda}) = (1/T_n)\sum_i \varphi(\boldsymbol{O}_i, \boldsymbol{\lambda})$, where T_n is number of frames in utterance $\mathcal{O}_n$, and $\varphi(\boldsymbol{O}_i, \boldsymbol{\lambda})$ is the log-likelihood feature described in equation (4). Given the features $\{\phi(\mathcal{O}_n;\boldsymbol{\lambda})\}_{n=1}^{N}$ and corresponding indicators $\boldsymbol{z}^{(k)}$, $\Theta^{(k)}$ can be sampled through the methods described in [11, 12].

In terms of the parameters of the experts H, each expert is a log-linear model with large margin training, so the parameter of the m^{th} expert $\boldsymbol{\eta}_m$ is obtained through equation (6) with the data associated with expert m. If there are few observations associated with an expert, generalisation can become a problem. Thus, each expert uses an informative prior. Similar to the method used in [6], the mean of the prior $\boldsymbol{\mu}_{\boldsymbol{\eta}}$ is obtained from the SSVM trained on the whole training set. By introducing this mean, the iSSVM should recover the performance of the SSVM, if C is small enough (i.e. if the variance is small enough). Better performance could be achieved by gradually increasing C. The 1-slack cutting plane algorithm [16] is used to train the SSVM. The constraint set that this algorithm uses for training the current SSVM parameter $\boldsymbol{\eta}_m^{(k)}$ can be cached and propagate to the next iteration of obtaining $\boldsymbol{\eta}_m^{(k+1)}$. This caching method can make the training more efficient, especially when applying the iSSVM to large vocabulary CSR.

The indicator variable z_n is sampled according to the following posterior distribution:

$$P(z_n = m|\mathcal{A}^{(k)}, \boldsymbol{z}_{-n}, \boldsymbol{\lambda}, \mathcal{D}) \propto \quad (11)$$
$$P(z_n = m|\boldsymbol{z}_{-n}, \alpha)p\big(\phi(\mathcal{O}_n, \boldsymbol{\lambda})|\boldsymbol{\theta}_m^{(k)}\big)P(\mathcal{W}_n, \rho_n|\mathcal{O}_n, \boldsymbol{\lambda}, \boldsymbol{\eta}_m^{(k)})$$

where $\boldsymbol{z}_{-n}$ denotes all the indicators except z_n. The first term $P(z_n = m|\boldsymbol{z}_{-n}, \alpha)$ is given by the *Chinese Restaurant Process* (CRP) with concentration parameter α [17]. The term $P(\mathcal{W}_n, \rho_n|\mathcal{O}_n, \boldsymbol{\lambda}, \boldsymbol{\eta}_m^{(k)})$ is the posterior distribution given by the log-linear model described in equation (2), and term $p\big(\phi(\mathcal{O}_n, \boldsymbol{\lambda})|\boldsymbol{\theta}_m^{(k)}\big)$ is the component likelihood. When z_i indicates an existing expert, it is straightforward to calculate the conditional posterior distribution of z_n. When z_n denotes a new expert, following the method introduced in [11], in calculating the likelihood $p\big(\phi(\mathcal{O}_n, \boldsymbol{\lambda})|\boldsymbol{\theta}\big)$, the parameter $\boldsymbol{\theta}$ is sampled from its prior distribution as an auxiliary parameter, then the likelihood can be easily obtained. In order to make the newly generated expert have good generalisation, in calculating the third term, the parameter for the expert $\boldsymbol{\eta}$ is given as the the mean of its prior, namely the optimised parameter of the SSVM trained on the whole training set.

5. CLASSIFICATION

The equation used for classification has been given in equation (9). By substituting the log-linear model given in equation (2) into (9), it becomes:

$$P(\mathcal{W}, \rho|\mathcal{O}, \boldsymbol{\lambda}, \mathcal{D}) \quad (12)$$
$$\approx \frac{1}{K}\sum_{k=1}^{K}\sum_{m=1}^{M_k} P(z_k = m|\mathcal{O}, \boldsymbol{\lambda}, \Theta^{(k)})\frac{\exp\big(\boldsymbol{\eta}_m^{(k)\mathsf{T}}\Phi(\mathcal{O}, \mathcal{W}; \boldsymbol{\lambda}, \rho)\big)}{\mathcal{S}_m^k}$$

where all possible $(\mathcal{W}, \rho)$ are obtained from a lattice, and $\mathcal{S}_m^k$ is the normalisation term:

$$\mathcal{S}_m^k = \sum_{\mathcal{W}', \rho'} \exp\big(\boldsymbol{\eta}_m^{(k)\mathsf{T}}\Phi(\mathcal{O}, \mathcal{W}'; \boldsymbol{\lambda}, \rho')\big) \quad (13)$$

In the SSVM, this term can be ignored, since no posterior needs to be calculated and the normalisation term stays the same for all possible labels. For the iSVM on the other hand, the posterior, given by the log-linear model, does need to be calculated. Thus, this term cannot be ignored, but it is trivial to calculate, since the possible number of labels are small for each segment. In the iSSVM, the calculation of this term $\mathcal{S}_m^k$ is nontrivial, since the possible number of labels are exponentially large for the utterance $\mathcal{O}$.

Given that the possible number of labels are extremely large, the summation in equation (13) is quite inefficient. The forward algorithm can be adopted to calculate this summation efficiently on the lattice. According to the definition of the joint feature space given in equation (3), and describing the parameter of the m^{th} SSVM as $\boldsymbol{\eta}_m^{(k)} =$

System	Features	Test Set WER(%)			Avg
		testa	testb	testc	
HMM	MFCC	9.83	9.11	9.53	9.48
SVM	Log-Like	8.29	7.90	8.61	8.20
iSVM		8.25	7.87	8.53	8.15
SSVM	Joint Feat	7.78	7.29	7.98	7.63
iSSVM		7.60	7.25	7.77	7.49

Table 1. The results on Aurora 2 database

$\{\boldsymbol{\eta}_{m,\tilde{w}_1}^{(k)\mathsf{T}}, \ldots, \boldsymbol{\eta}_{m,\tilde{w}_L}^{(k)\mathsf{T}}, \eta_{m,\mathcal{W}}^{(k)}\}^{\mathsf{T}}$, the dot product can be performed for each arc instead of for the whole utterance, so that the normalisation term in equation (13) becomes:

$$\mathcal{S}_m^k = \sum_{\mathcal{W}',\rho'} \exp\Big(\frac{1}{T}\Big[\sum_{i=1}^{I_{\rho'}} \boldsymbol{\eta}_{m,w_i}^{(k)\mathsf{T}} \varphi(\boldsymbol{O}_i;\boldsymbol{\lambda}) + \eta_{m,\mathcal{W}}^{(k)} \log P(\mathcal{W})\Big]\Big)$$
$$= \sum_{\mathcal{W}',\rho'} \Big[P(\mathcal{W})^{\frac{\eta_{m,\mathcal{W}}^{(k)}}{T}} \prod_{i=1}^{I_{\rho'}} \exp\big(\frac{1}{T}\boldsymbol{\eta}_{m,w_i}^{(k)\mathsf{T}} \varphi(\boldsymbol{O}_i;\boldsymbol{\lambda})\big)\Big] \quad (14)$$

Again, T is the number of frames in utterance $\mathcal{O}$, and $I_{\rho'}$ is the number of segments given the segmentation ρ', which is one path in the lattice. $P(\mathcal{W})$ is the probability of the word sequence. If the bigram language model is used, the probability can be described as $P(\mathcal{W}) = \prod_{i=1}^{I_{\rho'}} P(w_i|w_{i-1})$, and here $P(w_1|w_0)$ is defined as $P(w_1|w_0) = P(w_1)$. Then, equation (14) can be further described as follows:

$$\mathcal{S}_m^k = \sum_{\mathcal{W}',\rho'} \Big\{\prod_{i=1}^{I_{\rho'}} \Big[P(w_i|w_{i-1})^{\frac{\eta_{m,\mathcal{W}}^{(k)}}{T}} \exp\big(\frac{1}{T}\boldsymbol{\eta}_{m,w_i}^{(k)\mathsf{T}} \varphi(\boldsymbol{O}_i;\boldsymbol{\lambda})\big)\Big]\Big\} \quad (15)$$

Because the term inside the product is now a scalar, the forward algorithm can be applied to calculate this summation on the lattice. The forward algorithm is discussed in [18]. At each node in the lattice, the scores are merged. Thus, $\mathcal{S}_m^k$ can be calculated in $\mathrm{O}(N_{\text{arc}}L)$ time, where N_{arc} is number of arcs in the lattice, and L is the unique number of segment labels.

In classification, the best hypothesis $(\mathcal{W}, \rho)$ needs to be found through equation (12). But, finding a path ρ and corresponding $\mathcal{W}$, that maximises the posterior $P(\mathcal{W}, \rho|\mathcal{O}, \boldsymbol{\lambda}, \mathcal{D})$, might be a problem here. In the structured SVM, the Viterbi algorithm is applied to search the best hypothesis $(\mathcal{W}, \rho)$ [9]. However, in the iSSVM, the parameter $\boldsymbol{\eta}_m^{(k)}$ varies with experts m and samples k. Only when m and k are given can the Viterbi algorithm be applied. Here, however, the summation over m and k in equation (12) makes it impossible to use the Viterbi algorithm. Instead of enumerating all possible $(\mathcal{W}, \rho)$, this exponentially large set is approximated. The candidate set $\mathbb{P}$ is constructed with the N best hypotheses[1] from each expert separately, which it is possible to find using the Viterbi algorithm. After the set $\mathbb{P}$ is obtained, equation (12) can be used in classification as follows:

$$(\mathcal{W}, \rho) \approx \arg\max_{\mathcal{W},\rho\in\mathbb{P}} P(\mathcal{W}, \rho|\mathcal{O}, \boldsymbol{\lambda}, \mathcal{D}) \quad (16)$$

[1]Here, only the 1-best hypothesis is considered.

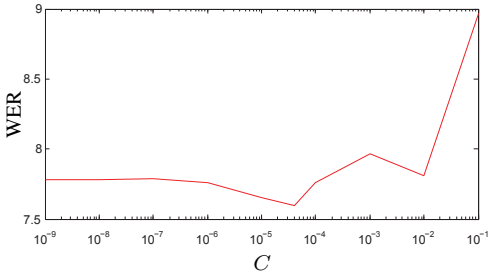

Fig. 2. The iSSVM performance on set A with different C

6. EXPERIMENTS AND CONCLUSIONS

The performance of the proposed iSSVM is evaluated on the Aurora 2 database [19]. The utterances in this database are continuous digit strings with vocabulary size 12 (one to nine, plus zero, oh and silence). The generative models (HMMs) are trained on the clean data with 8840 utterances. The noise model for VTS compensation [20] is estimated on each utterance, and the performance of the VTS compensated HMM (which is used to obtain log-likelihood features for the discriminative models in the experiments) is listed in Table 1. The SVM, iSVM, SSVM and iSSVM are trained on a subset of the multi-style training data containing 3 noise conditions (N2, N3 and N4) and 3 SNRs (20dB, 15dB and 10dB). All 3 test databases, A, B and C with numbers of utterances 20020, 20020 and 10010 respectively, are used in the evaluation.

In the experiments, the log-likelihood features described in equation (4) are used by the SVM and iSVM, and the joint features described in equation (3) are used by the SSVM and iSSVM. All experiments are conducted with the number of samples $K = 10$. The results are listed in Table 1. On test set A and C, the iSSVM get around 3% relative improvement in all SNRs, but only a small improvement is achieved on test set B. The large margin training criterion described in equation (6) is adopted to train the experts (SSVM) of the iSSVM, and different experts share the same C. The parameter C is tuned on test set A, with word error rates illustrated in Fig. 2. Since the prior mean $\boldsymbol{\mu}_{\boldsymbol{\eta}}$ in equation (6) is given the optimised parameter of the SSVM trained on the whole training set, when C is small, the SSVM performance is recovered, and the optimised C can be found by increasing C.

This paper studies the combination of Bayesian nonparametrics with structured models. Specifically, the infinite structured SVM is detailed, which is an extension of the iSVM described in previous paper [6]. Taking advantage of the infinite mixture of experts that are structured models, the iSSVM outperforms the iSVM and SSVM. As discussed in Section 4, the indicator variable of the iSSVM is a scalar, which means all the segments in an utterance share the same indicator. This might limit the flexibility of the gating network. In order to make better use of the data, a more granular (vector) indicator could be introduced. Thus, future work will study a structured model with a mixture of experts for each word, the structured infinite SVM (SiSVM).

7. REFERENCES

[1] Martin Layton, *Augmented Statistical Models for Classifying Sequence Data*, Ph.D. thesis, Cambridge University, 2006.

[2] Yongqiang Wang and Mark Gales, "Speaker and noise factorization for robust speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 7, pp. 2149–2158, 2012.

[3] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[4] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] Jun Zhu, Ning Chen, and Eric Xing, "Infinite SVM: a Dirichlet process mixture of large-margin kernel machines," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, New York, NY, USA, June 2011, pp. 617–624, ACM.

[6] Jingzhou Yang, Rogier C. van Dalen, and Mark Gales, "Infinite support vector machines in speech recognition," in *Proceedings of Interspeech*, Lyon, France, 2013, pp. 3303–3307.

[7] Veera Venkataramani, Shantanu Chakrabartty, and William Byrne, "Support vector machines for segmental minimum bayes risk decoding of continuous speech," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2003, pp. 13–18.

[8] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the twenty-first international conference on Machine learning (ICML)*, New York, NY, USA, 2004, pp. 104–111.

[9] Shi-Xiong Zhang and Mark Gales, "Structured SVMs for automatic speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, pp. 544–555, 2013.

[10] Shi-Xiong Zhang and Mark Gales, "Kernelized log linear models for continuous speech recognition," in *ICASSP*, 2013, pp. 6950–6954.

[11] Radford M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of computational and graphical statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[12] Carl Edward Rasmussen and Zoubin Ghahramani, "Infinite mixtures of Gaussian process experts," in *NIPS*, 2001, pp. 881–888.

[13] Shi-Xiong Zhang and Mark Gales, "Structured support vector machines for noise robust continuous speech recognition," in *Proceedings of Interspeech*, Florence, Italy, 2011, pp. 989–992.

[14] David Wingate, "Markov chain Monte Carlo and Gibbs sampling," 2004.

[15] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I. Jordan, "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, pp. 5–43, 2003.

[16] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[17] Erik B. Sudderth, *Graphical Models for Visual Object Recognition and Tracking*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.

[18] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken language processing*, Prentice Hall PTR New Jersey, 2001.

[19] David Pearce and Hans-Günter Hirsch, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proceedings of the 6th International Conference on Spoken Language Processing*, 2000, pp. 29–32.

[20] Alex Acero, Li Deng, Trausti Kristjansson, and Jerry Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proceedings of ICSLP 2000*, Beijing, China, 2000, pp. 869–872.