Structured Support Vector Machines for Continuous Speech Recognition

Austin Zhang and Mark Gales

1 March 2011

Cambridge University Engineering Department

Toshiba Presentation, March 2011
Overview

• SVMs for multi-class sequential data
  – Segment-level Feature Space
  – Combining binary SVMs, Multi-Class SVMs

• Structured SVMs
  – Sentence-level Joint Feature Space
  – relationship to Large Margin Log Linear Models

• Optimal Segmentation for Structured SVMs
  – Non-convex Optimization
  – Inference problems, relationship with other models (Factorial HMMs)
  – Results

• Implementation for Large Vocabulary Speech Recognition
  – Parallelization of Structured SVMs,
  – Results
Segment-level Feature Space

- Feature spaces based on **generative** models

\[
\varphi^{LL}(O; \lambda) = \begin{bmatrix}
\log \left( p(O; \lambda^{(\omega_1)}) \right) \\
\log \left( p(O; \lambda^{(\omega_2)}) \right) \\
\vdots \\
\log \left( p(O; \lambda^{(\omega_M)}) \right)
\end{bmatrix}, \quad \varphi^{LL}(O; \lambda) = \begin{bmatrix}
\log \left( p(O; \lambda^{(a-a+c)}) \right) \\
\log \left( p(O; \lambda^{(a-b+c)}) \right) \\
\vdots \\
\log \left( p(O; \lambda^{(a-z+c)}) \right)
\end{bmatrix}
\]

- \( \varphi^{\nabla} \) has additional discriminative information, but with very high dimension

- **Generative model** to extract features + **Discriminative model** to classify data.

  Three Advantages:

  - Elegant way of combining generative and discriminative models
  - State-of-the-art model-based compensation can be easily applied
  - Enable efficient decoding with optimal segmentation (New!)
Multi-Class SVMs for ASR

- Two ways to extend SVMs to multi-class classification
  - combining binary SVMs: 1-v-1 voting - $M(M - 1)/2$ classifiers
  - multi-class SVM: modifying the SVM training - $M$ set of parameters

- Using HMM-based hypothesis
  - “force-align” - word start/end

- Foreach segment
  - find “best” digit + silence

- Foreach segment, decoding
  - binary SVMs combination - 1-v-1 voting
  - Multi-Class SVMs
    - $\hat{\omega} = \arg \max_{\omega \in \{\text{ONE}, \ldots, \text{SIL}\}} \alpha^{(\omega)^T} \varphi(O)$
Multi-Class SVMs Results

- Multi-Class SVM Results on AURORA 2

<table>
<thead>
<tr>
<th>Model</th>
<th>Features</th>
<th>Dim</th>
<th>Classifiers</th>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>9.84</td>
<td>9.11</td>
<td>9.53</td>
</tr>
<tr>
<td>SVM</td>
<td>$\phi_{LL}(O; \lambda)$</td>
<td>12</td>
<td>66</td>
<td>9.10</td>
<td>8.68</td>
<td>9.25</td>
</tr>
<tr>
<td>MSVM</td>
<td>$\phi_{LL}(O; \lambda)$</td>
<td>12</td>
<td>12</td>
<td>8.27</td>
<td>8.06</td>
<td>8.64</td>
</tr>
</tbody>
</table>

- Limitations of multi-class SVMs and combining binary SVMs
  - Restrict to one alignment generated by HMMs
  - Each segment is treated independently

- How to model the whole continuous speech?
Sentence-level Joint Feature Spaces

- Continuous speech: variable-length input $O$, complex structures in output $w$

- Joint feature space $\phi(O, w; \lambda)$ modelling the structures between $O$ and $w$

- How to construct a joint feature space?
  - structures must be introduced - decomposed to structure unit level,

\[
\phi(O, w; \hat{\theta}, \lambda) = \frac{1}{T} \sum_i \delta(w_i - \text{ONE}) \begin{bmatrix} \phi^{\text{LL}}(O_t(w_i; \hat{\theta}); \lambda) \\ \phi^{\text{LL}}(O_t(w_i; \hat{\theta}); \lambda) \\ \vdots \\ \phi^{\text{LL}}(O_t(w_i; \hat{\theta}); \lambda) \end{bmatrix}
\]
Structured SVMs for ASR

- Learning an unique $\alpha$ and decoding by match score:

$$\arg \max_{w \in \mathbb{L}} \alpha^T \phi(O, w; \hat{\theta}, \lambda)$$

$\hat{\theta}$ is Viterbi alignment

- Training $\alpha$ so that $\alpha^T \phi(O, w; \hat{\theta}, \lambda)$ is max for correct $w$ (Hard-margin case):

$$\min_{\alpha} \frac{1}{2} ||\alpha||^2$$
$$\text{s.t. } \alpha^T \phi(\text{"A B C"}, \text{"A A A"}) \geq \alpha^T \phi(\text{"A A A"}) + 1, \quad \text{a lot :}$$
$$\alpha^T \phi(\text{"A B C"}, \text{"Z Z Z"}) \geq \alpha^T \phi(\text{"Z Z Z"}) + 1, \quad \text{a lot :}$$
$$\vdots$$
$$\alpha^T \phi(\text{"X Y Z"}, \text{"A A A"}) \geq \alpha^T \phi(\text{"A A A"}) + 1, \quad \text{a lot :}$$
$$\text{Training Sample 1}
\text{"A B C"}$$
$$\alpha^T \phi(\text{"X Y Z"}, \text{"Z Z Z"}) \geq \alpha^T \phi(\text{"Z Z Z"}) + 1, \quad \text{a lot :}$$
$$\text{Training Sample R}
\text{"X Y Z"}$$
Relationship with Log Linear Model

• Example of log linear models

\[ P(w|O; \alpha, \lambda) = \frac{1}{Z(O; \alpha, \lambda)} \exp (\alpha^T \phi(O, w; \theta, \lambda)) \]

• Require log-posterior-ratio

\[
\min_{w \neq w_{\text{ref}}} \left\{ \log \left( \frac{P(w_{\text{ref}}|O; \lambda, \alpha)}{P(w|O; \lambda, \alpha)} \right) \right\}
\]

to be beyond margin.

• General form of regularised LM criterion - minimise \( \mathcal{F}_{\text{LM}}(\alpha) \)

\[
- \log(P(\alpha)) + \frac{1}{R} \sum_{r=1}^{R} \left[ \max_{w \neq w_{\text{ref}}}^{(r)} \left\{ \mathcal{L}(w, w_{\text{ref}}^{(r)}) - \log \left( \frac{P(w_{\text{ref}}^{(r)}|O^{(r)}; \lambda, \alpha)}{P(w|O^{(r)}; \lambda, \alpha)} \right) \right\} \right] +
\]
Large Margin LLM ≡ Structured SVM

- $P(\alpha)$ - Gaussian priors $\mathcal{N}(0, C\mathbf{1})$, substituting the log linear model - minimise

$$
F_{\text{LM}}(\alpha) = \frac{1}{2}\|\alpha\|^2 + \frac{C}{R} \sum_{r=1}^{R} \left[ -\alpha^T \phi(O^{(r)}, w^{(r)}_{\text{ref}}; \hat{\theta}(r), \lambda) \\
+ \max_{w \neq w^{(r)}_{\text{ref}}} \left\{ \mathcal{L}(w, w^{(r)}_{\text{ref}}) + \alpha^T \phi(O^{(r)}, w; \hat{\theta}, \lambda) \right\} \right]$$

- Training of Structured SVM

$$
\min_{\alpha, \xi} \frac{1}{2}\|\alpha\|^2 + \frac{C}{R} \sum_{r=1}^{R} \xi_r \\
\text{s.t.} \quad \alpha^T \phi(O^{(1)}, w^{(1)}_{\text{ref}}; \hat{\theta}^{(1)}) \geq \max_{w \neq w^{(1)}_{\text{ref}}} \left\{ \mathcal{L}(w^{(1)}_{\text{ref}}, w) + \alpha^T \phi(O^{(1)}, w; \hat{\theta}) \right\} - \xi_1, \\
\vdots \quad \vdots \quad \vdots \\
\alpha^T \phi(O^{(R)}, w^{(R)}_{\text{ref}}; \hat{\theta}^{(R)}) \geq \max_{w \neq w^{(R)}_{\text{ref}}} \left\{ \mathcal{L}(w^{(R)}_{\text{ref}}, w) + \alpha^T \phi(O^{(R)}, w; \hat{\theta}) \right\} - \xi_R, \\
\xi_r \geq 0, \quad r = 1, \ldots, R.
$$
Review Structured SVMs Results

• Structured SVM Results on AURORA 2

<table>
<thead>
<tr>
<th>Model</th>
<th>Features</th>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>—</td>
<td>9.84</td>
<td>9.11</td>
<td>9.53</td>
</tr>
<tr>
<td>SVM</td>
<td>$\phi^{\text{LL}}(O; \lambda)$</td>
<td>9.10</td>
<td>8.68</td>
<td>9.25</td>
</tr>
<tr>
<td>MSVM</td>
<td>$\phi^{\text{LL}}(O; \lambda)$</td>
<td>8.27</td>
<td>8.06</td>
<td>8.64</td>
</tr>
<tr>
<td>SSVM</td>
<td>$\phi(O, w; \hat{\theta}, \lambda)$</td>
<td>7.78</td>
<td>7.31</td>
<td>8.02</td>
</tr>
</tbody>
</table>

• Promising results, but could be better.
  Consider extending previous framework for ASR to handle latent variable $\theta$. 
Optimal State Sequence

• Optimal State Sequence
  – Previously, the alignment $\hat{\theta}$ is pre-fixed by HMM Viterbi likelihood.
  – How to find “optimal” alignment for current discriminative model, e.g., Log Linear Models or Structured SVMs?

• Several interesting questions to consider:
  – Loss function?
  – Non-convex training objective? (segmentation is unknown during training)
  – Decoding problems?

• loss function: $\mathcal{L}(w, w_{\text{ref}}^{(r)}) \rightarrow \mathcal{L}((w, \theta), (w_{\text{ref}}^{(r)}, \theta^{(r)}))$
  – assume $\mathcal{L}(w, w_{\text{ref}}^{(r)})$ does not depend on the alignment $\theta^{(r)}$ of reference, using loss $\mathcal{L}((w, \theta), w_{\text{ref}}^{(r)})$. (Explain later)
Training with Optimal State Sequence

• Previously, the “most likely” state sequence $\hat{\theta}(r)$ is given by HMM Viterbi likelihood (Convex Optimization)

$$F_{LM}(\alpha) = \frac{1}{2} ||\alpha||^2_2 + \frac{C}{R} \sum_{r=1}^{R} \left[ -\alpha^T \phi(O^{(r)}, w^{(r)}_{ref}, \hat{\theta}(r), \lambda) ight]$$

$$+ \max_{w \neq w_{ref}} \left\{ L(w, w^{(r)}_{ref}) + \alpha^T \phi(O^{(r)}, w; \hat{\theta}, \lambda) \right\}$$

• Now, Optimizing state sequence $\theta$ with $\alpha$ during the Large Margin training (Nonconvex Optimization)

$$F_{LM}(\alpha) = \frac{1}{2} ||\alpha||^2_2 + \frac{C}{R} \sum_{r=1}^{R} \left[ -\max_{\theta^{(r)}} \left( \alpha^T \phi(O^{(r)}, w^{(r)}_{ref}, \theta^{(r)}) \right) \right]$$

$$+ \max_{w \neq w_{ref}, \theta} \left\{ L((w, \theta), w^{(r)}_{ref}) + \alpha^T \phi(O^{(r)}, w, \theta) \right\}$$
Training with Optimal State Sequence

- Previously, the “most likely” state sequence $\hat{\theta}^{(r)}$ is given by HMM Viterbi likelihood (Convex Optimization)

$$
\mathcal{F}_{LM}(\alpha) = \frac{1}{2}||\alpha||^2_2 + \frac{C}{R} \sum_{r=1}^{R} \left[ -\alpha^T \phi(O^{(r)}, w^{(r)}_{ref}, \hat{\theta}^{(r)}, \lambda) \right] \\
+ \max_{w \neq w_{ref}} \left\{ \mathcal{L}(w, w^{(r)}_{ref}) + \alpha^T \phi(O^{(r)}, w; \hat{\theta}, \lambda) \right\}
$$

- Convex optimization
Nonconvex optimization

- training $\alpha$ with the optimal alignment $\theta$ using Concave-Convex Procedure (CCCP)
Decoding with Optimal State Sequence

- Decoding with optimal state sequence $\theta$

$$\arg\max_{w \in \mathbb{L}} \alpha^T \phi(O, w; \hat{\theta}, \lambda) \quad \rightarrow \quad \arg\max_{w, \theta} \alpha^T \phi(O, w, \theta; \lambda)$$

- Three related inference problems (Two in training, One in decoding)
  - Alignment: $\max_{\theta^{(r)}} \left( \alpha^T \phi(O^{(r)}, w_{ref}^{(r)}, \theta^{(r)}) \right)$
  - Loss-augmented: $\max_{w \neq w_{ref}, \theta} \left\{ L((w, \theta), w_{ref}^{(r)}) + \alpha^T \phi(O^{(r)}, w, \theta) \right\}$
  - Decoding: $\arg\max_{w, \theta} \alpha^T \phi(O, w, \theta; \lambda)$

- Based on the feature space we have, how to solve these inference problems. Note that all the segments in the lattices are HMM Viterbi alignment not “optimal”.
Decoding with Optimal State Sequence

- Decoding without latent variable:
  \[
  \arg \max_w \left( \alpha^T \phi(O, w, \hat{\theta}; \lambda) \right) = \\
  \arg \max_w \sum_{w_i} \alpha^{(w_i)}^T \varphi^{LL}(O; \hat{\theta}, \lambda)
  \]

- Decoding with optimal latent variable:
  \[
  \arg \max_{w, \theta} \left( \alpha^T \phi(O, w, \theta; \lambda) \right) = \\
  \arg \max_{w, \theta} \sum_{w_i} \alpha^{(w_i)}^T \varphi^{LL}(O; \theta, \lambda)
  \]

- Related to Factorial HMM inference
Results of Optimizing Alignment in Decoding

- Decoding with HMM Viterbi $\hat{\theta}$:

$$\arg\max_w \left( \alpha^T \phi(O, w, \hat{\theta}; \lambda) \right)$$

- Decoding with Optimal Alignment $\theta$:

$$\arg\max_{w, \theta} \left( \alpha^T \phi(O, w, \theta; \lambda) \right)$$

<table>
<thead>
<tr>
<th>Model</th>
<th>$\theta$ in training</th>
<th>$\theta$ in decoding</th>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>—</td>
<td>—</td>
<td>9.84</td>
<td>9.11</td>
<td>9.53</td>
</tr>
<tr>
<td>SSVM</td>
<td>HMM Viterbi $\hat{\theta}$</td>
<td>HMM Viterbi $\hat{\theta}$</td>
<td>7.78</td>
<td>7.31</td>
<td>8.02</td>
</tr>
<tr>
<td>SSVM</td>
<td>HMM Viterbi $\hat{\theta}$</td>
<td>Optimal $\theta$</td>
<td>7.55</td>
<td>7.15</td>
<td>8.00</td>
</tr>
</tbody>
</table>
Scaling to Large Vocabulary Speech Recognition

- Handling high-dimension sparse joint feature space using “alpha tying”

- **Batch Updates.** Structured SVMs recompute the solution of the QP sub-problem after each update to the training set. This allows the algorithm to potentially find better constraints to be added in next step, but it requires a lot of time on the QP solver.

  More efficient to use batch updates and restarting the QP sub-problem solver from the previous solution.

- **SSVM Parallelization.** Most of overall runtime is spent on the finding the most competing hypothesis $\max_{w \neq w_{ref}, \theta} \left\{ \mathcal{L}(w, \theta, w_{ref}^{(r)}) + \alpha^T \phi(O^{(r)}, w, \theta) \right\}$, parallelizing this process will lead to a substantial speed-up.
### Structured SVM Results

- Results of batch updates

<table>
<thead>
<tr>
<th>Model</th>
<th>θ in training</th>
<th>θ in decoding</th>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>—</td>
<td>—</td>
<td>9.84</td>
<td>9.11</td>
<td>9.53</td>
</tr>
<tr>
<td>SSVM</td>
<td>HMM Viterbi ̂θ</td>
<td>HMM Viterbi ̂θ</td>
<td>7.78</td>
<td>7.31</td>
<td>8.02</td>
</tr>
<tr>
<td>SSVM (batch)</td>
<td>HMM Viterbi ̂θ</td>
<td>HMM Viterbi ̂θ</td>
<td>7.89</td>
<td>7.42</td>
<td>8.19</td>
</tr>
</tbody>
</table>

- Results of optimal state sequence θ

<table>
<thead>
<tr>
<th>Model</th>
<th>θ in training</th>
<th>θ in decoding</th>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>—</td>
<td>—</td>
<td>9.84</td>
<td>9.11</td>
<td>9.53</td>
</tr>
<tr>
<td>SSVM (batch)</td>
<td>HMM Viterbi ̂θ</td>
<td>HMM Viterbi ̂θ</td>
<td>7.89</td>
<td>7.42</td>
<td>8.19</td>
</tr>
<tr>
<td>SSVM (batch)</td>
<td>HMM Viterbi ̂θ</td>
<td>Optimal θ</td>
<td>7.75</td>
<td>7.22</td>
<td>8.02</td>
</tr>
<tr>
<td>Latent SSVM (batch)</td>
<td>Optimal θ</td>
<td>Optimal θ</td>
<td>7.64</td>
<td>7.11</td>
<td>7.77</td>
</tr>
</tbody>
</table>
Conclusion

- **Optimizing the segmentation** during large margin training and decoding gives us small but consistency better performance in all test sets. Compare with structured SVM batch updates mode, the relative improvement of latent SSVM is around 4%.

- Although some performance reduction has been observed in batch updates mode, **Structured SVMs parallelization** significantly speed up the training. This make the application to large vocabulary system become possible.

- **Large vocabulary systems** have more segments which potentially more need optimizing state sequences, therefore we applied latent structured SVM to AURORA4. The result is coming soon.

- **Future Work**: Finish experiments on AURORA4; Joint large margin training for generative models and discriminative models; Learning discriminative joint feature space for LVCSR.
**Complementary**

- Summary of training process for structured SVMs.

- Summary of training process for latent structured SVMs.

- Joint Large Margin Training for generative and discriminative models.
• Decoding with optimal alignment.