Structured Discriminative Models for Speech Recognition

Mark Gales - work with Anton Ragni, Austin Zhang, Rogier van Dalen

April 2012

Cambridge University Engineering Department

NTT Visit

Overview

- Acoustic Models for Speech Recognition
  - dependency modelling
  - generative and discriminative models
- Sequence (dynamic) kernels
  - discrete and continuous observation forms
- Combining Generative and Discriminative Models
  - generative score-spaces and log-linear models
- Training Criteria
  - large-margin-based training
- Initial Evaluation
  - AURORA-2 and AURORA-4 experimental results

Acoustic Models

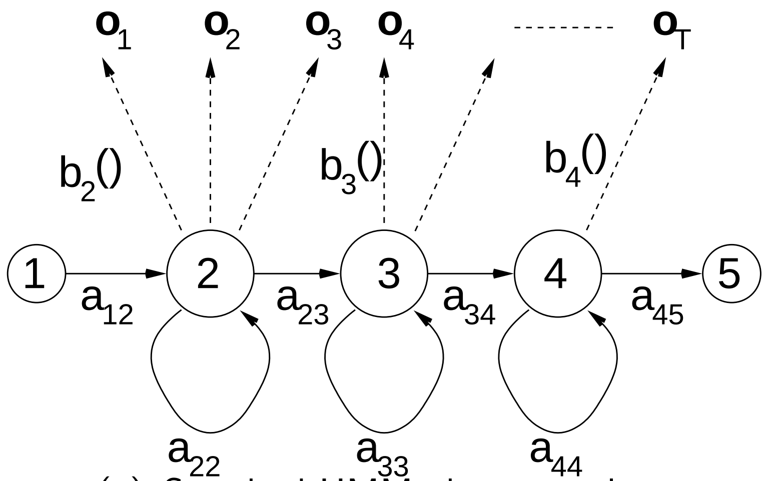
Dependency Modelling for Speech Recognition

- Sequence kernels for text-independent speaker verification used GMMs
  - for ASR interested modelling inter-frame dependencies
- Dependency modelling essential part of modelling sequence data:

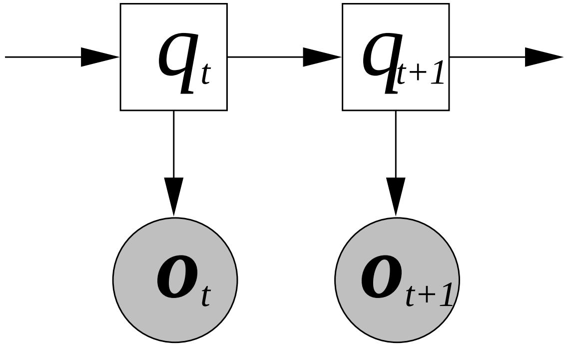
$$p(\boldsymbol{o}_1, \ldots, \mathbf{o}_T; \boldsymbol{\lambda}) = p(\boldsymbol{o}_1; \boldsymbol{\lambda}) p(\boldsymbol{o}_2 | \boldsymbol{o}_1; \boldsymbol{\lambda}) \ldots p(\boldsymbol{o}_T | \boldsymbol{o}_1, \ldots, \boldsymbol{o}_{T-1}; \boldsymbol{\lambda})$$

  - impractical to directly model in this form
- Two possible forms of conditional independence used:
  - observed variables
  - latent (unobserved) variables
- Even given dependencies (form of Bayesian Network):
  - need to determine how dependencies interact

Hidden Markov Model - A Dynamic Bayesian Network



(a) Standard HMM phone topology



(b) HMM Dynamic Bayesian Network

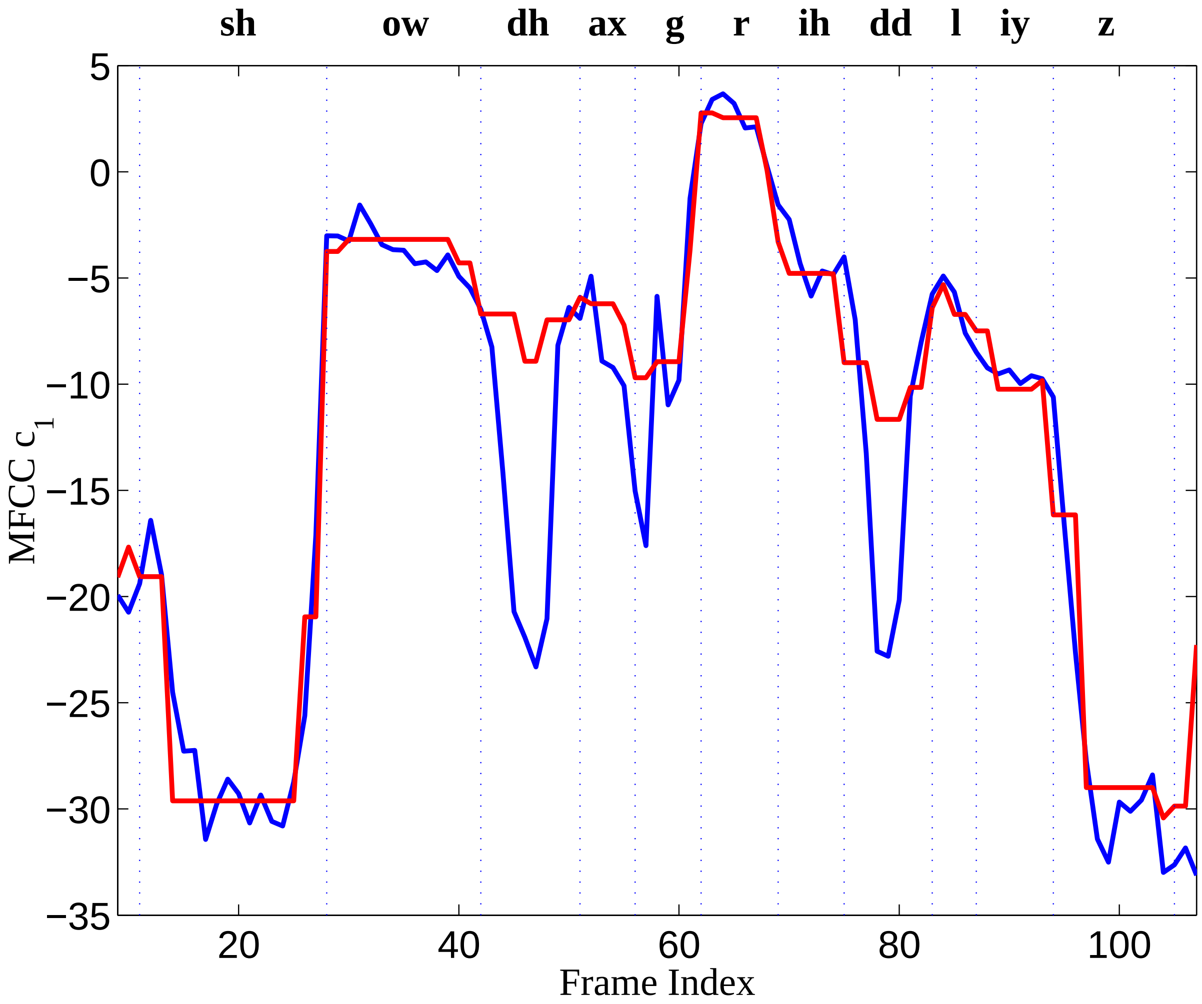
- Notation for DBNs [1]:

  circles - continuous variables   shaded - observed variables
  squares - discrete variables   non-shaded - unobserved variables
- Observations conditionally independent of other observations given state.
- States conditionally independent of other states given previous states.

$$p(\mathbf{O};\boldsymbol{\lambda}) = \sum_{\mathbf{q}} \prod_{t=1}^{T} P(q_t|q_{t-1})p(\boldsymbol{o}_t|q_t;\boldsymbol{\lambda})$$

HMM Trajectory Modelling

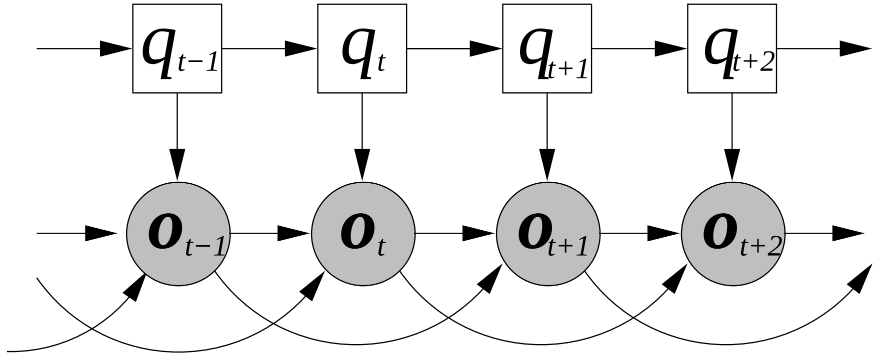
Frames from phrase:
`SHOW THE GRIDLEY'S ...`



Legend

- True
- HMM

Dependency Modelling using Observed Variables



- Commonly use member (or mixture) of the exponential family

$$p(\mathbf{O};\boldsymbol{\alpha}) = \prod_{t=1}^{T} \frac{1}{Z_t} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{o}_{t-n},\ldots,\boldsymbol{o}_t,q_t)\right)$$

  - $\boldsymbol{\phi}(\boldsymbol{o}_{t-n},\ldots,\boldsymbol{o}_t)$ are the sufficient statistic from window of n frames
  - $\boldsymbol{\alpha}$ are the natural parameters, Z_t the (local) normalisation term

$$Z_t = \int \exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{o}_{t-n},\ldots,\boldsymbol{o}_t)\right) d\boldsymbol{o}_{t-n},\ldots,d\boldsymbol{o}_t$$

- What is the appropriate form of statistics ($\boldsymbol{\phi}(\mathbf{O})$) - needs DBN to be known

Discriminative Models

- Classification requires class posteriors $P(\mathbf{w}|\mathbf{O})$
  - Generative model - e.g. HMM previously discussed

$$P(\mathbf{w}|\mathbf{O};\boldsymbol{\lambda}) = \frac{p(\mathbf{O}|\mathbf{w};\boldsymbol{\lambda})P(\mathbf{w})}{\sum_{\tilde{\mathbf{w}}} p(\mathbf{O}|\tilde{\mathbf{w}};\boldsymbol{\lambda})P(\tilde{\mathbf{w}})}$$

  - Discriminative model - directly model posterior
- Log-Linear Model discriminative form of interest here

$$P(\mathbf{w}|\mathbf{O};\boldsymbol{\alpha}) = \frac{1}{Z}\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w})\right)$$

  - normalisation term Z (simpler to compute than generative model)

$$Z = \sum_{\tilde{\mathbf{w}}}\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\tilde{\mathbf{w}})\right)$$

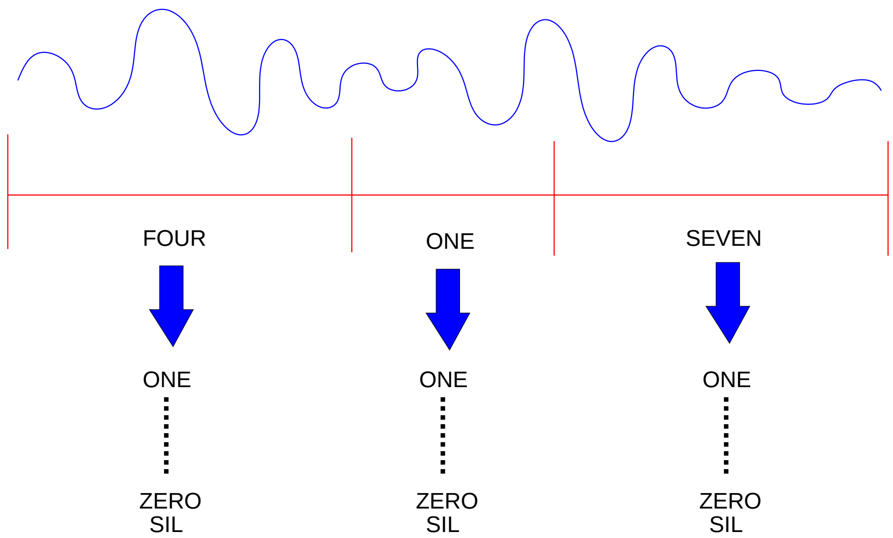
- BUT still need to decide form of features $\boldsymbol{\phi}(\mathbf{O},\mathbf{w})$

Sequence Discriminative Models

- Applying discriminative models to speech data is non-trivial:
  1. Number of possible classes is vast
     – motivates the use of structured discriminative models
  2. Length of observation $\mathbf{O}$ varies from utterance to utterance
     – motivates the use of sequence kernels to obtain features
  3. Number of labels (words) and observations (frames) differ
     – addressed by combining solutions to (1) and (2)
- To handle these a segmentation $\boldsymbol{a}$ is often required
- A range of features are then possible based on:
  – word sequences $\boldsymbol{\phi}(\mathbf{w})$ - "language-model"-like
  – segmentation-word sequences $\boldsymbol{\phi}(\boldsymbol{a}, \mathbf{w})$ - "pronunciation-model"-like
  – segmentation-observation sequences $\boldsymbol{\phi}(\mathbf{O}_{\{a_i\}}, a_i^{\mathtt{i}})$ - "acoustic-model"-like

Code-Breaking Style

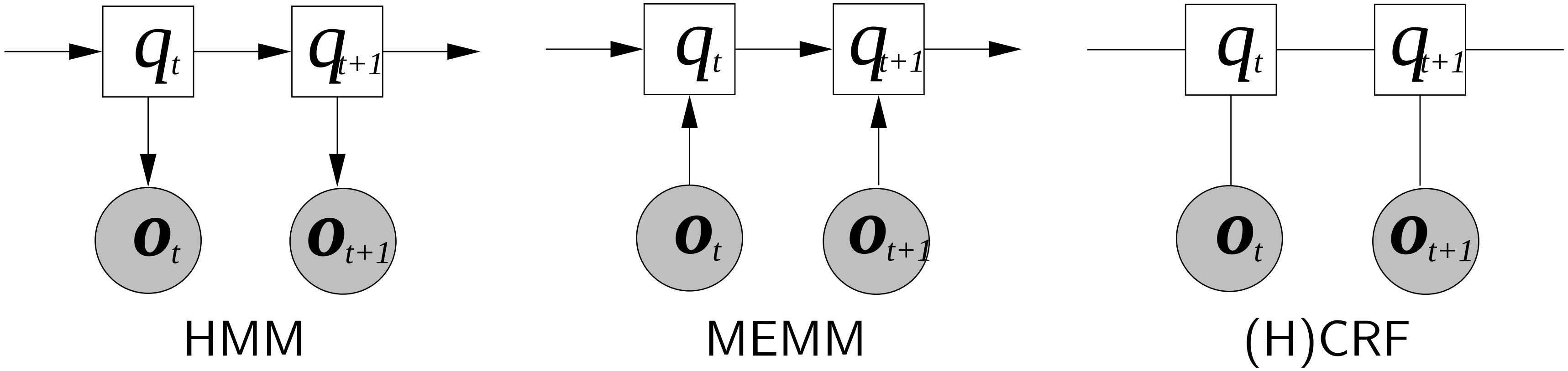
- Rather than handle complete sequence - split into segments
  - perform simpler classification for each segment
  - complexity determined by segment (simplest word)



1. Using HMM-based hypothesis
   - word start/end
2. Foreach segment of $\boldsymbol{a}$:
   - binary SVMs voting
   - $\displaystyle \arg\max_{\omega\in\{\texttt{ONE},\ldots,\texttt{SIL}\}} \boldsymbol{\alpha}^{(\omega)^{\mathsf{T}}}\boldsymbol{\phi}(\mathbf{O}_{\{a_i\}},\omega)$

- Limitations of code-breaking approach [2]
  - each segment is treated independently
  - restrict to one segmentation, generated by HMMs

Example Standard Sequence Models



- The segmentation, $\boldsymbol{a}$, determines the state-sequence $\mathbf{q}$
  - maximum entropy Markov model [3]

$$P(\mathbf{q}|\mathbf{O}) = \prod_{t=1}^{T} \frac{1}{Z_t} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(q_t, q_{t-1}, \boldsymbol{o}_t)\right)$$

  - hidden conditional random field (simplified linear form only) [4]

$$P(\mathbf{q}|\mathbf{O}) = \frac{1}{Z}\prod_{t=1}^{T} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(q_t, q_{t-1}, \boldsymbol{o}_t)\right)$$

Features

- Discriminative sequence models have simple sufficient statistics
  - simple models - second-order statistics (almost) a discriminative HMM
  - simplest approach extend frame features (for each state $\mathbf{s}_i$)

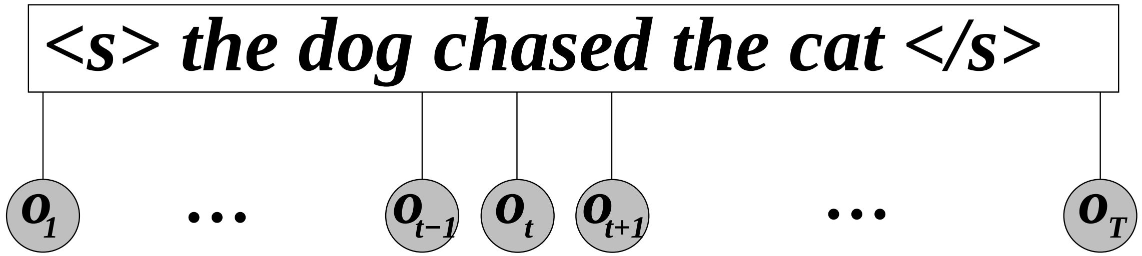
$$\boldsymbol{\phi}(q_t, q_{t-1}, \boldsymbol{o}_t) = \begin{bmatrix} \delta(q_t, \mathbf{s}_i) \\ \delta(q_t, \mathbf{s}_i)\delta(q_{t-1}, \mathbf{s}_j) \\ \delta(q_t, \mathbf{s}_i)\boldsymbol{o}_t \\ \delta(q_t, \mathbf{s}_i)\boldsymbol{o}_t \otimes \boldsymbol{o}_t \\ \delta(q_t, \mathbf{s}_i)\boldsymbol{o}_t \otimes \boldsymbol{o}_t \otimes \boldsymbol{o}_t \end{bmatrix}$$

  - still same conditional independence assumption as HMM

How to extend range of features?

- Consider features a particular segment of speech
  - size of each segment may vary from segment to segment
  - need to map to a fixed dimensionality independent of number of frames

Flat Direct Models

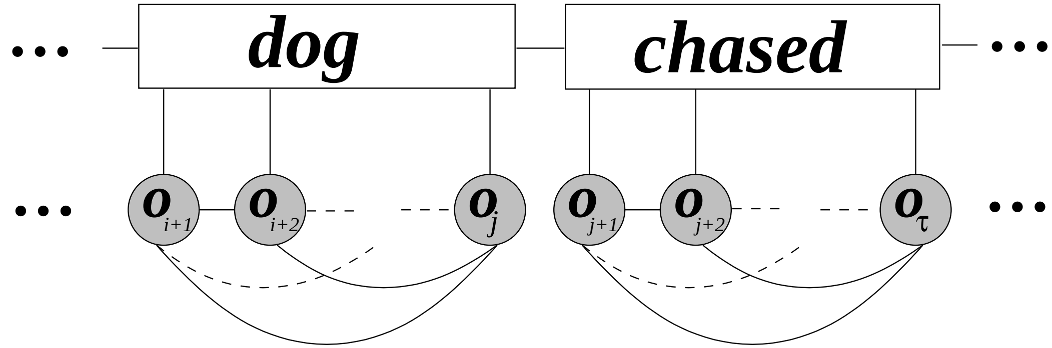


- Remove conditional independence assumptions

$$P(\mathbf{w}|\mathbf{O}) = \frac{1}{Z}\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w})\right)$$

- Simple model, but lack of structure causes problems
  - extracted feature-space becomes vast (number of possible sentences)
  - associated parameter vector is vast
  - large number of unseen examples

Structured Discriminative Models



- Introduce structure into observation sequence - segmentation $\boldsymbol{a}$
  - comprises: segmentation identity $a^{\mathtt{i}}$, set of observations $\mathbf{O}_{\{a\}}$

$$P(\mathbf{w}|\mathbf{O}) = \frac{1}{Z}\sum_{\boldsymbol{a}} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\left[\sum_{\tau=1}^{|\boldsymbol{a}|}\boldsymbol{\phi}(\mathbf{O}_{\{a_\tau\}}, a^{\mathtt{i}}_{\tau})\right]\right)$$

  - segmentation may be at word, (context-dependent) phone, etc etc
- What form should $\boldsymbol{\phi}(\mathbf{O}_{\{a_\tau\}}, a^{\mathtt{i}}_{\tau})$ have?
  - must be able to handle variable length $\mathbf{O}_{\{a_\tau\}}$

Sequence Kernels

Sequence Kernel

- Sequence kernels are a class of kernel that handles sequence data
  - also applied in a range of biological applications, text processing, speech
  - in this talk these kernels will be partitioned into three classes
- Discrete-observation kernels
  - appropriate for text data
  - string kernels simplest form
- Distributional kernels
  - distances between distributions trained on sequences
- Generative kernels:
  - parametric form: use the parameters of the generative model
  - derivative form: use the derivatives with respect to the model parameters

String Kernel

- For speech and text processing input space has variable dimension:
  - use a kernel to map from variable to a fixed length;
  - string kernels are an example for text [5].
- Consider the words cat, cart, bar and a character string kernel

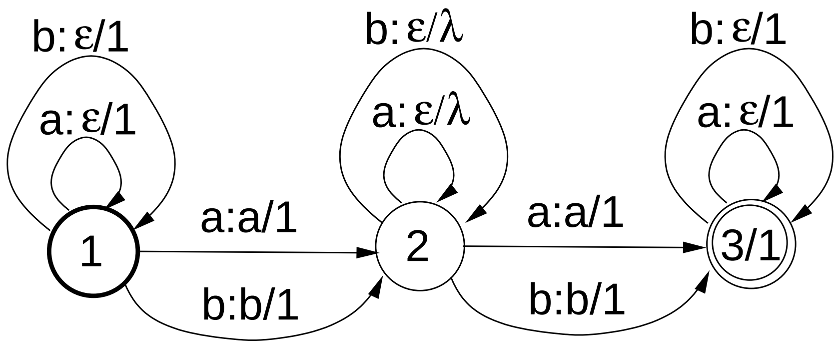
	c-a	c-t	c-r	a-r	r-t	b-a	b-r
$\phi(\texttt{cat})$	1	λ	0	0	0	0	0
$\phi(\texttt{cart})$	1	λ^2	λ	1	1	0	0
$\phi(\texttt{bar})$	0	0	0	1	0	1	λ

$$K(\texttt{cat},\texttt{cart}) = 1+\lambda^3, \quad K(\texttt{cat},\texttt{bar}) = 0, \quad K(\texttt{cart},\texttt{bar}) = 1$$

- Successfully applied to various text classification tasks:
  - how to make process efficient (and more general)?

Rational Kernels

- Rational kernels [6] encompass various standard feature-spaces and kernels:
  - bag-of-words and N-gram counts, gappy N-grams (string Kernel),
- A transducer, T, for the string kernel (gappy bigram) (vocab $\{\mathtt{a}, \mathtt{b}\}$)



The kernel is: $K(\boldsymbol{O}_i, \boldsymbol{O}_j) = w\left[\boldsymbol{O}_i \circ (T \circ T^{-1}) \circ \boldsymbol{O}_j\right]$

- This form can also handle uncertainty in decoding:
  - lattices can be used rather than the 1-best output ($\boldsymbol{O}_i$).
- Can also be applied for continuous data kernels [7].

Generative Score-Spaces

- Generative kernels use scores of the following form [8]

$$\phi(\mathbf{O};\boldsymbol{\lambda}) = [\log(p(\mathbf{O};\boldsymbol{\lambda}))]$$

  - simplest form maps sequence to 1-dimensional score-space
- Parametric score-space increase the score-space size

$$\phi(\mathbf{O};\boldsymbol{\lambda}) = \left[\begin{array}{c} \hat{\boldsymbol{\lambda}}^{(1)} \\ \vdots \\ \hat{\boldsymbol{\lambda}}^{(K)} \end{array}\right]$$

  - parameters estimated on $\mathbf{O}$: related to the mean-supervector kernel
- Derivative score-space take the following form

$$\phi\left(\mathbf{O};\boldsymbol{\lambda}\right) = \left[\boldsymbol{\nabla}_{\boldsymbol{\lambda}} \log\left(p(\mathbf{O};\boldsymbol{\lambda})\right)\right]$$

  - using the appropriate metric this is the Fisher kernel [9]

Generative Kernels

- Associated kernel for generative score-spaces is:

$$K(\mathbf{O}_i, \mathbf{O}_j; \boldsymbol{\lambda}) = \boldsymbol{\phi}(\mathbf{O}_i; \boldsymbol{\lambda})^{\mathsf{T}} \mathbf{G}^{-1} \boldsymbol{\phi}(\mathbf{O}_j; \boldsymbol{\lambda})$$

  - $\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda})$ is the score-space for $\mathbf{O}$ using parameters $\boldsymbol{\lambda}$
  - $\mathbf{G}$ is the appropriate metric for the score-space

- The exact form of the metric is important

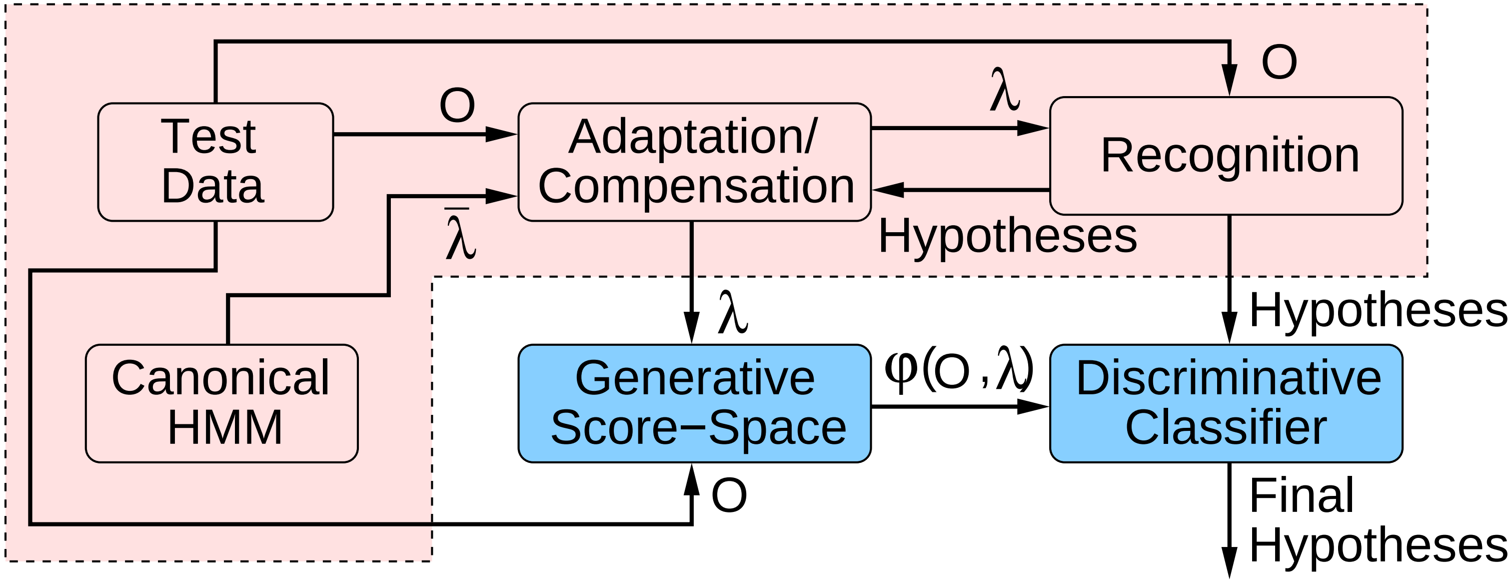
  - standard form is a maximally non-committal metric

$$\boldsymbol{\mu}_{\mathrm{g}} = \mathcal{E}\left\{\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda})\right\}; \quad \mathbf{G} = \boldsymbol{\Sigma}_{\mathrm{g}} = \mathcal{E}\left\{(\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda}) - \boldsymbol{\mu}_{\mathrm{g}})(\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda}) - \boldsymbol{\mu}_{\mathrm{g}})^{\mathsf{T}}\right\}$$

  - empirical approximation based on training data is often used
  - equal "weight" given to all dimensions
  - Fisher kernel with ML-trained models $\mathbf{G}$ Fisher Information Matrix

Combining Generative & Discriminative Models

Combining Discriminative and Generative Models



- Use generative model to extract features [9, 8] (we do like HMMs!)
  - adapt generative model - speaker/noise independent discriminative model
- Use favourite form of discriminative classifier for example
  - log-linear model/logistic regression
  - binary/multi-class support vector machines

Score-Space Sufficient Statistics

- Need a systematic approach to extracting sufficient statistics
  - what about using the sequence-kernel score-spaces?

$$\boldsymbol{\phi}(\mathbf{O}) = \boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda})$$

  - does this help with the dependencies?

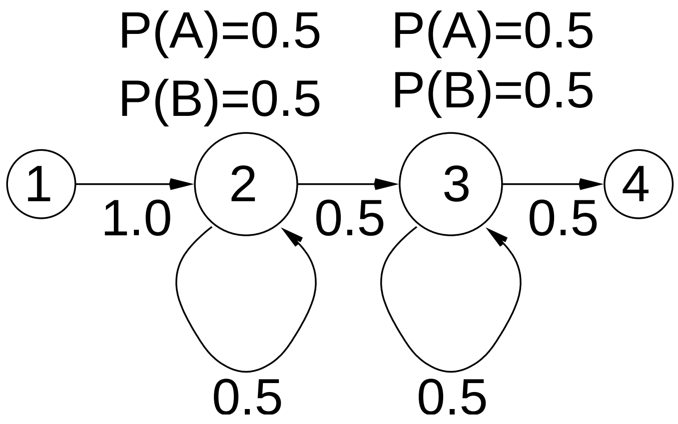
- For an HMM the mean derivative elements become

$$\boldsymbol{\nabla}_{\boldsymbol{\mu}^{(jm)}} \log(p(\mathbf{O}; \boldsymbol{\lambda})) = \sum_{t=1}^{T} P(\mathbf{q}_t = \{\theta_j, m\} | \mathbf{O}; \boldsymbol{\lambda}) \boldsymbol{\Sigma}^{(jm)\text{-}1}(\boldsymbol{o}_t - \boldsymbol{\mu}^{(jm)})$$

  - state/component posterior a function of complete sequence $\mathbf{O}$
  - introduces longer term dependencies
  - different conditional-independence assumptions than generative model

Score-Space Dependencies

- Consider a simple 2-class, 2-symbol $\{\texttt{A}, \texttt{B}\}$ problem:
  - Class ω_1: AAAA, BBBB
  - Class ω_2: AABB, BBAA



Feature	Class ω_1		Class ω_2	
	AAAA	BBBB	AABB	BBAA
Log-Lik	-1.11	-1.11	-1.11	-1.11
∇_{2A}	0.50	-0.50	0.33	-0.33
$\nabla_{2A}\nabla_{2A}^{\mathsf{T}}$	-3.83	0.17	-3.28	-0.61
$\nabla_{2A}\nabla_{3A}^{\mathsf{T}}$	-0.17	-0.17	-0.06	-0.06

- ML-trained HMMs are the same for both classes
- First derivative classes separable, but not linearly separable
  - also true of second derivative within a state
- Second derivative across state linearly separable

Score-Spaces for ASR

- Forms of score-space used in the experiments:

$$\boldsymbol{\phi}^{\mathsf{a}}_{0}(\mathbf{O};\boldsymbol{\lambda}) = \begin{bmatrix} \log\left(p(\mathbf{O};\boldsymbol{\lambda}^{(1)})\right) \\ \vdots \\ \log\left(p(\mathbf{O};\boldsymbol{\lambda}^{(K)})\right) \end{bmatrix}; \quad \boldsymbol{\phi}^{\mathsf{b}}_{1\mu}(\mathbf{O};\boldsymbol{\lambda}) = \begin{bmatrix} \log\left(p(\mathbf{O};\boldsymbol{\lambda}^{(i)})\right) \\ \boldsymbol{\nabla}_{\boldsymbol{\mu}^{(i)}} \log\left(p(\mathbf{O};\boldsymbol{\lambda}^{(i)})\right) \end{bmatrix}$$

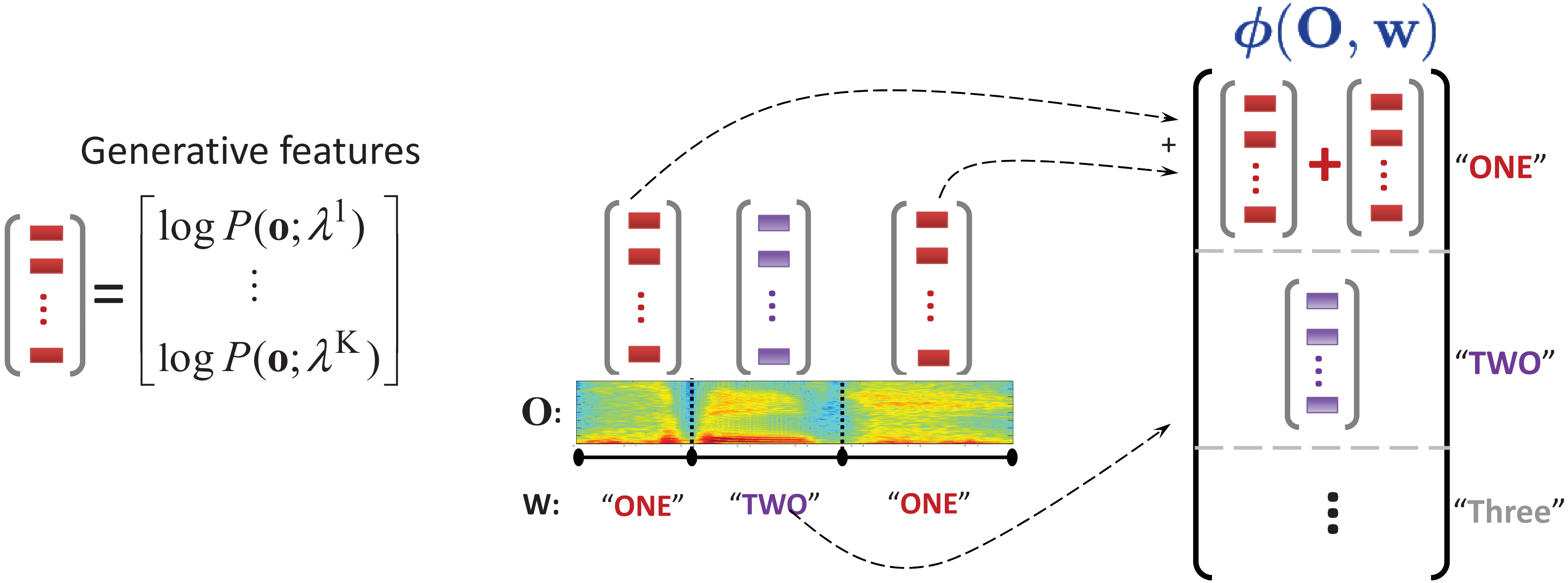
  - appended log-likelihood: $\boldsymbol{\phi}^{\mathsf{a}}_{0}(\mathbf{O};\boldsymbol{\lambda})$
  - derivative (means only for class ω_i): $\boldsymbol{\phi}^{\mathsf{b}}_{1\mu}(\mathbf{O};\boldsymbol{\lambda})$
  - log-likelihood (for class ω_i): $\boldsymbol{\phi}^{\mathsf{b}}_{0}(\mathbf{O};\boldsymbol{\lambda}) = \left[\log\left(p(\mathbf{O};\boldsymbol{\lambda}^{(i)})\right)\right]$

- In common with most discriminative models Joint Feature Spaces,

$$\boldsymbol{\phi}(\mathbf{O},\boldsymbol{a};\boldsymbol{\lambda}) = \begin{bmatrix} \sum_{\tau=1}^{|\boldsymbol{a}|} \delta(a^{\mathtt{i}}_{\tau}, w^{(1)})\boldsymbol{\phi}(\mathbf{O}_{\{a_\tau\}};\boldsymbol{\lambda}) \\ \vdots \\ \sum_{\tau=1}^{|\boldsymbol{a}|} \delta(a^{\mathtt{i}}_{\tau}, w^{(P)})\boldsymbol{\phi}(\mathbf{O}_{\{a_\tau\}};\boldsymbol{\lambda}) \end{bmatrix}$$

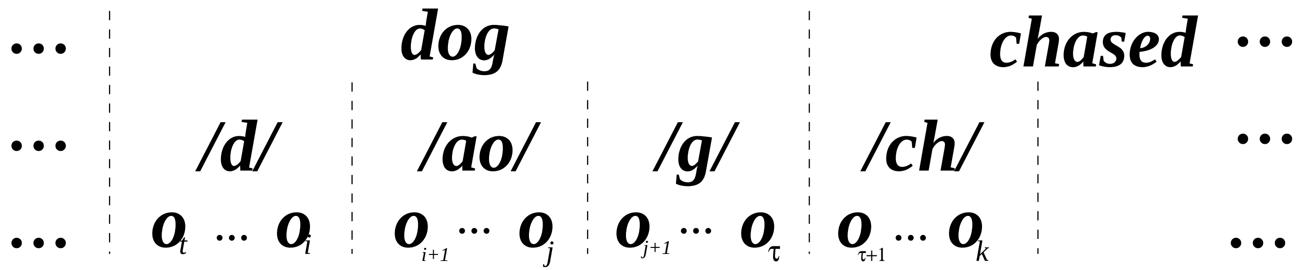
for $\boldsymbol{\alpha}$-tied yielding "units" $\{w^{(1)},\ldots,w^{(P)}\}$, underlying score-space $\boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda})$.

Joint Feature-Space Example



- Size of joint feature-space is the product of
  1. feature-space size (K)- determined by generative model
  2. number of α classes (P) - determined by discriminative model
- Segmentation of the sentence will alter scores

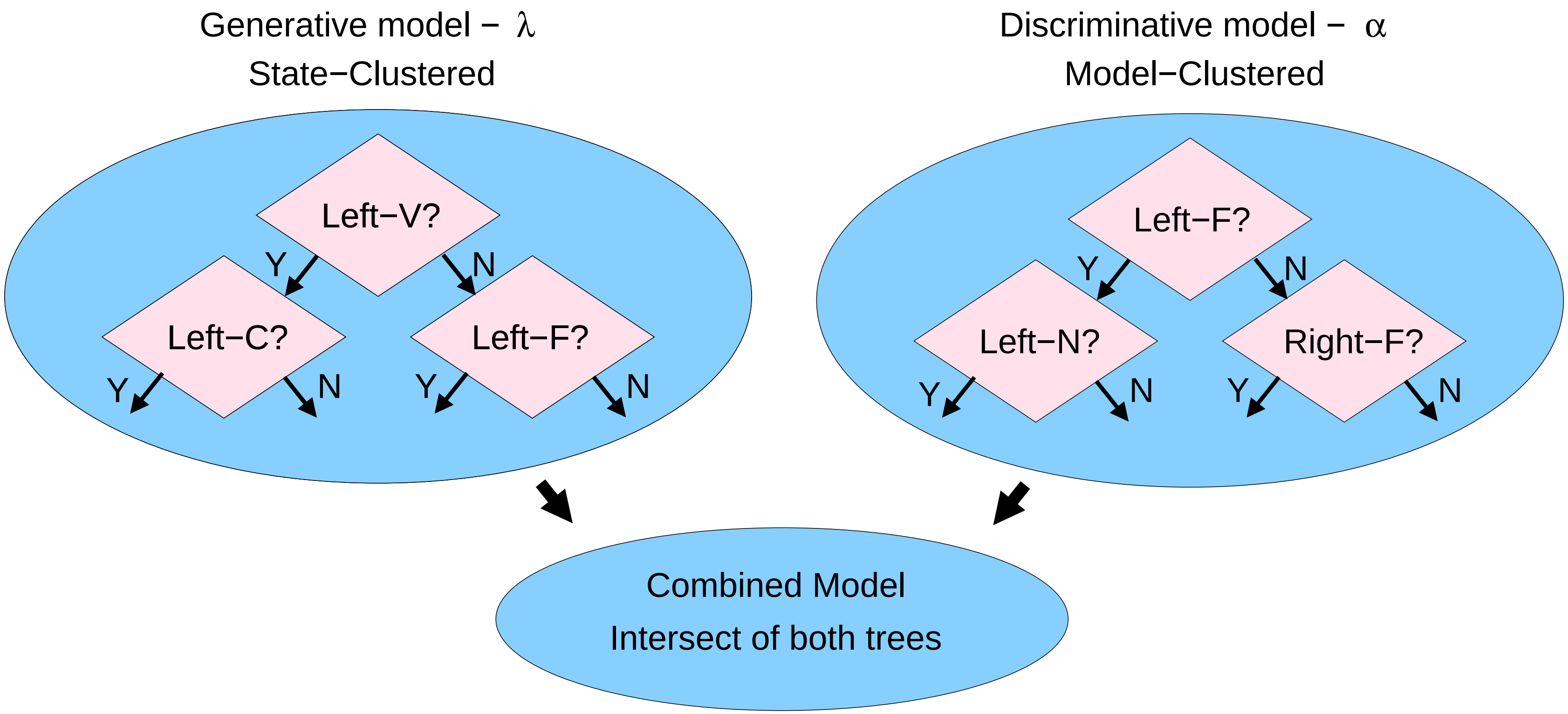
Segmentation



- Segmentation can be viewed at multiple levels
  - sentence: yields flat direct model - standard problems
  - word: easy implementation for small vocab, sparsity issues
  - phone: may be context-dependent
  - state: very flexible, but large number of segments
- Multiple levels of segmentation can be used/combined
  - multiple segmentations can be used to derive features
  - can use different segmentations for generative/discriminative models

Parameter Tying

- Parameter tying in combined classifier [10]
  - two sets of parameters discriminative $\boldsymbol{\alpha}$, generative $\boldsymbol{\lambda}$



  - tree-intersect can cause generalisation problems

Handling Latent Variables

- Two forms of model can be used:

1. marginalise over all possible segmentations

$$P(\mathbf{w}|\mathbf{O}) = \frac{1}{Z}\sum_{\boldsymbol{a}} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\left[\sum_{\tau=1}^{|\boldsymbol{a}|}\boldsymbol{\phi}(\mathbf{O}_{\{a_\tau\}}, a_\tau^{\mathtt{i}})\right]\right)$$

2. use "best" segmentation

$$P(\mathbf{w}|\mathbf{O}, \hat{\boldsymbol{a}}) = \frac{1}{Z}\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\left[\sum_{\tau=1}^{|\hat{\boldsymbol{a}}|}\boldsymbol{\phi}(\mathbf{O}_{\{\hat{a}_i\}}, \hat{a}_\tau^{\mathtt{i}})\right]\right)$$

$$\hat{\boldsymbol{a}} = \operatorname*{argmax}_{\boldsymbol{a}}\left\{\exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\left[\sum_{\tau=1}^{|\boldsymbol{a}|}\boldsymbol{\phi}(\mathbf{O}_{\{a_\tau\}}, a_\tau^{\mathtt{i}})\right]\right)\right\}$$

Approximate Training/Inference Schemes

- If HMMs are being used anyway - use for segmentation $\mathcal{O}(T)$
  - simplest approach use Viterbi (1-best) segmentation from HMM, $\hat{\boldsymbol{a}}_{\text{hmm}}$
  - use fixed segmentation in training and test - highly efficient

$$P(\mathbf{w}|\mathbf{O}) \approx \frac{1}{Z}\prod_{\tau=1}^{|\hat{\boldsymbol{a}}_{\text{hmm}}|} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O}_{\{\hat{a}_{\text{hmm}\tau}\}}, \hat{a}^{\mathtt{i}}_{\text{hmm}\tau})\right)$$

$$\hat{\boldsymbol{a}}_{\text{hmm}} = \underset{\boldsymbol{a}}{\operatorname{argmax}}\left\{p(\mathbf{O}|\boldsymbol{a},\boldsymbol{\lambda})P(\boldsymbol{a})\right\}$$

- Assumption: segmentation not dependent on discriminative model parameters
  - unclear how accurate appropriate this is!
- Schemes for efficient inference feature extraction possible [11]

Handling Speaker/Noise Differences

- A standard problem with kernel-based approaches is adaptation/robustness
  - not a problem with generative kernels
  - adapt generative models using model-based adaptation
- Standard approaches for speaker/environment adaptation
  - (Constrained) Maximum Likelihood Linear Regression [12]

$$\boldsymbol{x}_t = \mathbf{A}\boldsymbol{o}_t + \mathbf{b}; \quad \boldsymbol{\mu}^{(m)} = \mathbf{A}\boldsymbol{\mu}_{\mathrm{x}}^{(m)} + \mathbf{b}$$

  - Vector Taylor Series Compensation [13] (used in this work)

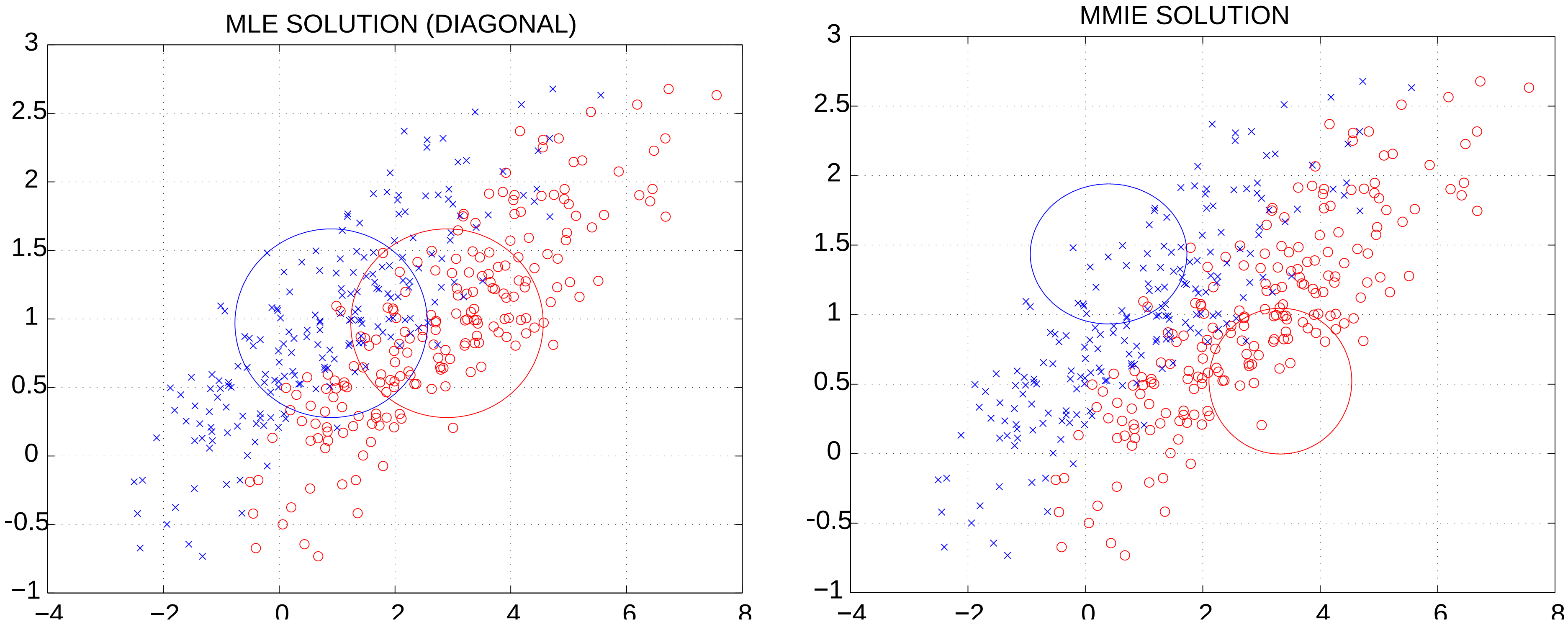
$$\boldsymbol{\mu}^{(m)} = \mathbf{C}\log\left(\exp(\mathbf{C}^{-1}(\boldsymbol{\mu}_{\mathrm{x}}^{(m)} + \boldsymbol{\mu}_{\mathrm{h}}^{(m)})) + \exp(\mathbf{C}^{-1}\boldsymbol{\mu}_{\mathrm{n}}^{(m)})\right)$$

- Adapting the generative model will alter score-space

Training Criteria

Simple MMIE Example

- HMMs are not the correct model - discriminative criteria a possibility



- Discriminative criteria a function of posteriors $P(\mathbf{w}|\mathbf{O};\boldsymbol{\lambda})$
  - use to train the discriminative model parameters $\boldsymbol{\alpha}$

Discriminative Training Criteria

- Apply discriminative criteria to train discriminative model parameters$\boldsymbol{\alpha}$
  - Conditional Maximum Likelihood (CML) [14, 15]: maximise

$$\mathcal{F}_{\tt cml}(\boldsymbol{\alpha}) = \frac{1}{R}\sum_{r=1}^{R}\log(P(\mathbf{w}_{\tt ref}^{(r)}|\mathbf{O}^{(r)};\boldsymbol{\alpha}))$$

  - Minimum Classification Error (MCE) [16]: minimise

$$\mathcal{F}_{\tt mce}(\boldsymbol{\alpha}) = \frac{1}{R}\sum_{r=1}^{R}\left(1+\left[\frac{P(\mathbf{w}_{\tt ref}^{(r)}|\mathbf{O}^{(r)};\boldsymbol{\alpha})}{\sum_{\mathbf{w}\neq\mathbf{w}_{\tt ref}^{(r)}}P(\mathbf{w}|\mathbf{O}^{(r)};\boldsymbol{\alpha})}\right]^{\varrho}\right)^{-1}$$

  - Minimum Bayes' Risk (MBR) [17, 18]: minimise

$$\mathcal{F}_{\tt mbr}(\boldsymbol{\alpha}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{\mathbf{w}}P(\mathbf{w}|\mathbf{O}^{(r)};\boldsymbol{\alpha})\mathcal{L}(\mathbf{w},\mathbf{w}_{\tt ref}^{(r)})$$

MBR Loss Functions for ASR

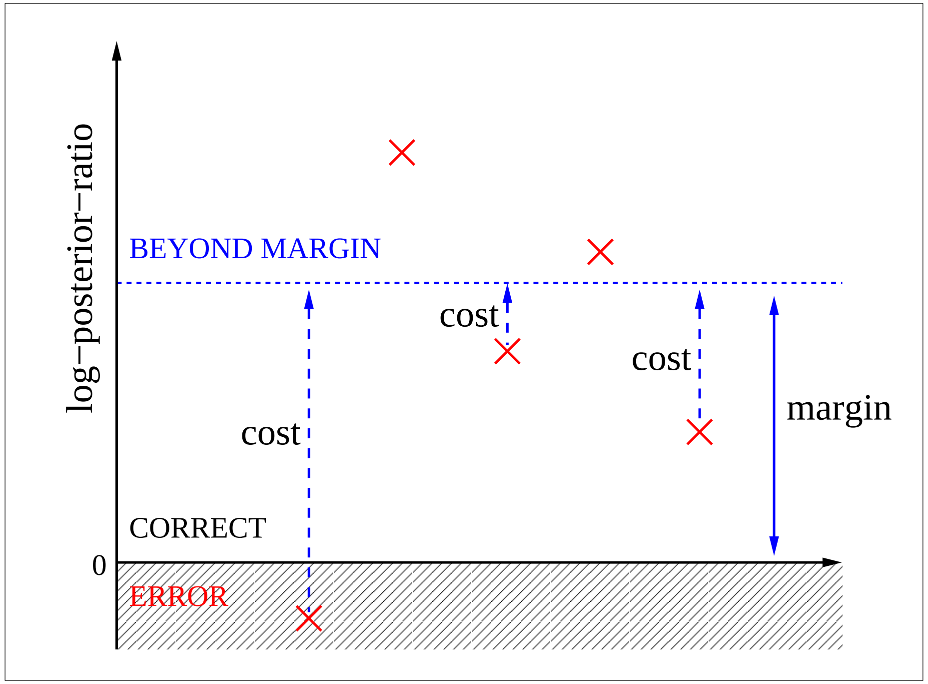
- Sentence (1/0 loss):

$$\mathcal{L}(\mathbf{w}, \mathbf{w}_{\texttt{ref}}^{(r)}) = \begin{cases} 1; & \mathbf{w} \neq \mathbf{w}_{\texttt{ref}}^{(r)} \\ 0; & \mathbf{w} = \mathbf{w}_{\texttt{ref}}^{(r)} \end{cases}$$

When $\varrho = 1$, $\mathcal{F}_{\texttt{mce}}(\boldsymbol{\alpha}) = \mathcal{F}_{\texttt{mbr}}(\boldsymbol{\alpha})$

- Word: directly related to minimising the expected Word Error Rate (WER)
  - normally computed by minimising the Levenshtein edit distance.
- Phone: consider phone rather word loss
  - improved generalisation as more "error's" observed
  - this is known as Minimum Phone Error (MPE) training [19, 20].
- Hamming (MPFE): number of erroneous frames measured at the phone level

Large Margin Based Criteria



- Standard criterion for SVMs
  - improves generalisation
- Require log-posterior-ratio

$$\min_{\mathbf{w}\neq\mathbf{w}_{\tt ref}} \left\{ \log\left(\frac{P(\mathbf{w}_{\tt ref}|\mathbf{O};\boldsymbol{\alpha})}{P(\mathbf{w}|\mathbf{O};\boldsymbol{\alpha})}\right)\right\}$$

  to be beyond margin

- As sequences being used can make margin function of the “loss” - minimise

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

  use hinge-loss $[f(x)]_+$. Many variants possible [21, 22, 23, 24]

Relationship to (Structured) SVM

- Commonly add a Gaussian prior for regularisation

$$\mathcal{F}(\boldsymbol{\alpha}) = \log\left(\mathcal{N}(\boldsymbol{\alpha};\boldsymbol{\mu}_\alpha;\boldsymbol{\Sigma}_\alpha)\right) + \mathcal{F}_{\tt lm}(\boldsymbol{\alpha})$$

- Make the posteriors a log-linear model ($\boldsymbol{\alpha}$) with generative score-space ($\boldsymbol{\lambda}$) [25]
  - restrict parameters of the prior: $\mathcal{N}(\boldsymbol{\alpha};\boldsymbol{\mu}_\alpha;\boldsymbol{\Sigma}_\alpha) = \mathcal{N}(\boldsymbol{\alpha};\mathbf{0},C\mathbf{I})$

$$\mathcal{F}(\boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{\alpha}||^2 + \frac{C}{R}\sum_{r=1}^{R}\left[\max_{\mathbf{w}\neq\mathbf{w}^{(r)}_{\tt ref}}\left\{\mathcal{L}(\mathbf{w},\mathbf{w}^{(r)}_{\tt ref}) - \log\left(\frac{\boldsymbol{\alpha}^{\sf T}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w}^{(r)}_{\tt ref};\boldsymbol{\lambda})}{\boldsymbol{\alpha}^{\sf T}\boldsymbol{\phi}(\mathbf{O}^{(r)},\mathbf{w};\boldsymbol{\lambda})}\right)\right\}\right]_{+}$$

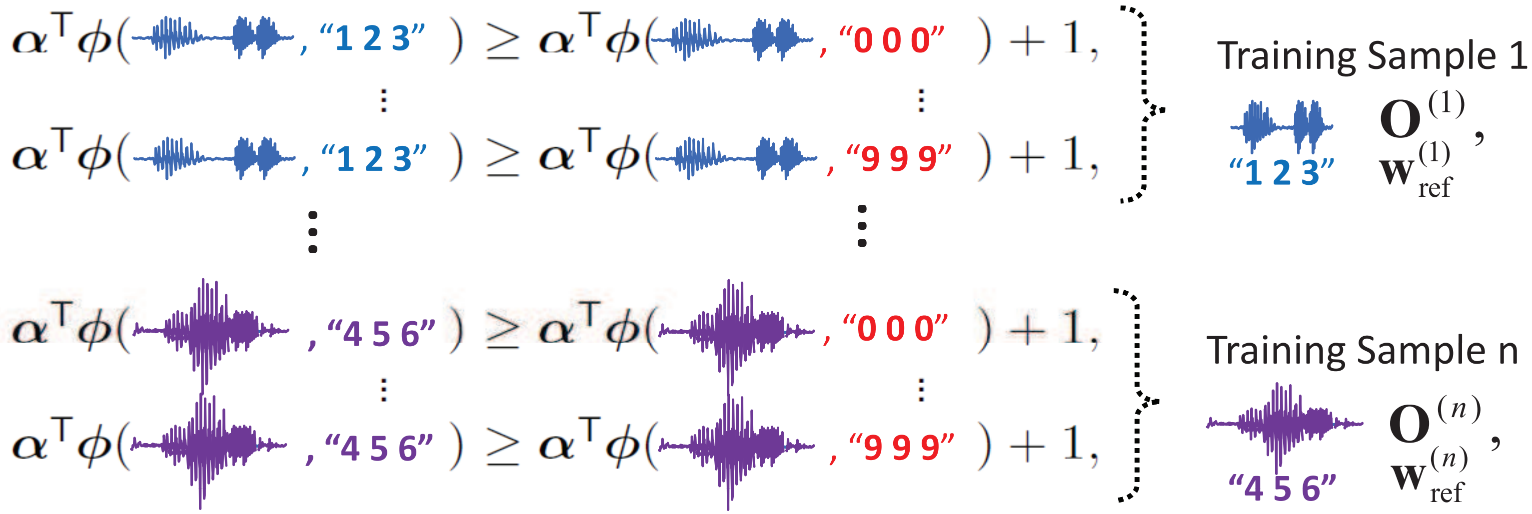
- Standard result - it's a structured SVM [26, 25]

Structured SVM Training

- Training $\boldsymbol{\alpha}$, so that $\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{O},\mathbf{w})$ is max for correct reference $\mathbf{w}_{\texttt{ref}}$:

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}||\boldsymbol{\alpha}||^2$$

s.t.



- General unconstrained form: use cutting plane algorithm to solve [27, 28]

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

Handling Latent Variables

- Ignored the issue of alignment so far
  - for SSVM necessary to use the "best" segmentation
- Simplest solution is to use the single segmentation from the original HMM

$$\hat{\boldsymbol{a}}_{\tt hmm} = \underset{\boldsymbol{a}}{\operatorname{argmax}} \left\{ \log\left(P(\boldsymbol{a}|\mathbf{O}, \mathbf{w}; \boldsymbol{\lambda}) \right) \right\} = \underset{\boldsymbol{a}}{\operatorname{argmax}} \left\{ \log\left(P(\mathbf{O}|\boldsymbol{a}, \mathbf{w}; \boldsymbol{\lambda}) P(\boldsymbol{a}|\mathbf{w}; \boldsymbol{\lambda}) \right) \right\}$$

  - equivalent of phone/word-marking lattices
  - BUT underlying model changes: would like

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a}}{\operatorname{argmax}} \left\{ \log\left(P(\mathbf{O}|\boldsymbol{a}, \mathbf{w}; \boldsymbol{\lambda}, \boldsymbol{\alpha}) \right) + \log\left(P(\boldsymbol{a}|\mathbf{w}; \boldsymbol{\lambda}, \boldsymbol{\alpha}) \right) \right\}$$

Maps into a Concave-Convex Procedure (CCCP) [29]

$$\Bigg[\overbrace{-\max_{\boldsymbol{a}} \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(i)}, \mathbf{w}_{\tt ref}^{(i)}, \boldsymbol{a})}^{\text{concave}} + \overbrace{\max_{\mathbf{w} \neq \mathbf{w}_{\tt ref}, \boldsymbol{a}} \left\{ \mathcal{L}(\mathbf{w}, \mathbf{w}_{\tt ref}^{(i)}) + \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(i)}, \mathbf{w}, \boldsymbol{a}) \right\}}^{\text{convex}} \Bigg]_{+}$$

Joint Training of Parameters and Segmentation

Input: $\{\mathbf{O}^{(1)}, \mathbf{w}_{\texttt{ref}}^{(1)}\}, \ldots, \{\mathbf{O}^{(R)}, \mathbf{w}_{\texttt{ref}}^{(R)}\}, C, \epsilon$
Output: $\{\boldsymbol{\alpha}, \boldsymbol{\xi}\}$
initialise: constraints $\mathcal{W}_i \leftarrow 0$; slack variables $\boldsymbol{\xi} \leftarrow \mathbf{0}$; segmentation $\boldsymbol{a} \leftarrow \boldsymbol{a}_{\text{hmm}}$;
repeat

foreach *observation* i **do**

optimise reference segmentation given $\boldsymbol{\alpha}$:

$$\boldsymbol{a}_{\texttt{ref}}^{(i)} \leftarrow \text{argmax}_{a} \left\{ \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(i)}, \mathbf{w}_{\texttt{ref}}^{(i)}, \boldsymbol{a}; \boldsymbol{\lambda}) \right\};$$

end

optimise parameters: $\boldsymbol{\alpha} \leftarrow \text{argmin}_{\boldsymbol{\alpha}} \left\{ \mathcal{F}_{\texttt{ssvm}}(\boldsymbol{\alpha} | \boldsymbol{a}_{\texttt{ref}}^{(1)}, \ldots, \boldsymbol{a}_{\texttt{ref}}^{(R)}) \right\}$

until *all* $\boldsymbol{a}_{\texttt{ref}}^{(i)}$ *unchanged*;

$$\mathcal{F}_{\texttt{ssvm}}(\boldsymbol{\alpha} | \boldsymbol{a}_{\texttt{ref}}^{(1)}, \ldots, \boldsymbol{a}_{\texttt{ref}}^{(R)}) = ||\boldsymbol{\alpha}||^2/2 +$$

$$\frac{C}{R} \sum_{r=1}^{R} \left[\max_{\mathbf{w} \neq \mathbf{w}_{\texttt{ref}}^{(r)}} \left\{ \mathcal{L}(\mathbf{w}, \mathbf{w}_{\texttt{ref}}^{(r)}) - \min_{\boldsymbol{a}} \left\{ \log \left(\frac{\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}_{\texttt{ref}}^{(r)}, \boldsymbol{a}_{\texttt{ref}}^{(i)}; \boldsymbol{\lambda})}{\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{O}^{(r)}, \mathbf{w}, \boldsymbol{a}; \boldsymbol{\lambda})} \right) \right\} \right\} \right]_{+}$$

Evaluation Tasks

Preliminary Evaluation Tasks

- AURORA-2 small vocabulary digit string recognition task
  - whole-word models, 16 emitting-states with 3 components per state
  - clean training data for HMM training - HTK parametrisation SNR
  - Set B and Set C unseen noise conditions even for multi-style data
  - Noise estimated in a ML-fashion for each utterance
- AURORA-4 medium vocabulary speech recognition
  - training data from WSJ0 SI84 to train clean acoustic models
  - state-clustered states, cross-word triphones ($\approx$3K states $\approx$50k components)
  - 5-15dB SNR range of noises added
  - Noise estimated in a ML-fashion for each utterance
- WARNING: optimisation techniques improved over time
  - don't compare results cross-tables!

AURORA-2 - Training Criterion

Model	Criterion	Test set A	Test set B	Test set C	Avg
HMM	—	9.8	9.1	9.5	**9.5**
LLM (ϕ_0^{a})	CML	8.1	7.7	8.3	**8.1**
	MWE	7.9	7.4	8.2	**7.9**
	LM	7.8	7.3	8.0	**7.6**

- All approaches yield gains over the baseline VTS system
  - very few additional parameters added $(12 \times 12 = 144)$ for log-linear models (though these parameters are discriminatively trained
- Large-margin log-linear model will be referred to as Structured SVM

AURORA-2 - Support Vector Machines

Model	Features	Test set A	Test set B	Test set C	Avg
HMM	—	9.8	9.1	9.5	**9.5**
SVM	ϕ_0^{a}	9.1	8.7	9.2	**9.0**
MSVM		8.3	8.1	8.6	**8.3**
SSVM		7.8	7.3	8.0	**7.6**

- Possible to compare SSVM with more standard SVMs
  - segmentation for SVMs and multi-class SVMs (MSVMs) obtained from HMM
  - majority voting (HMM decision for ties on standard SVM)
- The difference between the MSVM and SSVM is the fixed HMM segmentation
  - does have an important on the performance

AURORA-2 - Optimising Segmentation

Model	Training	Segmentation $\{\texttt{trn}, \texttt{tst}\}$	Test set A	Test set B	Test set C	Avg
HMM	—	—	9.8	9.1	9.5	**9.5**
SSVM	n-slack	$\{\hat{\boldsymbol{a}}_{\rm hmm}, \hat{\boldsymbol{a}}_{\rm hmm}\}$	7.8	7.3	8.0	**7.6**
		$\{\hat{\boldsymbol{a}}_{\rm hmm}, \hat{\boldsymbol{a}}\}$	7.6	7.2	8.0	**7.5**
SSVM	n-slack batch	$\{\hat{\boldsymbol{a}}_{\rm hmm}, \hat{\boldsymbol{a}}_{\rm hmm}\}$	7.9	7.4	8.2	**7.8**
		$\{\hat{\boldsymbol{a}}_{\rm hmm}, \hat{\boldsymbol{a}}\}$	7.8	7.2	8.0	**7.6**
		$\{\hat{\boldsymbol{a}}, \hat{\boldsymbol{a}}\}$	7.6	7.1	7.8	**7.4**
SSVM	1-slack	$\{\hat{\boldsymbol{a}}_{\rm hmm}, \hat{\boldsymbol{a}}\}$	7.6	7.3	7.9	**7.5**

- Just using the HMM segmentation is suboptimal in terms of WER
  - n-slack batch and 1-slack schemes similar to full approach

AURORA-2 - Derivative Score-Spaces

HMM	SDM	$\hat{\boldsymbol{a}}$	Test set A	Test set B	Test set C	Avg
VTS	–	–	9.8	9.1	9.5	9.5
	$\phi^{\mathrm{b}}_{1\mu}$	$\hat{\boldsymbol{a}}_{\mathrm{hmm}}$	7.0	6.6	7.6	**7.0**
		$\hat{\boldsymbol{a}}$	6.8	6.4	7.3	**6.7**
VAT	–	–	8.9	8.3	8.8	8.6
	$\phi^{\mathrm{b}}_{1\mu}$	$\hat{\boldsymbol{a}}_{\mathrm{hmm}}$	6.6	6.5	7.0	**6.6**
		$\hat{\boldsymbol{a}}$	6.2	6.1	6.8	**6.3**
DVAT	–	–	6.7	6.6	7.0	6.7
	$\phi^{\mathrm{b}}_{1\mu}$	$\hat{\boldsymbol{a}}_{\mathrm{hmm}}$	6.1	6.2	6.7	**6.3**
		$\hat{\boldsymbol{a}}$	6.1	6.1	6.6	**6.2**

- Derivative score-spaces ($\phi^{\mathrm{b}}_{1\mu}$) consistent gains over all baseline HMM systems
  - derivative score-space larger (1873 dimensions for each base score-space)
  - adds approximately 50% more parameters to the system

AURORA-4 - Structured SVM Results

- SSVM training configuration:
  - 1-slack variable training
  - prior distribution matched to score-space ϕ_0^{a}, mean set to $1/\mathtt{LM}-\mathtt{scale}$
  - α tied at the monophone-level (47-classes)

Model	Segmentation $\{\mathtt{trn},\mathtt{tst}\}$	Test set A	Test set B	Test set C	Test set D	Avg
HMM	—	7.1	15.3	12.1	23.1	**17.9**
SSVM	$\{\hat{\boldsymbol{a}}_{\mathrm{hmm}},\hat{\boldsymbol{a}}_{\mathrm{hmm}}\}$	7.5	14.3	11.4	21.9	**16.9**
	$\{\hat{\boldsymbol{a}}_{\mathrm{hmm}},\hat{\boldsymbol{a}}\}$	7.4	14.2	11.3	21.9	**16.8**

- SSVM gains over baseline HMM-VTS system
  - disappointing gain from segmentation - though only in test at the moment
  - working on optimal training segmentation as well

AURORA-4 - Derivative Score-Space

Classes	System	Comp tied α	Test set A	Test set B	Test set C	Test set D	Avg
	VTS		7.1	15.3	12.1	23.1	**17.9**
47	$\phi^{\mathrm{b}}_{1\mu}$	yes	7.5	14.1	11.3	21.6	**16.6**
		no	7.4	14.3	11.7	21.9	**16.9**
4020	$\phi^{\mathrm{b}}_{1\mu}$	yes	6.8	13.7	10.6	21.3	**16.2**
		no	6.7	13.5	10.2	21.1	**16.0**

- MPE training for the log-linear model parameters
  - derivative score-spaces give large gains over (ML VTS) baseline
- Component tying important for heavily tied α (47 monophone classes)

AURORA-4 - Derivative Score-Space

System	Test set A	B	C	D	Avg
VTS	7.1	15.3	12.1	23.1	**17.9**
VAT	8.6	13.8	12.0	20.1	**16.0**
DVAT	7.2	12.8	11.5	19.7	**15.3**
VAT+$\phi_0^{\rm b}$	7.7	13.1	11.0	19.5	**15.3**
VAT+$\phi_{1\mu}^{\rm b}$	7.4	12.6	10.7	19.0	**14.8**

- Contrast of DVAT system with log-linear system (4020 classes)
  - single dimension space ($\phi_0^{\rm b}$) with VAT system yields DVAT performance
- Gains from derivative score-space disappointing (limited training data)
  - need to look at DVAT+$\phi_{1\mu}^{\rm b}$ (need to try on more data)

Conclusions

- Combination of generative and discriminative models
  - use generative models to derive features for discriminative model
  - robustness and adaptation achieved by adapting underlying acoustic model
- Derivative features of generative models
  - different conditional independence assumptions to underlying model
  - systematic way to incorporate different dependencies into model
- Large margin training criterion
  - yields structured SVM (use standard optimisation code)
  - still an issue scaling to large tasks/score-spaces

Interesting classifier options - without throwing away HMMs

Acknowledgements

- This work has been funded from the following sources:
  - Cambridge Research Lab, Toshiba Research Europe Ltd
  - EPSRC project - Generative Kernels and Score-Spaces for Classification of Speech

References

[1] J.A. Bilmes, "Graphical models and automatic speech recognition," in *Mathmatical Foundations of Speech and Language Processing*, 2003.

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

[3] H-K. Kuo and Y. Gao, "Maximum entropy direct models for speech recognition," *IEEE Transactions Audio Speech and Language Processing*, 2006.

[4] A. Gunawardana, M. Mahajan, A. Acero, and J.C. Platt, "Hidden conditional random fields for phone classification," in *Interspeech*, 2005.

[5] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[6] C. Cortes, P. Haffner, and M. Mohri, "Weighted automata kernels - general framework and algorithms," in *Proc. Eurospeech*, 2003.

[7] Layton MI and MJF Gales, "Acoustic modelling using continuous rational kernels," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, August 2007.

[8] N.D. Smith and M.J.F. Gales, "Speech recognition using SVMs," in *Advances in Neural Information Processing Systems*, 2001.

[9] T. Jaakkola and D. Haussler, "Exploiting generative models in disciminative classifiers," in *Advances in Neural Information Processing Systems 11*, S.A. Solla and D.A. Cohn, Eds. 1999, pp. 487–493, MIT Press.

[10] A. Ragni and M. J. F. Gales, "Structured discriminative models for noise robust continuous speech recognition," in *ICASSP*, 2011, pp. 4788–4791.

[11] R. C. van Dalen, A. Ragni, and M. J. F. Gales, "Efficient decoding with continuous rational kernels using the expectation semiring," Tech. Rep. CUED/F-INFENG/TR.674, 2012.

[12] M J F Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[13] 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.

[14] P.S. Gopalakrishnan, D. Kanevsky, A. Nádas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Trans. Information Theory*, 1991.

[15] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Computer Speech & Language*, vol. 16, pp. 25–47, 2002.

[16] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing*, 1992.

[17] J. Kaiser, B. Horvat, and Z. Kacic, "A novel loss function for the overall risk criterion based discriminative training of HMM models," in *Proc. ICSLP*, 2000.

[18] W. Byrne, "Minimum Bayes risk estimation and decoding in large vocabulary continuous speech recognition," *IEICE Special Issue on Statistical Modelling for Speech Recognition*, 2006.

[19] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Orlando, FL, May 2002.

[20] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University, 2004.

[21] F. Sha and L.K. Saul, "Large margin gaussian mixture modelling for phonetic classification and recognition," in *ICASSP*, 2007.

[22] J. Li, M. Siniscalchi, and C-H. Lee, "Approximate test risk minimization theough soft margin training," in *ICASSP*, 2007.

[23] G Heigold, T Deselaers, R Schluter, and H Ney, "Modified MMI/MPE: A direct evaluation of the margin in speech recognition," in *Proc. ICML*, 2008.

[24] G Saon and D Povey, "Penalty function maximization for large margin HMM training," in *Proc. Interspeech*, 2008.

[25] S.-X. Zhang, Anton 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.

[26] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, 2005.

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

[28] S.-X. Zhang and M. J. F. Gales, "Extending noise robust structured support vector machines to larger vocabulary tasks," in *Proc. ASRU*, 2011.

[29] Chun-Nam Yu and Thorsten Joachims, "Learning structural SVMs with latent variables," in *Proceedings of ICML*, 2009.