Augmented Statistical Models for Speech Recognition

Mark Gales & Martin Layton

31 August 2005

Cambridge University Engineering Department

Trajectory Models For Speech Processing Workshop

Overview

- Dependency Modelling in Speech Recognition:
  - latent variables
  - exponential family
- Augmented Statistical Models
  - augments standard models, e.g. GMMs and HMMs
  - extends representation of dependencies
- Augmented Statistical Model Training
  - use maximum margin training
  - relationship to "dynamic" kernels
- Preliminary LVCSR experiments

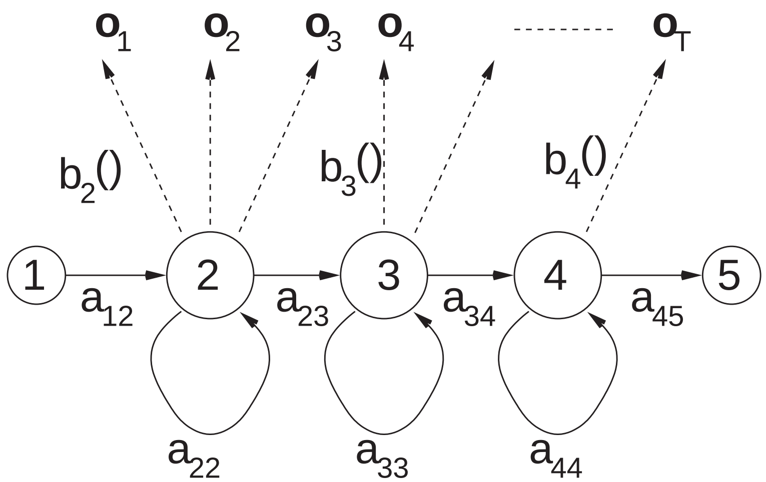
Dependency Modelling

- Speech data is dynamic - observations are not of a fixed length
- Dependency modelling essential part of speech recognition:

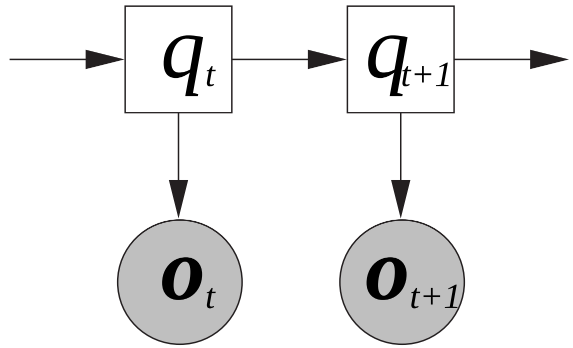
$$p(\boldsymbol{o}_1, \ldots, \mathbf{o}_T; \boldsymbol{\lambda}) = p(\boldsymbol{o}_1; \boldsymbol{\lambda}) p(\boldsymbol{o}_2|\boldsymbol{o}_1; \boldsymbol{\lambda}) \ldots p(\boldsymbol{o}_T|\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{T-1}; \boldsymbol{\lambda})$$

  - impractical to directly model in this form
  - make extensive use of conditional independence
- Two possible forms of conditional independence used:
  - observed variables
  - latent (unobserved) variables
- Even given dependency (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:

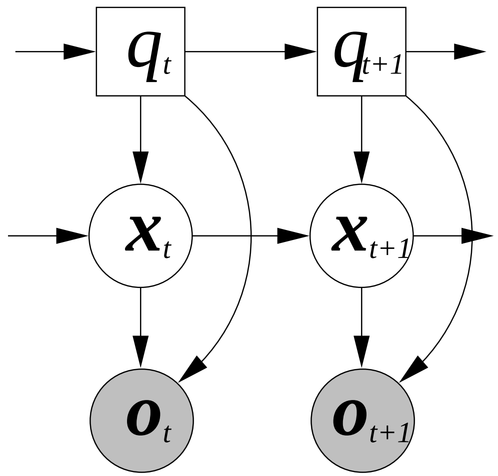
  circles - continuous variables   shaded - observed variables
  squares - discrete variables   non-shaded - unobserved variables

- Observations conditionally independent of other observations given state.
- States conditionally independent of other states given previous states.
- Poor model of the speech process - piecewise constant state-space.

Dependency Modelling using Latent Variables

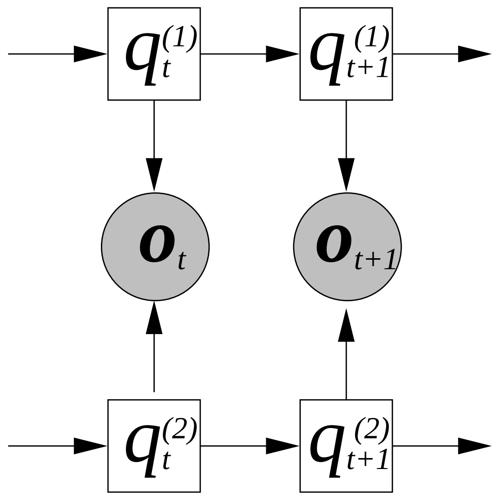
Switching linear dynamical system:

- discrete and continuous state-spaces
- observations conditionally independent given continuous and discrete state;
- approximate inference required
  $\Rightarrow$ Rao-Blackwellised Gibbs sampling.



Multiple data stream DBN:

- e.g. factorial HMM/mixed memory model;
- asynchronous data common:
  - speech and video/noise;
  - speech and brain activation patterns.
- observation depends on state of both streams

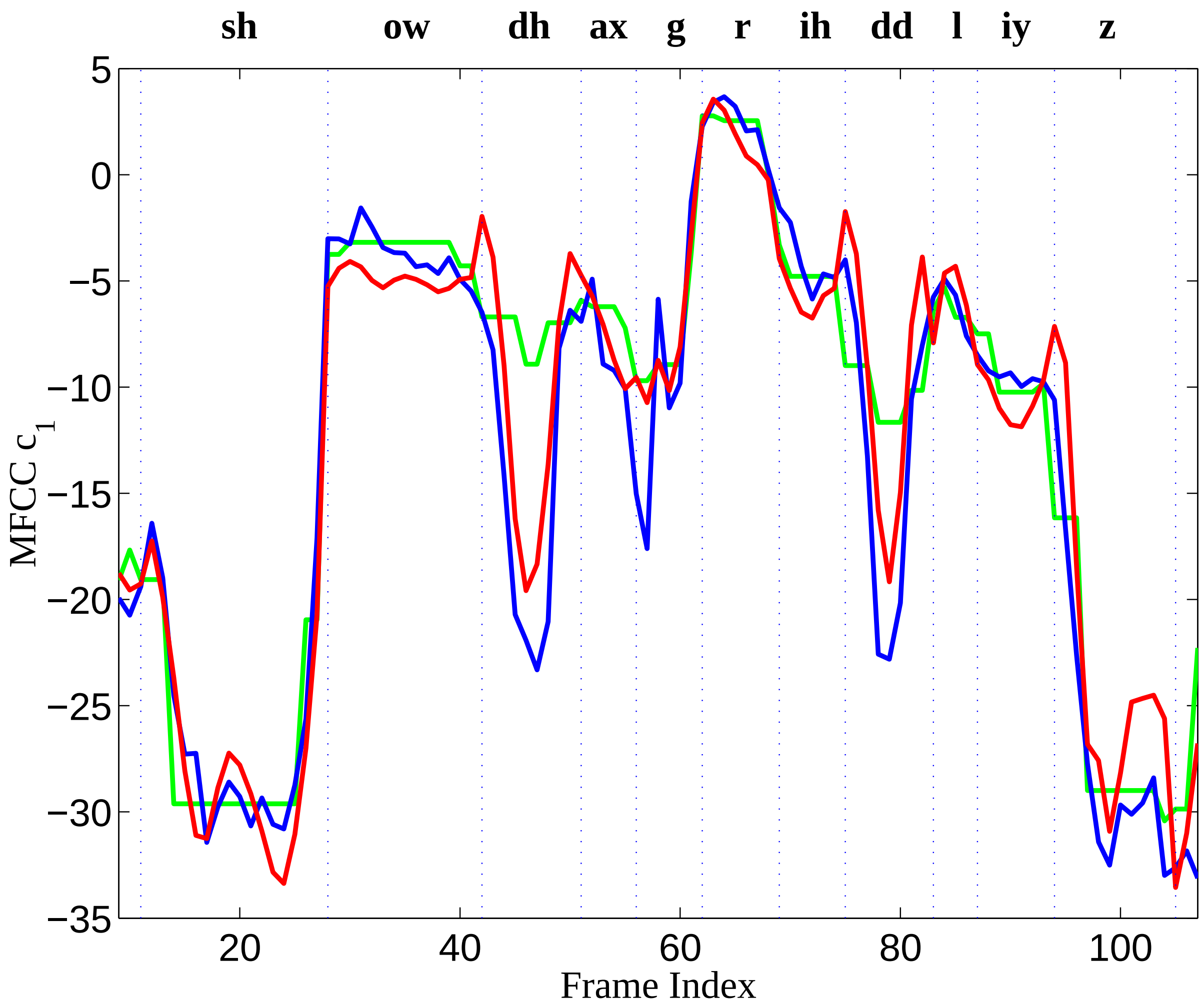


SLDS Trajectory Modelling

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

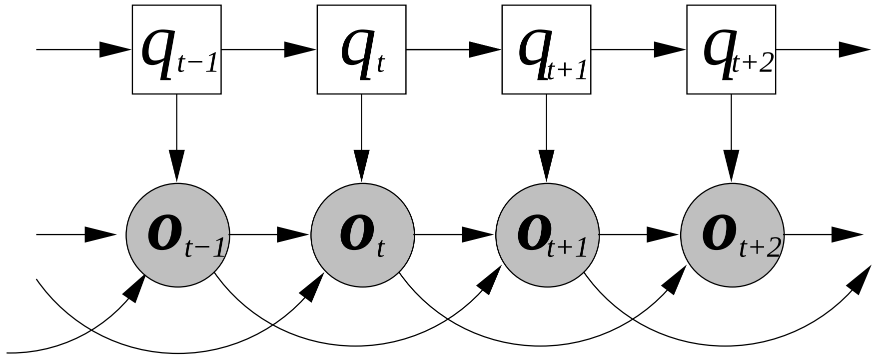
Legend

- True
- HMM
- SLDS



- Unfortunately doesn't currently classify better than an HMM!

Dependency Modelling using Observed Variables



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

$$p(\mathbf{O};\boldsymbol{\alpha}) = \frac{1}{\tau}h(\mathbf{O})\exp\left(\boldsymbol{\alpha}'\mathbf{T}(\mathbf{O})\right)$$

  - $h(\mathbf{O})$ is the reference distribution; τ is the normalisation term
  - $\boldsymbol{\alpha}$ are the natural parameters
  - the function $\mathbf{T}(\mathbf{O})$ is a sufficient statistic.

- What is the appropriate form of statistics $(\mathbf{T}(\mathbf{O}))$ - needs DBN to be known
  - for example in diagram, $T(\mathbf{O}) = \sum_{t=1}^{T-2} \mathbf{o}_t\mathbf{o}_{t+1}\mathbf{o}_{t+2}$

Constrained Exponential Family

- Could hypothesise all possible dependencies and prune
  - discriminative pruning found to be useful (buried Markov models)
  - impractical for wide range (and lengths) of dependencies
- Consider constrained form of statistics
  - local exponential approximation to the reference distribution
  - ρ^{th}-order differential form considered (related to Taylor-series)
- Distribution has two parts
  - reference distribution defines latent variables
  - local exponential model defines statistics ($\mathbf{T}(\mathbf{O})$)
- Slightly more general form is the augmented statistical model
  - train all the parameters (including the reference, base, distribution)

Augmented Statistical Models

- Augmented statistical models (related to fibre bundles)

$$p(\mathbf{O};\boldsymbol{\lambda},\boldsymbol{\alpha}) = \frac{1}{\tau}\check{p}(\mathbf{O};\boldsymbol{\lambda})\exp\left(\boldsymbol{\alpha}'\begin{bmatrix} \boldsymbol{\nabla}_{\lambda}\log(\check{p}(\mathbf{O};\boldsymbol{\lambda})) \\ \frac{1}{2!}\text{vec}\left(\boldsymbol{\nabla}^2_{\lambda}\log(\check{p}(\mathbf{O};\boldsymbol{\lambda}))\right) \\ \vdots \\ \frac{1}{\rho!}\text{vec}\left(\boldsymbol{\nabla}^{\rho}_{\lambda}\log(\check{p}(\mathbf{O};\boldsymbol{\lambda}))\right) \end{bmatrix}\right)$$

- Two sets of parameters
  - $\boldsymbol{\lambda}$ - parameters of base distribution ($\check{p}(\mathbf{O};\boldsymbol{\lambda})$)
  - $\boldsymbol{\alpha}$ - natural parameters of local exponential model
- Normalisation term τ ensures that

$$\int_{\mathcal{R}^{nT}} p(\mathbf{O};\boldsymbol{\lambda},\boldsymbol{\alpha})d\mathbf{O} = 1; \qquad p(\mathbf{O};\boldsymbol{\lambda},\boldsymbol{\alpha}) = \overline{p}(\mathbf{O};\boldsymbol{\lambda},\boldsymbol{\alpha})/\tau$$

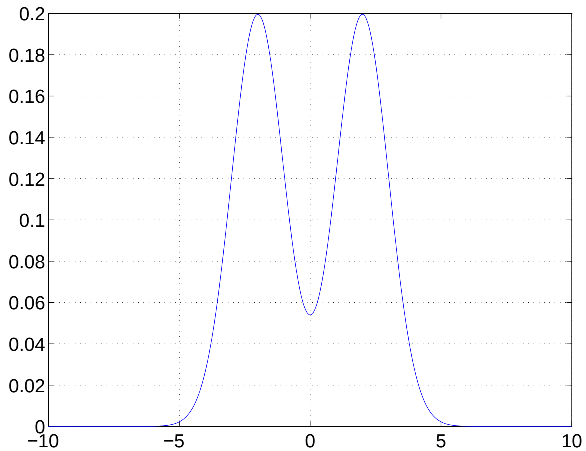
  - can be very complex to estimate

Augmented Gaussian Mixture Model

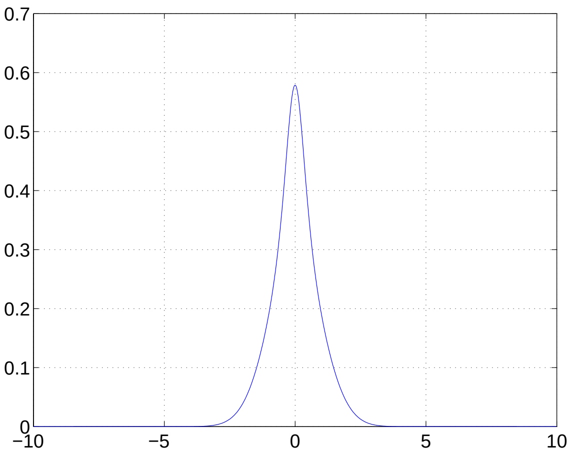
- Use a GMM as the base distribution: $\check{p}(\boldsymbol{o};\boldsymbol{\lambda}) = \sum_{m=1}^{M} c_m \mathcal{N}(\boldsymbol{o};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)$
  - considering only the first derivatives of the means

$$p(\boldsymbol{o};\boldsymbol{\lambda},\boldsymbol{\alpha}) = \frac{1}{\tau}\sum_{m=1}^{M} c_m \mathcal{N}(\boldsymbol{o};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)\exp\left(\sum_{n=1}^{M} P(n|\boldsymbol{o};\boldsymbol{\lambda})\boldsymbol{\alpha}_n'\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_n)\right)$$

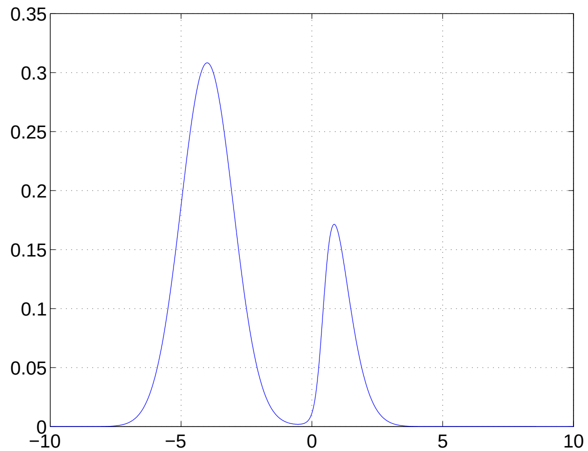
- Simple two component one-dimensional example:



$\boldsymbol{\alpha} = [0.0, 0.0]'$



$\boldsymbol{\alpha} = [-1.0, -1.0]'$



$\boldsymbol{\alpha} = [1.0, -1.0]'$

Augmented Model Dependencies

- If the base distribution is a mixture of members of the exponential family

$$\check{p}(\mathbf{O};\boldsymbol{\lambda}) = \prod_{t=1}^{T}\sum_{m=1}^{M} c_m \exp\left(\sum_{j=1}^{J}\lambda_j^{(m)}T_j^{(m)}(\boldsymbol{o}_t)\right)/\tau^{(m)}$$

  - consider a first order differential

$$\frac{\partial}{\partial\lambda_k^{(n)}}\log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda})\right) = \sum_{t=1}^{T} P(n|\mathbf{o}_t;\boldsymbol{\lambda})\left(T_k^{(n)}(\mathbf{o}_t) - \frac{\partial}{\partial\lambda_k^{(n)}}\log(\tau^{(n)})\right)$$

- Augmented models of this form
  - keep independence assumptions of the base distribution
  - remove conditional independence assumptions of the base model
    - the local exponential model depends on a posterior ...

- Augmented GMMs do not improve temporal modelling ...

Augmented HMM Dependencies

- For an HMM: $\check{p}(\mathbf{O};\boldsymbol{\lambda}) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left\{ \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} \left(\sum_{m\in\theta_t} c_m \mathcal{N}(\mathbf{o}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \right) \right\}$
- Derivative depends on posterior, $\gamma_{jm}(t) = P(\theta_t = \{s_j, m\}|\mathbf{O};\boldsymbol{\lambda})$,

$$T(\mathbf{O}) = \sum_{t=1}^{T} \gamma_{jm}(t) \boldsymbol{\Sigma}_{jm}^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_{jm})$$

  - posterior depends on complete observation sequence, $\mathbf{O}$
  - introduces dependencies beyond conditional state independence
  - compact representation of effects of all observations

- Higher-order derivatives incorporate higher-order dependencies
  - increasing order of derivatives - increasingly powerful trajectory model
  - systematic approach to incorporating additional dependencies

Augmented Model Summary

- Extension to standard forms of statistical model
- Consists of two parts:
  - base distribution determines the latent variables
  - local exponential distribution augments base distribution
- Base distribution:
  - standard form of statistical model
  - examples considered: Gaussian mixture models and hidden Markov models
- Local exponential distribution:
  - currently based on ρ^{th}-order differential form
  - gives additional dependencies not present in base distribution
- Normalisation term may be highly complex to calculate
  - maximum likelihood training may be very awkward

Augmented Model Training

- Only consider simplified two-class problem
- Bayes' decision rule for binary case (prior $P(\omega_1)$ and $P(\omega_2)$):

$$\frac{P(\omega_1)\tau^{(2)}\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)},\boldsymbol{\alpha}^{(1)})}{P(\omega_2)\tau^{(1)}\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)},\boldsymbol{\alpha}^{(2)})} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 1; \qquad \frac{1}{T}\log\left(\frac{\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)},\boldsymbol{\alpha}^{(1)})}{\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)},\boldsymbol{\alpha}^{(2)})}\right) + b \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 0$$

  - $b = \frac{1}{T}\log\left(\frac{P(\omega_1)\tau^{(2)}}{P(\omega_2)\tau^{(1)}}\right)$ - no need to explicitly calculate τ
- Can express decision rule as the following scalar product

$$\left[\begin{array}{c}\mathbf{w}\\ b\end{array}\right]'\left[\begin{array}{c}\boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda})\\ 1\end{array}\right] \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 0$$

  - form of score-space and linear decision boundary
- Note - restrictions on α's to ensure a valid distribution.

Augmented Model Training - Binary Case (cont)

- Generative score-space is given by (first order derivatives)

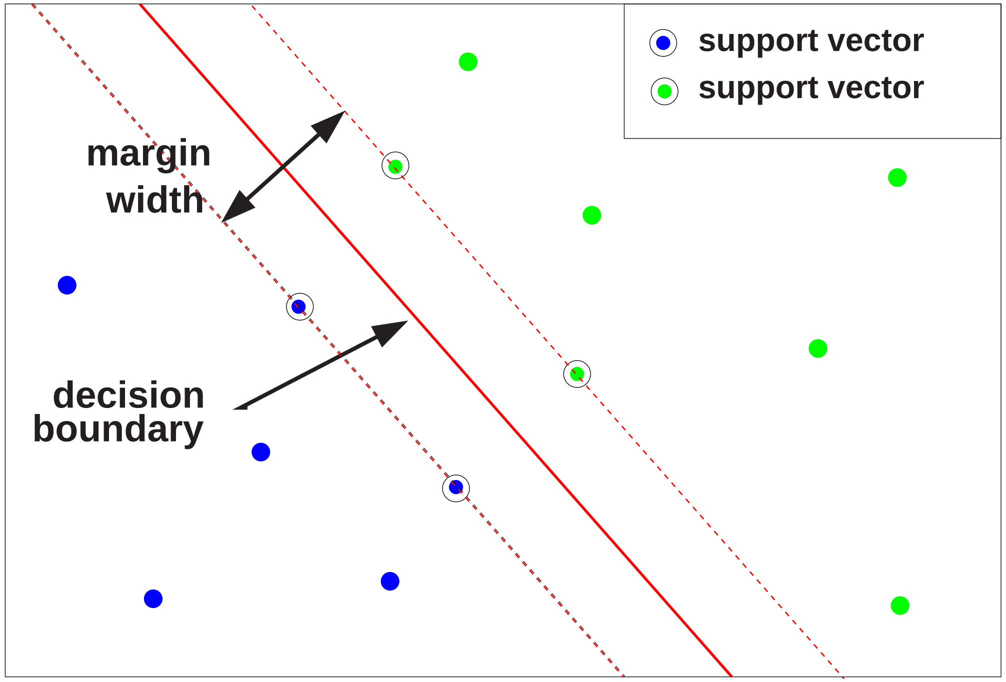
$$\phi(\mathbf{O};\boldsymbol{\lambda}) = \frac{1}{T}\left[\begin{array}{c} \log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)})\right) - \log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)})\right) \\ \boldsymbol{\nabla}_{\lambda^{(1)}} \log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)})\right) \\ -\boldsymbol{\nabla}_{\lambda^{(2)}} \log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)})\right) \end{array}\right]$$

  - only a function of the base-distribution parameters $\boldsymbol{\lambda}$
- Linear decision boundary given by

$$\mathbf{w}' = \left[\begin{array}{ccc} 1 & \boldsymbol{\alpha}^{(1)\prime} & \boldsymbol{\alpha}^{(2)\prime} \end{array}\right]'$$

  - only a function of the exponential model parameters $\boldsymbol{\alpha}$
- Bias is represented by b - depends on both $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$
- Possibly large number of parameters for linear decision boundary
  - maximum margin (MM) estimation good choice - SVM training

Support Vector Machines



- SVMs are a maximum margin, binary, classifier:
  - related to minimising generalisation error;
  - unique solution (compare to neural networks);
  - may be kernelised - training/classification a function of dot-product $(\mathbf{x}_i.\mathbf{x}_j)$.
- Can be applied to speech - use a kernel to map variable data to a fixed length.

Estimating Model Parameters

- Two sets of parameters to be estimated using training data $\{\mathbf{O}_1, \dots, \mathbf{O}_n\}$:
  - base distribution (Kernel) $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}\}$
  - direction of decision boundary ($y_i \in \{-1, 1\}$ label of training example)

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i^{\mathtt{svm}} y_i \mathbf{G}^{-1} \boldsymbol{\phi}(\mathbf{O}_i; \boldsymbol{\lambda})$$

$\boldsymbol{\alpha}^{\mathtt{svm}} = \{\alpha_1^{\mathtt{svm}}, \dots, \alpha_n^{\mathtt{svm}}\}$ set of SVM Lagrange multipliers
$\mathbf{G}$ associated with distance metric for SVM kernel

- Kernel parameters may be estimated using:
  - maximum likelihood (ML) training;
  - discriminative training, e.g. maximum mutual information (MMI)
  - maximum margin (MM) training (consistent with α's).

SVMs and Class Posteriors

- Common objection to SVMs - no probabilistic interpretation
  - use of additional sigmoidal mapping/relevance vector machines
- Generative kernels - distance from the decision boundary is the posterior ratio

$$\frac{1}{w_1}\left(\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix}'\begin{bmatrix}\boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda})\\ 1\end{bmatrix}\right)=\frac{1}{T}\log\left(\frac{P(\omega_1|\mathbf{O})}{P(\omega_2|\mathbf{O})}\right)$$

  - w_1 is required to ensure first element of $\mathbf{w}$ is 1
  - augmented version of the kernel PDF becomes the class-conditional PDF
- Decision boundary also yields the exponential natural parameters

$$\begin{bmatrix}1\\ \boldsymbol{\alpha}^{(1)}\\ \boldsymbol{\alpha}^{(2)}\end{bmatrix}=\frac{1}{w_1}\mathbf{w}=\frac{1}{w_1}\sum_{i=1}^{n}\alpha_i^{\texttt{svm}}y_i\mathbf{G}^{-1}\boldsymbol{\phi}(\mathbf{O}_i;\boldsymbol{\lambda})$$

Relationship to "Dynamic Kernels"

- Dynamic kernels popular for applying SVMs to sequence data
- Two standard kernels, related to generative kernels are:
  - Fisher kernel
  - Marginalised count kernel
- Fisher Kernel:
  - equivalent to generative kernel with two base distributions the same

$$\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)}) = \check{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)})$$

    and only using first order derivatives.
  - Fisher kernel useful with large amounts of unsupervised data.
- Fisher kernel can also be described as a marginalised count kernel.

Marginalised Count Kernel

- Another related kernel is the marginalised count kernel.
  - used for discrete data (bioinformatics applications)
  - score space element for second-order token pairings $\mathtt{ab}$ and states $\theta_{\mathtt{a}}\theta_{\mathtt{b}}$

$$\phi(\mathbf{O};\boldsymbol{\lambda}) = \sum_{t=1}^{T-1} \mathcal{I}(\mathbf{o}_t = \mathtt{a}, \mathbf{o}_{t+1} = \mathtt{b}) P(\theta_t = \theta_a, \theta_{t+1} = \theta_b | \mathbf{O};\boldsymbol{\lambda})$$

compare to an element of the second derivative of PMF of a discrete HMM

$$\phi(\mathbf{O};\boldsymbol{\lambda}) = \sum_{t=1}^{T} \sum_{\tau=1}^{T} \mathcal{I}(\mathbf{o}_t = \mathtt{a}, \mathbf{o}_\tau = \mathtt{b}) P(\theta_t = \theta_a, \theta_\tau = \theta_b | \mathbf{O};\boldsymbol{\lambda}) + \ldots$$

  - higher order derivatives yields higher order dependencies
  - generative kernels allow “continuous” forms of count kernels

ISOLET E-Set Experiments

- ISOLET - isolated letters from American English
  - E-set subset {B,C,D,E,G,P,T,V,Z} - highly confusable
- Standard features MFCC_E_D_A, 10 emitting state HMM 2 components/state
  - first-order mean derivative score-space for A-HMM

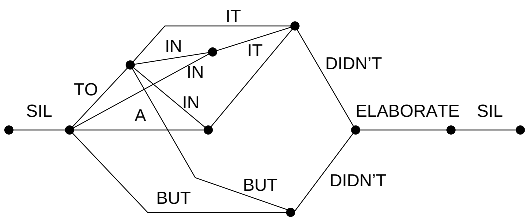
Classifier	Training		WER
	Base (λ)	Aug (α)	(%)
HMM	ML	—	8.7
	MMI	—	4.8
A-HMM	ML	MM	5.0
	MMI	MM	4.3

- Augmented HMMs outperform HMMs for both ML and MMI trained systems.
  - best performance using selection/more complex model - 3.2%

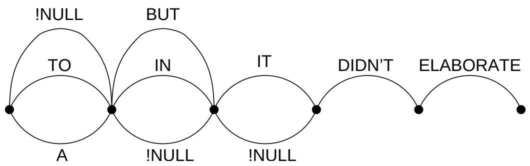
Binary Classifiers and LVCSR

- Many classifiers(e.g. SVMs) are inherently binary:
  - speech recognition has a vast number of possible classes;
  - how to map to a simple binary problem?

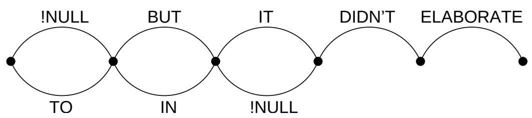
- Use pruned confusion networks (Venkataramani et al ASRU 2003):



Word lattice



Confusion Network



Pruned confusion network

  - use standard HMM decoder to generate word lattice;
  - generate confusion networks (CN) from word lattice
    - ∗ gives posterior for each arc being correct;
  - prune CN to a maximum of two arcs (based on posteriors).

LVCSR Experimental Setup

- HMMs trained on 400hours of conversational telephone speech (`fsh2004sub`):
  - standard CUHTK CTS frontend (CMN/CVN/VTLN/HLDA)
  - state-clustered triphones (~ 6000 states, ~ 28 components/state);
  - maximum likelihood training
- Confusion networks generated for `fsh2004sub`
- Perform 8-fold cross-validation on 400 hours training data:
  - use CN to obtain highly confusable common word pairs
  - ML/MMI-trained word HMMs - 3 emitting states, 4 components per state
  - first-order derivatives (prior/mean/variance) score-space A-HMMs
- Evaluation on held-out data (`eval03`)
  - 6 hours of test data
  - decoded using LVCSR trigram language model
  - baseline using confusion network decoding

8-Fold Cross-Validation LVCSR Results

Word Pair (Examples/class)	Classifier	Training: Base (λ)	Training: Aug (α)	WER (%)
CAN/CAN'T (3761)	HMM	ML	—	11.0
	HMM	MMI	—	10.4
	A-HMM	ML	MM	9.5
KNOW/NO (4475)	HMM	ML	—	27.7
	HMM	MMI	—	27.1
	A-HMM	ML	MM	23.8

- A-HMM outperforms both ML and MMI HMM
  - also outperforms using "equivalent" number of parameters
  - difficult to split dependency modelling gains from change in training criterion

Incorporating Posterior Information

- Useful to incorporate arc log-posterior $(\mathcal{F}(\omega_1), \mathcal{F}(\omega_2))$ into decision process
  - posterior contains e.g. N-gram LM, cross-word context acoustic information
- Two simple approaches:
  - combination of two as independent sources (β empirically set)

$$\frac{1}{T}\log\left(\frac{\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)},\boldsymbol{\alpha}^{(1)})}{\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)},\boldsymbol{\alpha}^{(2)})}\right) + b + \beta\left(\mathcal{F}(\omega_1) - \mathcal{F}(\omega_2)\right) \mathop{\gtrless}_{\omega_2}^{\omega_1} 0$$

  - incorporate posterior into score-space (β obtained from decision boundary)

$$\boldsymbol{\phi}^{\text{cn}}(\mathbf{O};\boldsymbol{\lambda}) = \left[\begin{array}{c} \mathcal{F}(\omega_1) - \mathcal{F}(\omega_2) \\ \boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda}) \end{array}\right]$$

- Incorporating in score-space requires consistency between train/test posteriors

Evaluation Data LVCSR Results

- Baseline performance using Viterbi and Confusion Network decoding

Decoding	trigram LM
Viterbi	30.8
Confusion Network	30.1

- Rescore word-pairs using 3-state/4-component A-HMM+βCN

SVM Rescoring	#corrected/#pairs	% corrected
10 SVMs	56/1250	4.5%

  - β roughly set - error rate relatively insensitive to exact value
  - only 1.6% of 76157 hypothesised words rescored - more SVMs required!

- More suitable to smaller tasks, e.g. digit recognition in low SNR conditions

Summary

- Dependency modelling for speech recognition
  - use of latent variables
  - use of sufficient statistics from the data
- Augmented statistical models
  - allows simple combination of latent variables and sufficient statistics
  - use of constrained exponential model to define statistics
  - simple to train using an SVM - related to various "dynamic" kernels
- Preliminary results of a large vocabulary speech recognition task
  - SVMs/Augmented models possibly useful for speech recognition
- Current work
  - maximum margin "kernel parameter" estimation
  - use of weighted finite-state transducers for higher-order derivative calculation
  - modified "variable-margin" training (constrains $w_1 = 1$)