Modelling Dependencies in Sequence Classification: Augmented Statistical Models

Mark Gales - work with Martin Layton

9 November 2006

Cambridge University Engineering Department

University of East Anglia Seminar

Overview

- Dependency Modelling in Sequence Data:
- 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
- Conditional augmented models
  - "relationship" to CRFs/HCRFs
- Speaker verification and ASR experiments

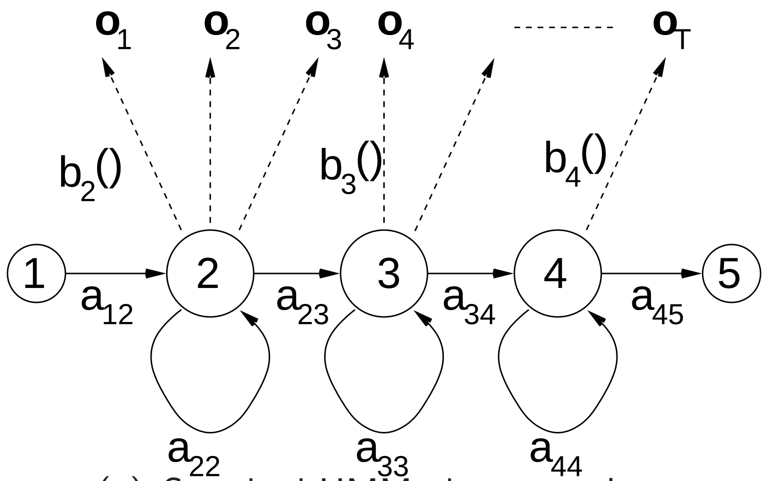
Dependency Modelling

- Range of applications require classification of sequence data:
  - observation sequences are not of a fixed length
  - examples include text/speech processing, computational biology etc
- Dependency modelling essential part of modelling sequence data:

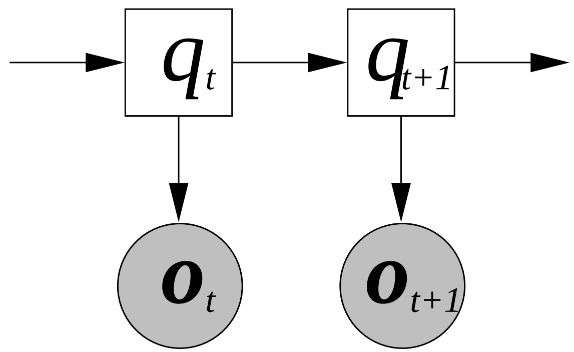
$$p(\boldsymbol{o}_1, \ldots, \mathbf{o}_T; \boldsymbol{\lambda}) = p(\boldsymbol{o}_1; \boldsymbol{\lambda})p(\boldsymbol{o}_2|\boldsymbol{o}_1; \boldsymbol{\lambda}) \ldots p(\boldsymbol{o}_T|\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{T-1}; \boldsymbol{\lambda})$$

  - 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

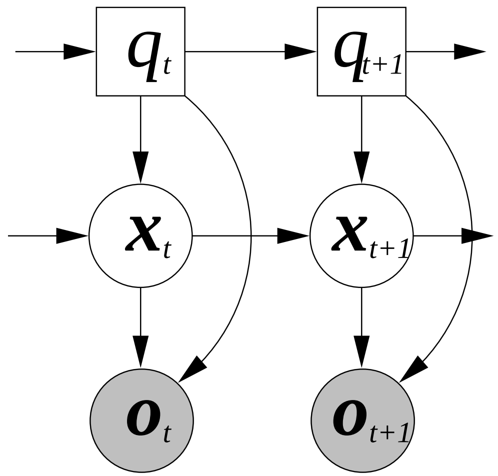
- Notation for DBNs:

  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.

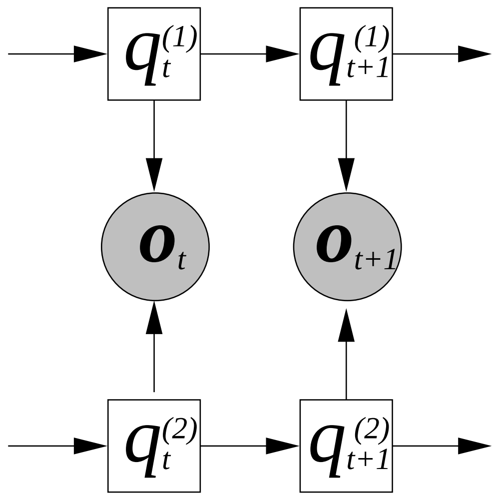
Dependency Modelling using Latent Variables

Switching linear dynamical system:



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

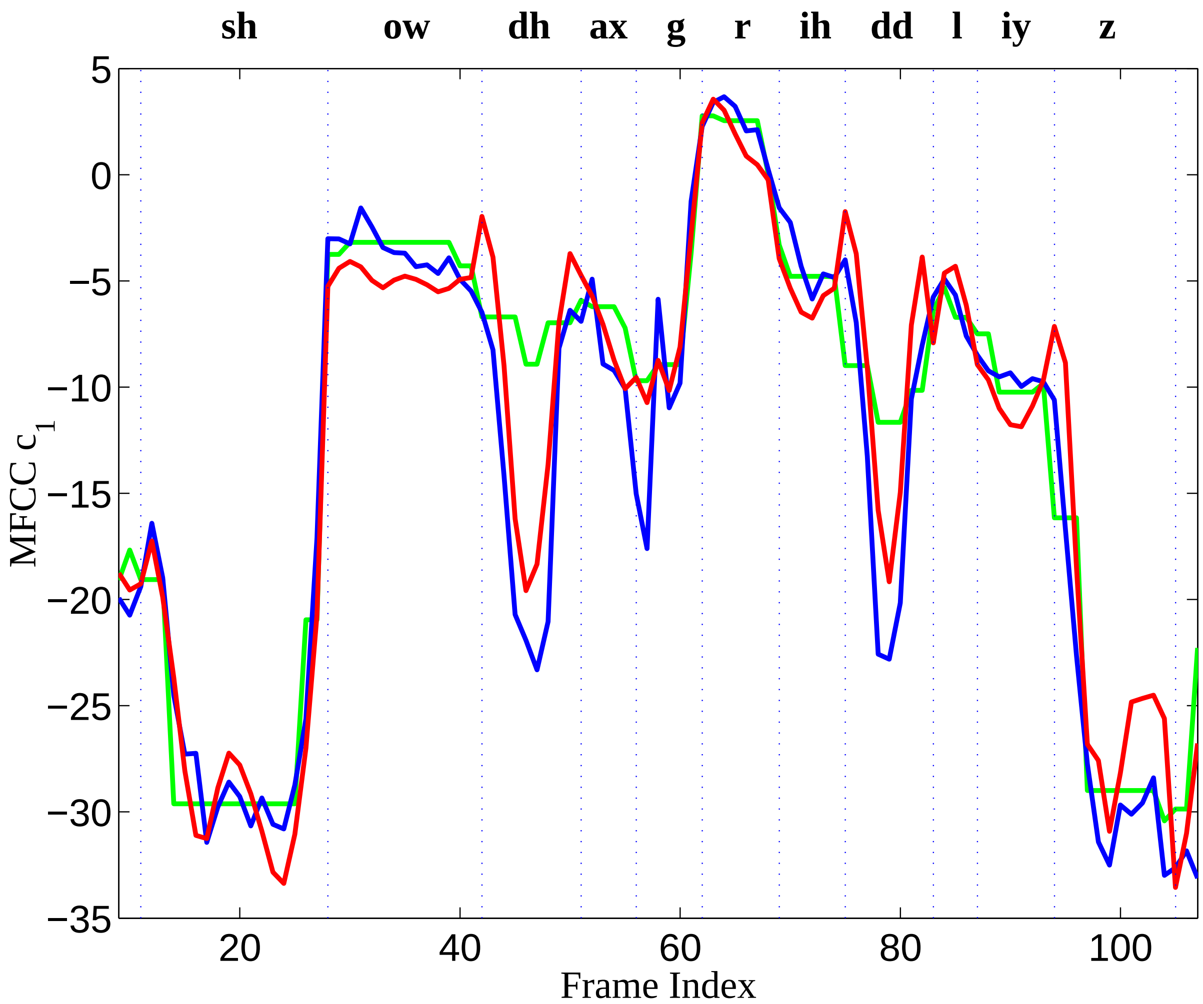
Multiple data stream DBN:



- 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 Speech Trajectory Modelling

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



Legend

- True
- HMM
- SLDS

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

Linear Transform as the Latent Variable

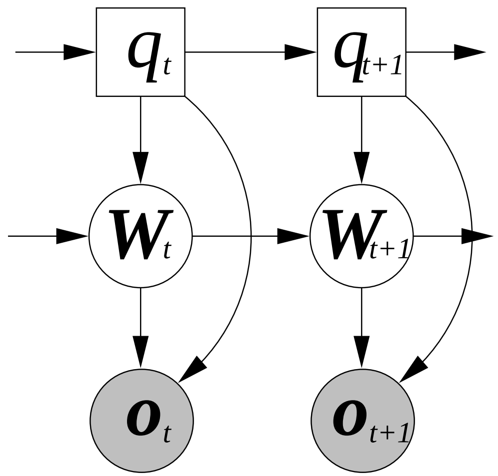
- Linear adaptation in speech recognition can be viewed as a latent variable
  - interesting interaction of latent variables and distribution

"Adaptive" HMMs:

- impact of "continuous-space" on distribution

$$p(\mathbf{o}_t|\mathbf{W}_t, q_t) = \sum_{m=1}^{M} c_m(\mathbf{o}_t; \mathbf{W}_t\boldsymbol{\mu}_m^{(q_t)}, \boldsymbol{\Sigma}_m^{(q_t)})$$

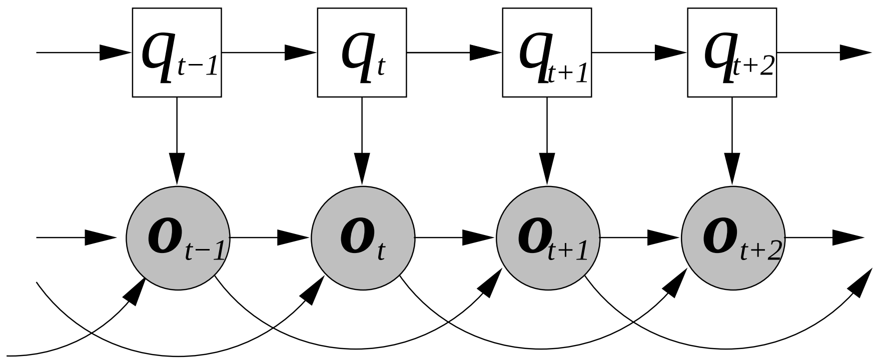
- restrict $\mathbf{W}_{t+1} = \mathbf{W}_t$ (homogeneous blocks)



- Inference performed by marginalising over prior distribution $p(\mathbf{W})$
  - approximate inference required, e.g. lower-bound Variational Bayes

Adaptive HMMs works for speech recognition!

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 $(\mathbf{T}(\mathbf{O};\boldsymbol{\lambda}))$
- 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

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

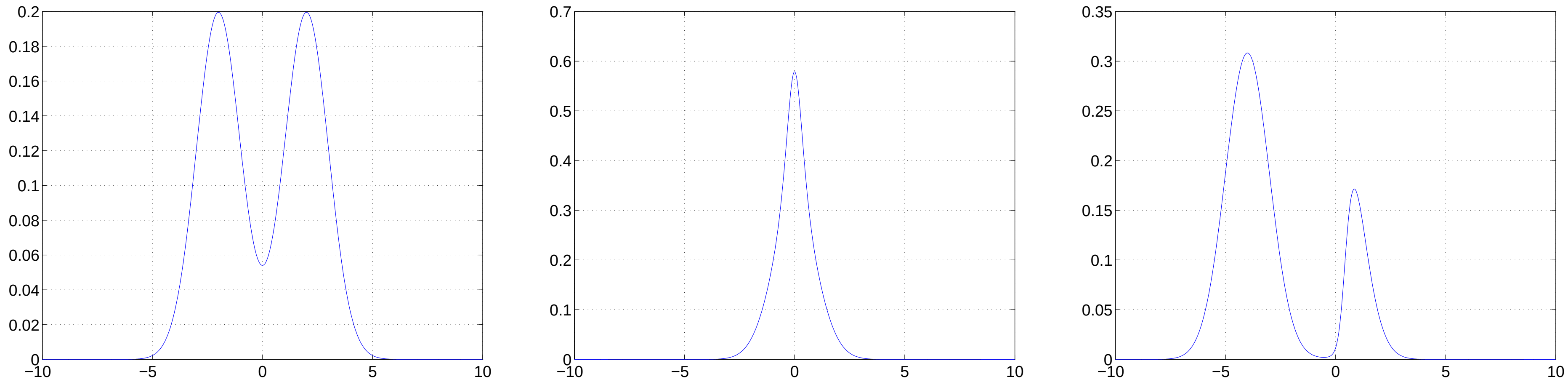
  - can be very complex to estimate

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

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

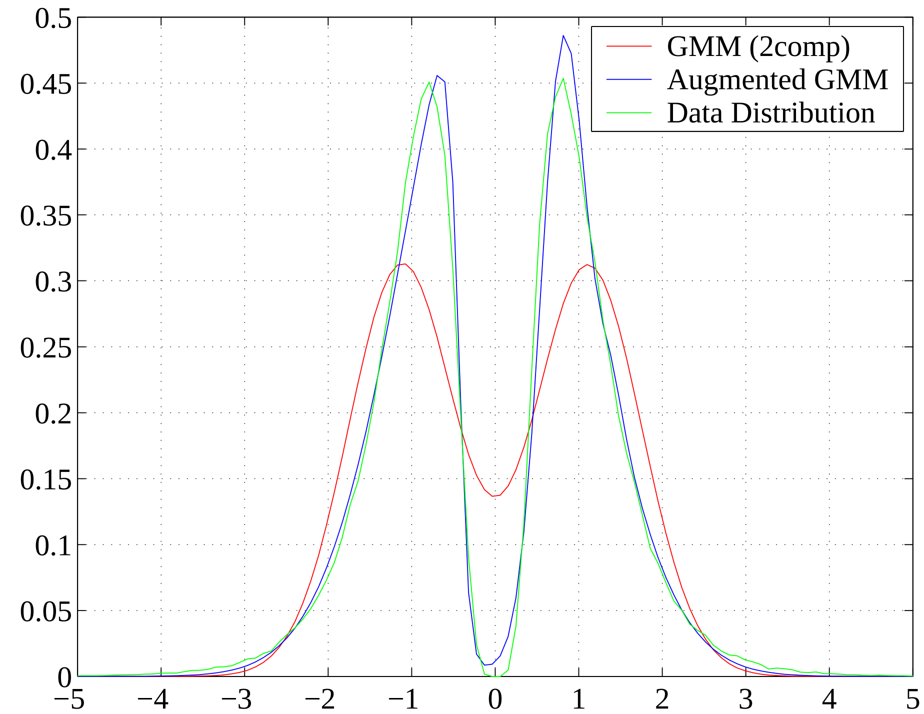
- Simple two component one-dimensional example:



$\boldsymbol{\alpha} = [0.0, 0.0]'$ $\boldsymbol{\alpha} = [-1.0, -1.0]'$ $\boldsymbol{\alpha} = [1.0, -1.0]'$

Augmented Gaussian Mixture Model Example

- Maximum likelihood training of A-GMM on symmetric log-normal data



  - 2-component base-distribution (poor model of data)
  - A-GMM better model of distribution (log-likelihood -1.45 vs -1.59 GMM)
  - approx. symmetry obtained without symmetry in parameters!

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

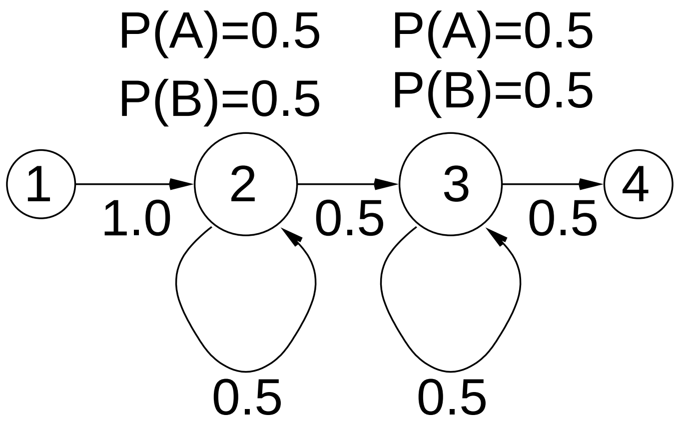
- For an HMM: $\check{p}(\mathbf{O};\boldsymbol{\lambda}) = \sum_{\mathbf{q}\in\boldsymbol{\Theta}} \left\{ \prod_{t=1}^{T} a_{q_{t-1}q_t} \left(\sum_{m\in q_t} c_m \mathcal{N}(\mathbf{o}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \right) \right\}$
- Derivative depends on posterior, $\gamma_{jm}(t)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

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

  - 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

Discrete Augmented Model Example

- Consider a simple 2-class, 2-symbol $\{\mathtt{A}, \mathtt{B}\}$ 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
$\nabla_{2A}\nabla'_{2A}$	-3.83	0.17	-3.28	-0.61
$\nabla_{2A}\nabla'_{3A}$	-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

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

- Normalisation term makes ML training of augmented models difficult
  - use discriminative training approaches instead

- Two forms of discriminative training have been examined:

- Maximum Margin based approaches:
  - implemented using Support Vector Machines (SVMs)
  - applicable to binary classification tasks

- Conditional Maximum Likelihood based approaches:
  - directly applicable to multi-class problems

Augmented Model Training- Binary Case

- 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

$$\begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}' \begin{bmatrix} \boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda}) \\ 1 \end{bmatrix} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 0$$

  - 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)

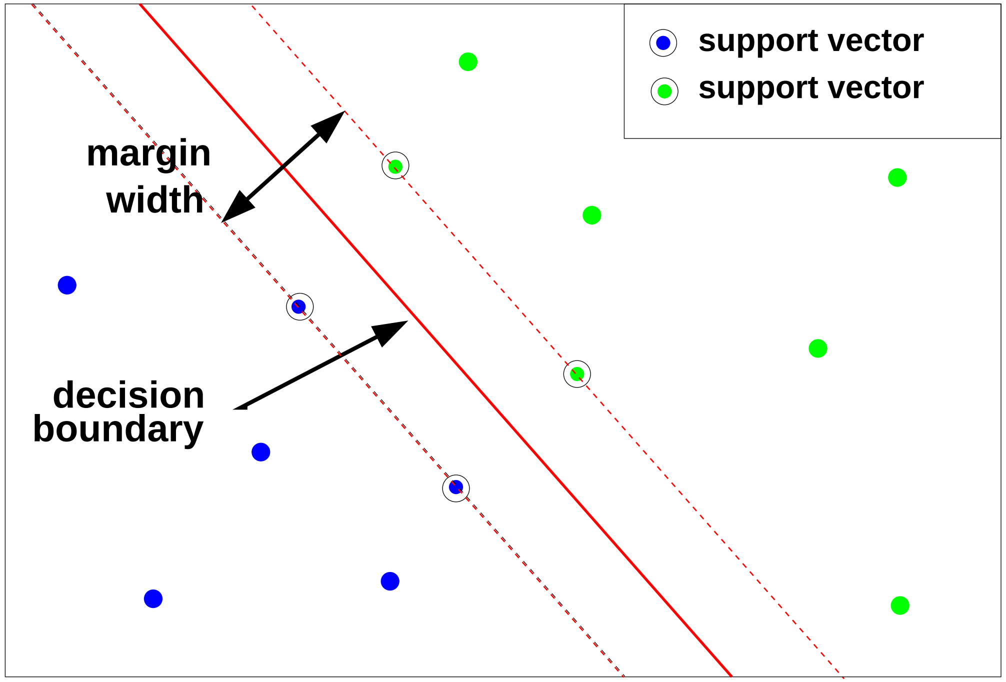
$$\boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda}) = \frac{1}{T}\left[\begin{array}{c} \log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)})\right) - \log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)})\right) \\ \boldsymbol{\nabla}_{\lambda^{(1)}}\log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)})\right)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

  - 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 $\{\mathbf{O}_1, \ldots, \mathbf{O}_n\}$:
  - base distribution (Kernel) $\boldsymbol{\lambda} = \left\{\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}\right\}$
  - direction of decision boundary ($y_i \in \{-1, 1\}$ label of training example)

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

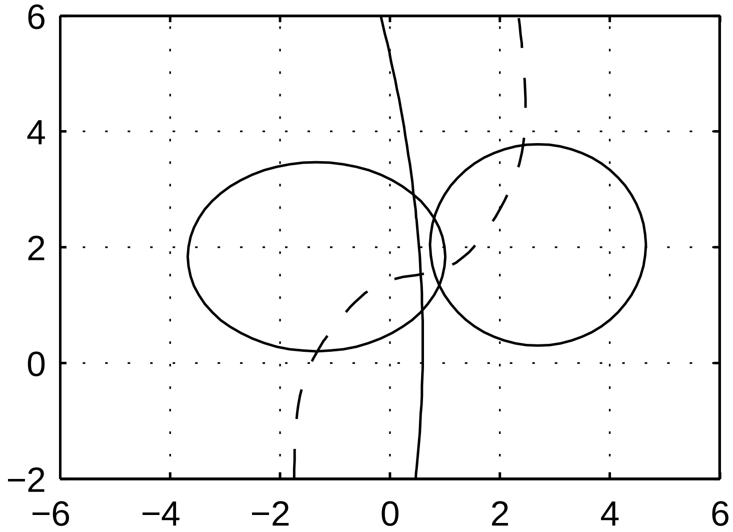
$\boldsymbol{\alpha}^{\texttt{svm}} = \{\alpha_1^{\texttt{svm}}, \ldots, \alpha_n^{\texttt{svm}}\}$ set of SVM Lagrange multipliers
$\mathbf{G}$ associated with distance metric for SVM kernel

- 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).

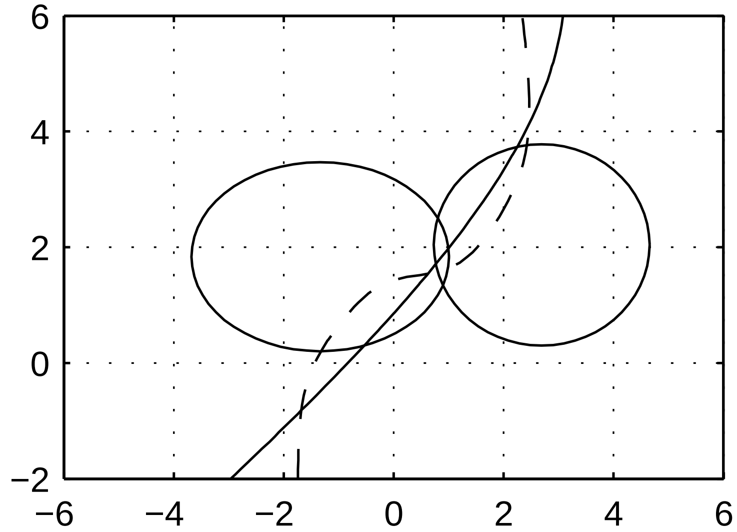
Maximum Margin α Example

- Artificial example training class-conditional Gaussian with score-space:

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



Maximum Likelihood



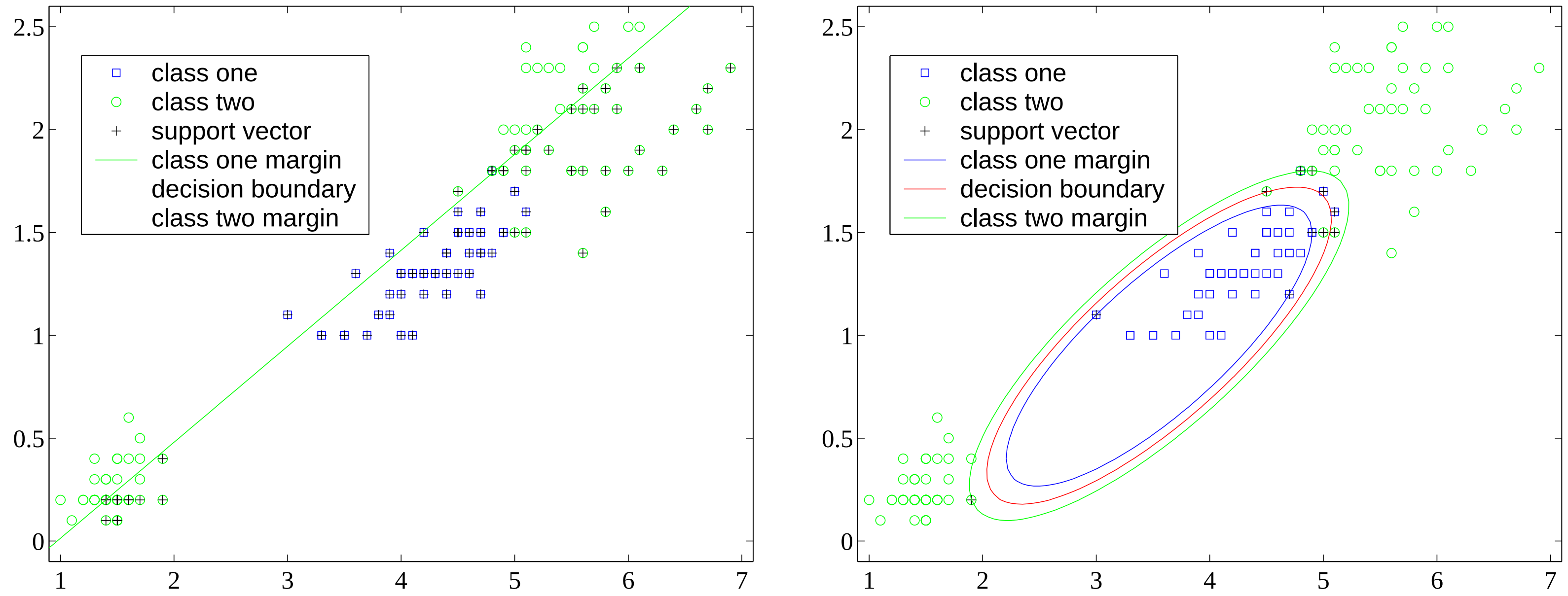
LLR+$\boldsymbol{\nabla}_{\mu,\Sigma}$

- Decision boundary closer to Bayes' decision boundary (dotted line)

Relationship to “Dynamic Kernels”

- Estimating augmented model parameters using an SVM is similar to using dynamic kernels
- Dynamic kernels map sequence data into a fixed dimensionality
  - standard SVM training can then be used
- Some standard kernels are related to augmented models:
  - generative kernels
  - Fisher kernel
  - marginalised count kernel

The "Kernel Trick"



- SVM decision boundary linear in the feature-space
  - may be made non-linear using a non-linear mapping $\phi()$ e.g.

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \quad K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

- Efficiently implemented using a Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j)^2$

Handling Sequence Data

- Sequence data (e.g. speech) has inherent variability in the number of samples:

The	cat	sat	on	the	mat	1200 frames

$$\mathbf{O}_1 = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{1200}\}$$

The	cat	sat	on	the	mat	900 frames

$$\mathbf{O}_2 = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{900}\}$$

- Kernels can be used to map from variable to fixed length data.
- SVMs can handle large dimensional data robustly:
  - higher dimensions - data more separable;
  - how to obtain high dimensional space?

Generative Kernels

- Generative models, e.g. HMMs and GMMs, handle variable length data
  - view as "mapping" sequence to a single dimension (log-likelihood)

$$\phi\left(\mathbf{O};\boldsymbol{\lambda}\right)=\frac{1}{T}\log\left(p(\mathbf{O};\boldsymbol{\lambda})\right)$$

- Extend feature-space:
  - add derivatives with respect to the model parameters
  - example is a log-likelihood ratio plus first derivative score-space:

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

  - Unrestricted form of Maximum Margin Augmented Model training

Fisher Kernel

- Fisher Kernels have the form

$$\phi\left(\mathbf{O};\boldsymbol{\lambda}\right)=\frac{1}{T}\left[\boldsymbol{\nabla}_{\lambda}\log\left(p(\mathbf{O};\boldsymbol{\lambda})\right)\right]$$

  Fisher kernel useful with large amounts of unsupervised data:

  – extracts general structure of data

- Generative kernels may be viewed as a supervised version of Fisher Kernels

  – are equivalent when 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.

Marginalised Count Kernel

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

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

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

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

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

Conditional Augmented Models

- Augmented models can be trained in a discriminative fashion, i.e. maximise

$$P(\omega_i|\mathbf{O};\boldsymbol{\lambda},\boldsymbol{\alpha}) = \frac{1}{Z(\boldsymbol{\lambda},\boldsymbol{\alpha})}\exp\left(\left[\begin{array}{c} 1 \\ \boldsymbol{\alpha}^{(i)} \end{array}\right]'\left[\begin{array}{c} \log(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(i)})) \\ \boldsymbol{\nabla}_{\lambda}\log(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(i)})) \end{array}\right]\right)$$

where for a K-class problem

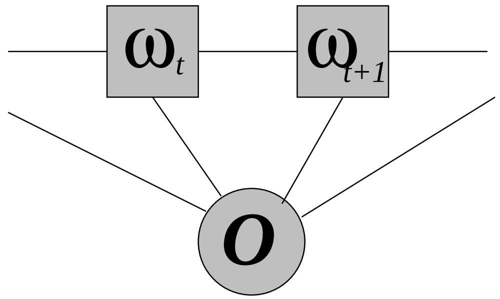
$$Z(\boldsymbol{\lambda},\boldsymbol{\alpha}) = \sum_{j=1}^{K}\exp\left(\left[\begin{array}{c} 1 \\ \boldsymbol{\alpha}^{(j)} \end{array}\right]'\left[\begin{array}{c} \log(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(j)})) \\ \boldsymbol{\nabla}_{\lambda}\log(\check{p}(\mathbf{O};\boldsymbol{\lambda}^{(j)})) \end{array}\right]\right)$$

Simple expression for normalisation term

- Standard gradient descent approaches may be used to train parameters
  - optimising $\boldsymbol{\alpha}$ is a convex optimisation problem - unique, global solution!

Conditional Random Fields

- Conditional Random Fields (CRFs) have become popular for classification
- undirected graph (see opposite)
- features extracted from graph
  - transition features - $T_k(\omega_{t-1}, \omega_t, \mathbf{O})$
  - state features - $T_k(\omega_t, \mathbf{O})$

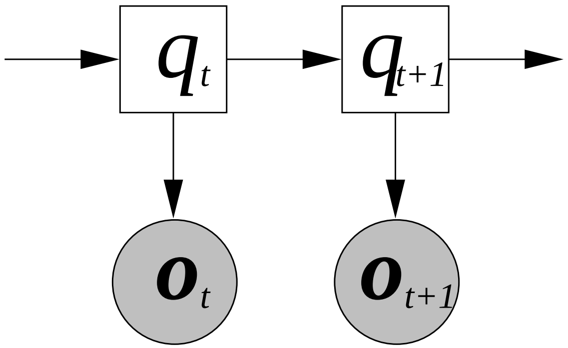


$$P(\omega_1, \ldots, \omega_T|\mathbf{O}) = \frac{1}{Z(\boldsymbol{\lambda})} \exp\left(\sum_t \boldsymbol{\lambda}_t' \mathbf{T}(\omega_{t-1}, \omega_t, \mathbf{O})\right)$$

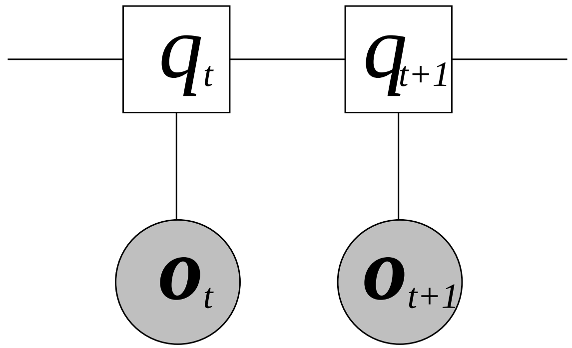
- Convex optimisation problem to find $\boldsymbol{\lambda}$
- Directly applicable to some sequence classes (POS tagging)
  - additional independence assumptions useful for speech

Hidden CRFs

- Hidden CRFs have been examined for speech recognition



(c) HMM DBN



(d) HCRF DBN

$$P(\omega_i|\mathbf{O};\boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{\lambda})}\sum_{\mathbf{q}\in\boldsymbol{\Theta}}\exp\left(\boldsymbol{\lambda}'\mathbf{T}(\omega_i,\mathbf{q},\mathbf{O})\right)$$

- No-longer convex optimisation problem
- Both CRFs and HCRFs assume knowledge of dependencies
  - A-HMM - extracts additional CRF statistics $\mathbf{T}(\omega_i,\mathbf{O};\boldsymbol{\lambda})$

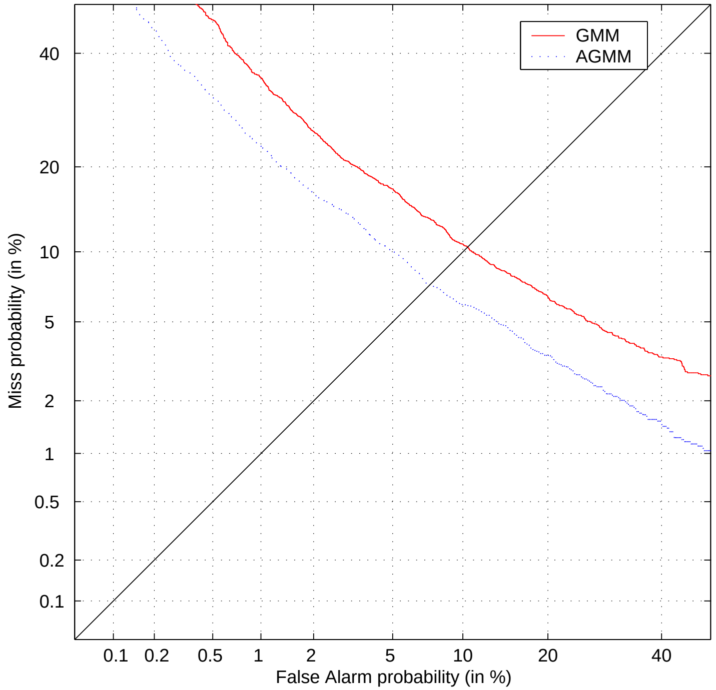
Speech Processing Experiments

- Augmented models examined on a range of speech processing tasks:
  - Speaker verification: binary classification task
  - Isolated letter classification: small number of classes (1-v-1 + voting)
  - LVCSR: mapping LVCSR task to binary task - acoustic codebreaking
- Conditional augmented models examined on:
  - TIMIT phone classification: multi-class classification

Speaker Verification

- GMM-MAP based speaker verification
  - enrolment MAP-adapted GMM used as the base distribution
  - first-order mean-derivative A-GMMs
  - evaluated on NIST 2002 SRE Task

# Comp.	EER (%)	
	GMM	A-GMM
128	12.17	8.62
256	11.24	7.88
512	11.13	7.48
1024	10.43†	7.31



- A-GMM consistently out-performs standard GMM

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 derivatives for A-HMM, 1-v-1 training, voting

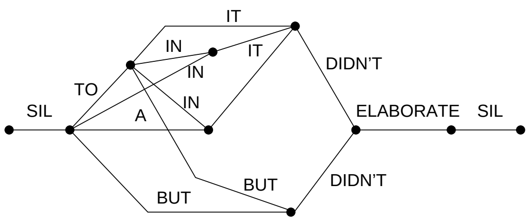
Classifier	Training		WER
	Base (λ)	Aug (α)	(%)
HMM	ML	—	8.7
	MMI	—	4.8
A-HMM	ML	MM	5.0
	MMI	MM	4.3

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

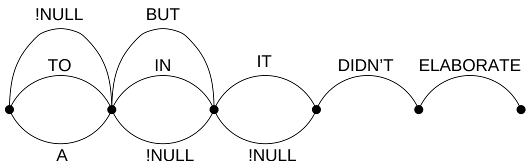
Binary Classifiers and LVCSR

- 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?

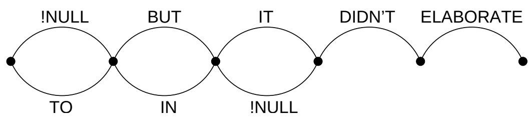
- Use pruned confusion networks (Venkataramani et al ASRU 2003):



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 - 640 selected) 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)	Classifier	Training		WER (%)	
		Base (λ)	Aug (α)	Trn	Tst
CAN/CAN'T (3761)	HMM	ML	—	10.4	11.0
		MMI	—	9.0	10.4
	A-HMM	ML	MM	7.1	9.2
	C-Aug	ML	CML	7.2	9.6

- A-HMM outperforms both ML and MMI HMM
  - also outperforms using "equivalent" number of parameters
- A-HMM outperforms C-Aug HMM
  - maximum margin found to (unsurprisingly) be more robust
- 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)

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

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

$$\boldsymbol{\phi}^{\text{cn}}(\mathbf{O};\boldsymbol{\lambda}) = \begin{bmatrix} \mathcal{F}(\omega_1) - \mathcal{F}(\omega_2) \\ \boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda}) \end{bmatrix}$$

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

Evaluation Data LVCSR Results

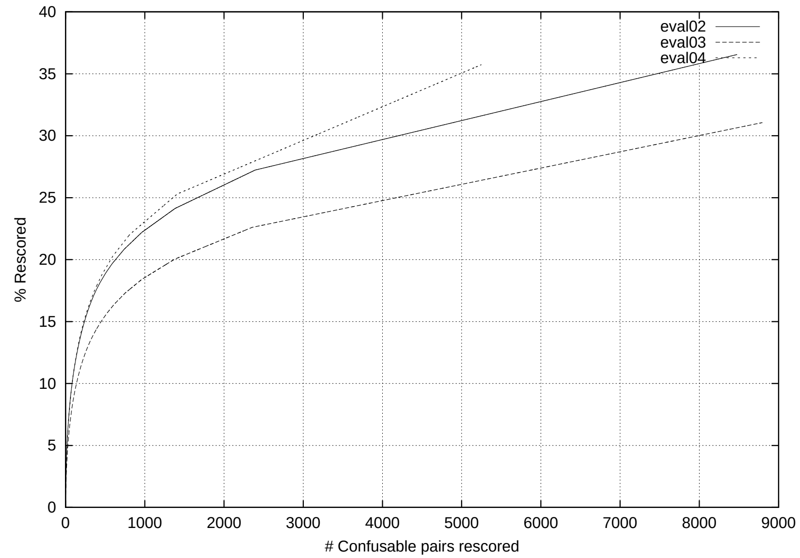
- Baseline performance using Viterbi and Confusion Network decoding

Decoding	trigram LM
Viterbi	30.8
Confusion Network	30.1

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

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

- performance on `eval03` CTS task
- only 1.6% of 76157 words rescored
- more SVMs required!



TIMIT Classification Experiments

- TIMIT phone-classification experiments
  - 48 base-phones modelled (mapped to 39 for scoring)
  - context-independent phone base models. 3-emitting state HMMs

Classifier	Training		Components	
	Base(λ)	Aug(α)	10	20
HMM	ML	–	29.4	27.3
C-Aug	ML	CML	24.2	–
HMM	MMI	–	25.3	24.8
C-Aug	MMI	CML	23.4	–

Classification error on the TIMIT core test set

- C-Aug outperforms HMMs for comparable numbers of parameters

Summary

- Dependency modelling for sequence data
  - 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
- ML-augmented model training complex
  - binary cases using linear classifier
  - C-Aug models an interesting alternative
- Evaluated on a speech processing tasks
  - interesting to see how it works on other tasks ...