SVMs, Generative Kernels & Maximum Margin Statistical Models

Mark Gales & Martin Layton

16 December 2004

Cambridge University Engineering Department

Institute of Statistical Mathematics

Overview

- Dependency Modelling in Speech Recognition:
  - latent variables
  - exponential family
- Augmented Statistical Models
  - Gaussian mixture models and hidden Markov models
- Support Vector Machines
  - Generative Kernels
  - maximum margin training
- 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

Bayesian networks

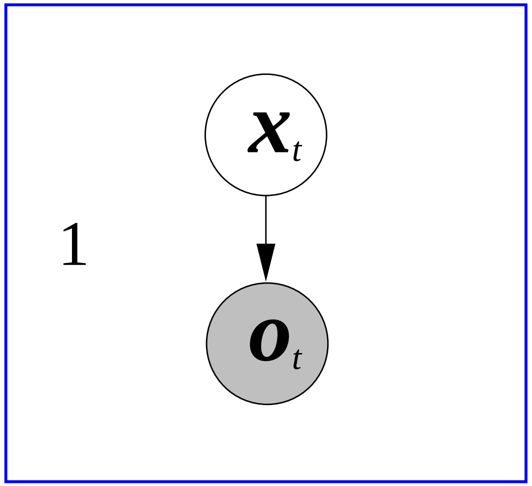
Yield conditional-independence assumptions

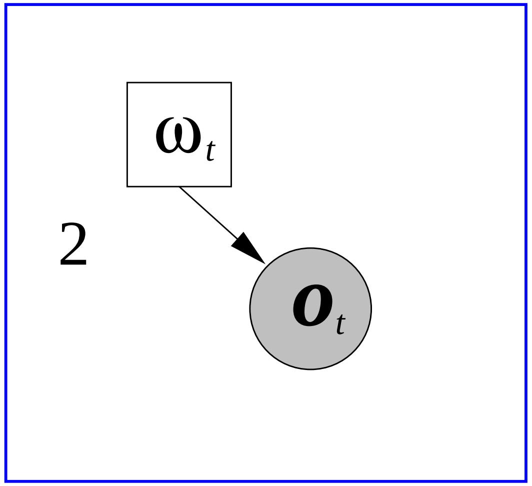
- round node: continuous variable;
- square node: discrete variable;
- shaded node: observable;
- no arrow: conditional independence.

Examples:

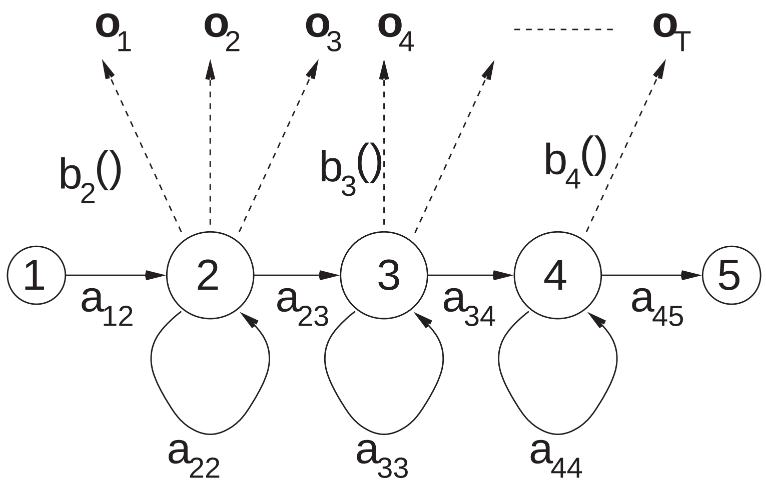
1. Factor Analysis:
$$p(\boldsymbol{o}_t|\boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{o}_t; \boldsymbol{C}_t\boldsymbol{x}_t + \boldsymbol{\mu}_t^{(o)}, \boldsymbol{\Sigma}_t^{(o)}\right)$$

2. Gaussian Mixture Model:
$$p(\boldsymbol{o}_t|\omega_t = n) = \mathcal{N}\left(\boldsymbol{o}_t; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\right)$$

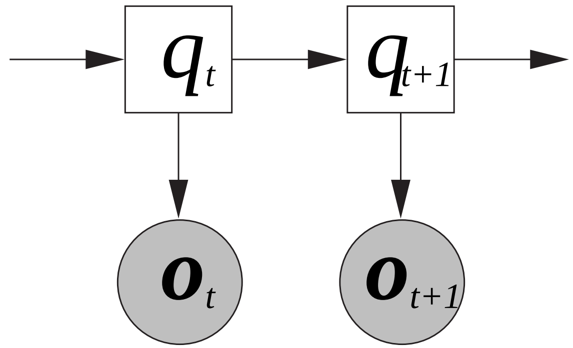




Hidden Markov Model - A Dynamic Bayesian Network



(a) Standard HMM phone topology



(b) HMM Dynamic Bayesian Network

- Notation for DBNs:

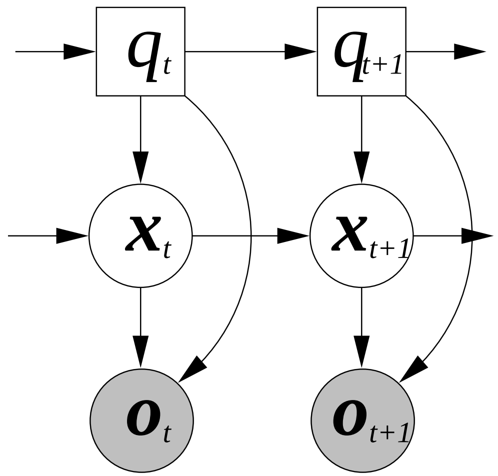
  circles - continuous variables   shaded - observed variables
  squares - discrete variables   non-shaded - unobserved variables

- 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

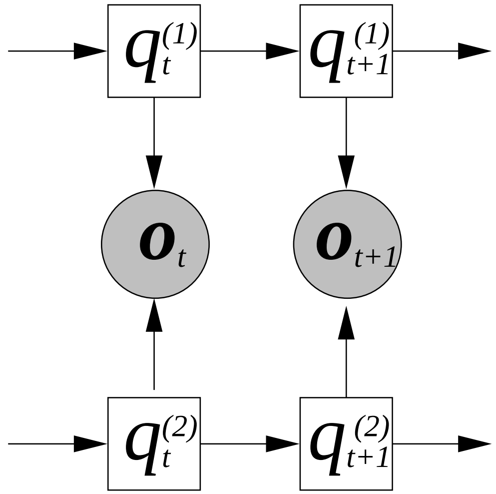
Switching linear dynamical system:

- discrete and continuous state-spaces
- observations conditionally independent given continuous and discrete state;
- exponential growth of paths, $O(N_s^T)$
  $\Rightarrow$ approximate inference required.



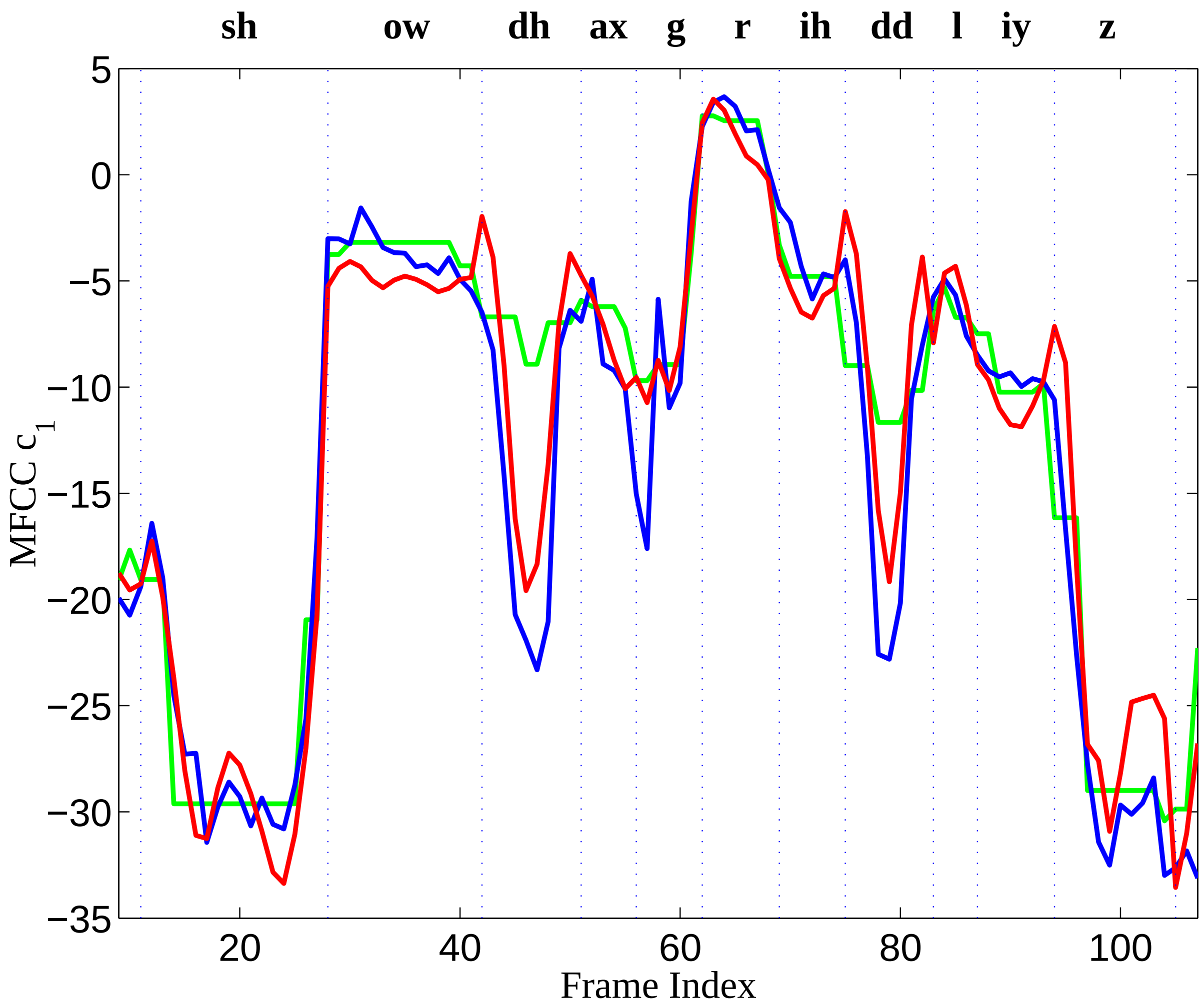
Multiple data stream DBN:

- 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 ...



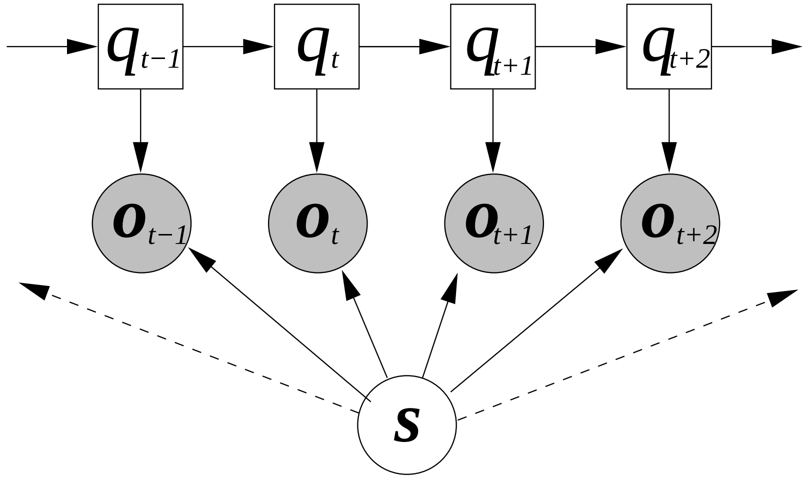
Legend

- True
- HMM
- SLDS

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

Adaptive Training

- Observations conditionally independent:
  - state that generated the observation
  - continuous latent variable(s) s



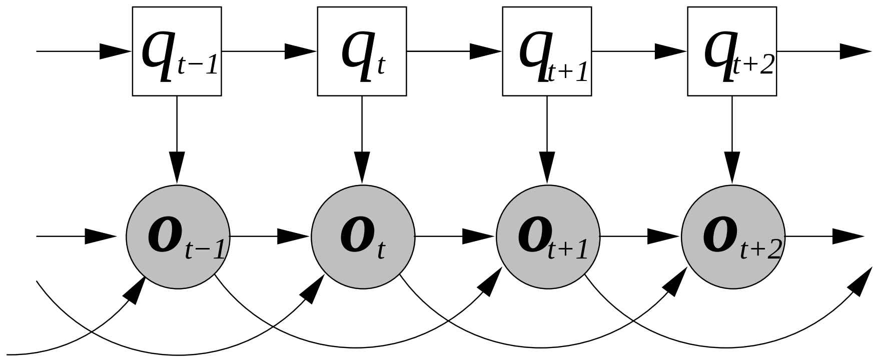
- Latent variable:
  - represents the speaker/environment
  - various forms CMN/CVN/VTLN

- One powerful form is Speaker Adaptive Training using constrained MLLR

$$p(\mathbf{O};\boldsymbol{\lambda}) = \sum_{\theta\in\Theta}\int_{\mathcal{R}^n}\left(\prod_{t=1}^{T} P(\theta_t|\theta_{t-1})|\mathbf{A}|p(\mathbf{A}\boldsymbol{o}_t+\mathbf{b}|\theta_t;\boldsymbol{\lambda})\right) p(\mathbf{A},\mathbf{b}|\boldsymbol{\lambda})d\mathbf{A}d\mathbf{b}$$

  - ML/MAP estimation commonly used for $\mathbf{A},\mathbf{b}$
  - exact Bayesian inference intractable (at the moment)
  - used in many state-of-the-art speech recognition systems

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

- Hard to determine the appropriate form of statistics $(\mathbf{T}(\mathbf{O}))$ to use ...

Sufficient Statistic Example

- For the one-dimensional observation sequences $\mathbf{O} = o_1, \ldots, o_T$ extract:
  - $T_1(\mathbf{O}) = \sum_{t=2}^{T} o_t; \quad T_2(\mathbf{O}) = \sum_{t=2}^{T} o_{t-1}$
  - $T_3(\mathbf{O}) = \sum_{t=2}^{T} o_t o_{t-1}; \quad T_4(\mathbf{O}) = \sum_{t=2}^{T} o_t^2; \quad T_5(\mathbf{O}) = \sum_{t=2}^{T} o_{t-1}^2$
- Probability (given the first observation) by

$$p(o_2, \ldots, o_T | o_1; \boldsymbol{\alpha}) = \exp\left(\sum_{i=1}^{5} \alpha_i T_i(\mathbf{O})\right) / \tau$$

  - $\boldsymbol{\alpha}$ and τ directly found from the joint distribution of $\{o_t, o_{t-1}\}$

$$\boldsymbol{\mu} = \frac{1}{T-1}\left[\begin{array}{c} T_1(\mathbf{O}) \\ T_2(\mathbf{O}) \end{array}\right]; \quad \boldsymbol{\Sigma} = \frac{1}{T-1}\left[\begin{array}{cc} T_4(\mathbf{O}) & T_3(\mathbf{O}) \\ T_3(\mathbf{O}) & T_5(\mathbf{O}) \end{array}\right] - \boldsymbol{\mu}\boldsymbol{\mu}'$$

  - has the form of a single component single-state buried Markov model

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

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

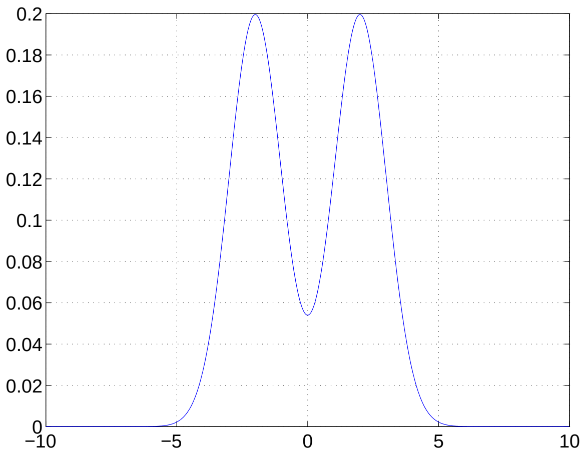
  - can be very complex to estimate

Augmented Gaussian Mixture Model

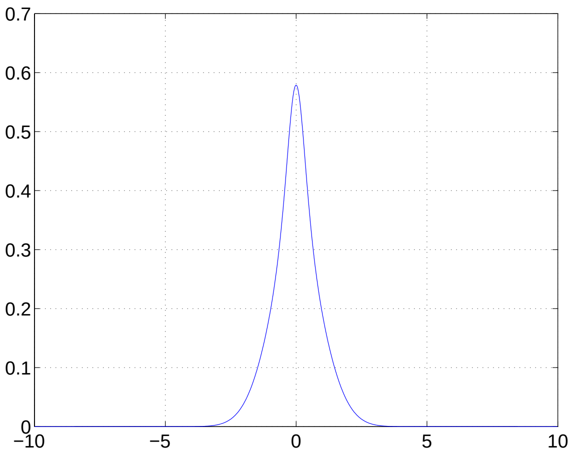
- Use a GMM as the base distribution: $\check{p}(\boldsymbol{o};\boldsymbol{\lambda}) = \sum_{m=1}^{M} c_m \mathcal{N}(\boldsymbol{o};\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)$
  - considering only the first derivatives of the means

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

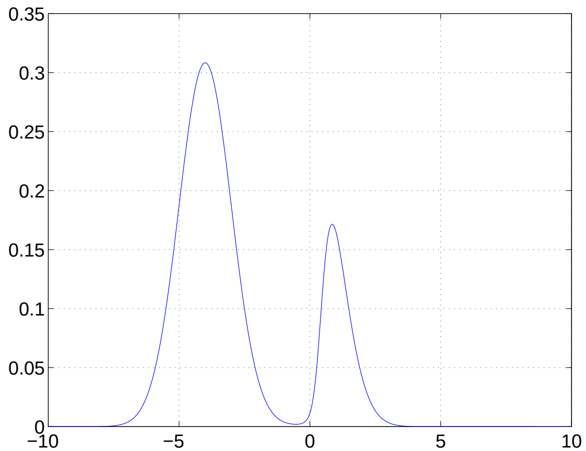
- Simple two component one-dimensional example:



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



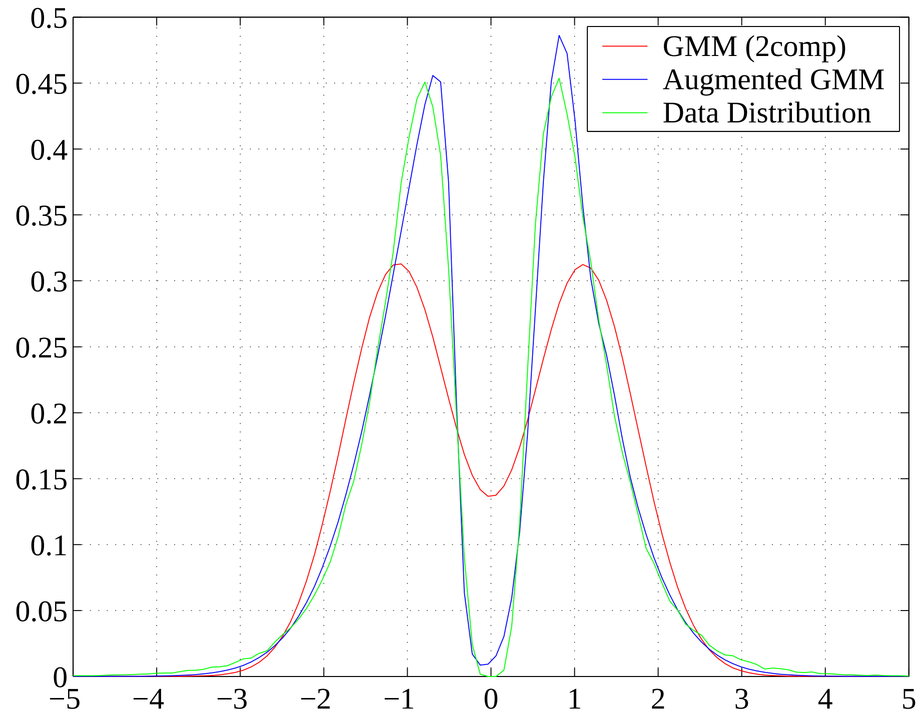
$\boldsymbol{\alpha} = [-1.0, -1.0]'$



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

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 Hidden Markov Model

- For an HMM: $\check{p}(\mathbf{O};\boldsymbol{\lambda}) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left\{ \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} \left(\sum_{m\in\theta_t} c_m \mathcal{N}(\mathbf{o}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \right) \right\}$
  - The form of the statistics when an HMM used as the base distribution

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

$\gamma_{jm}(t) = P(\theta_t = \{s_j, m\}|\mathbf{O};\boldsymbol{\lambda})$, θ_t is the state/component pairing at time t

  - An example higher order derivative has the form

$$\boldsymbol{\nabla}_{\mu_{in}} \boldsymbol{\nabla}'_{\mu_{jm}} \log\left(\check{p}(\mathbf{O};\boldsymbol{\lambda})\right) =$$

$$\sum_{t=1}^{T}\sum_{\tau=1}^{T} \left\{ \left(\gamma_{\{jm,in\}}(t,\tau) - \gamma_{jm}(t)\gamma_{in}(\tau)\right) \boldsymbol{\Sigma}_{in}^{-1} (\mathbf{o}_\tau - \boldsymbol{\mu}_{in}) (\mathbf{o}_t - \boldsymbol{\mu}_{jm})' \boldsymbol{\Sigma}_{jm}^{-1} \right\}$$

where $\gamma_{\{jm,in\}}(t,\tau)$ is the joint state/component posterior.

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

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

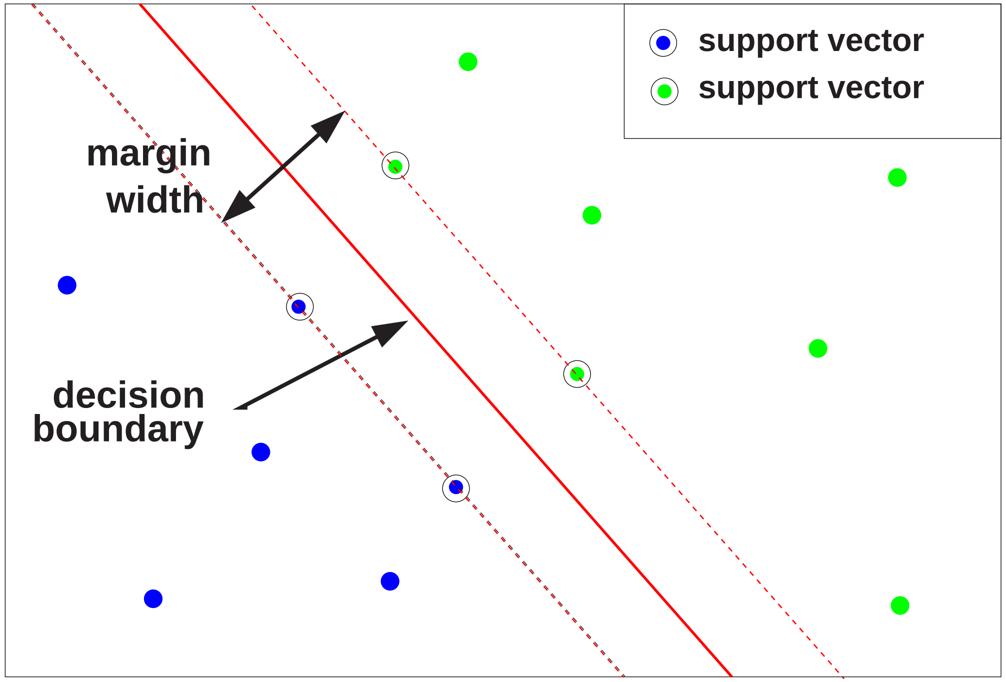
- Augmented models of this form
  - keep independence assumptions of the base distribution
  - remove conditional independence assumptions of the base model
    - the local exponential model depend on a posterior ...

- Same applies for dynamic models such as HMMs

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

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 $(\mathbf{x}_i.\mathbf{x}_j)$.
- Successfully applied to many tasks - how to apply to speech?

Support Vector Machine Training

- For non-linearly separable data a soft margin classifier is used: minimise

$$\tau(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i\left(\langle\mathbf{w},\mathbf{x}_i\rangle + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0$

  – two terms: k/margin^2 and error rate bound (C balances importance)

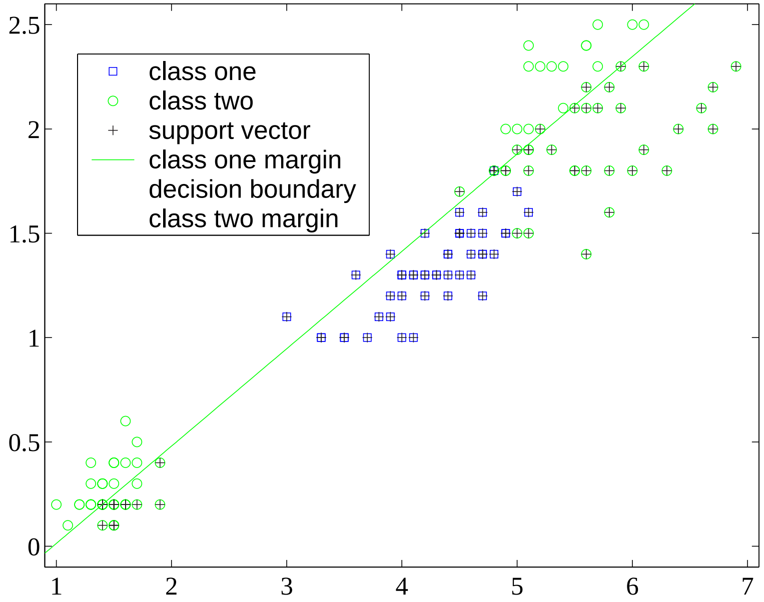
- The dual is commonly optimised (based only on $\boldsymbol{\alpha}^{\mathtt{svm}}$)

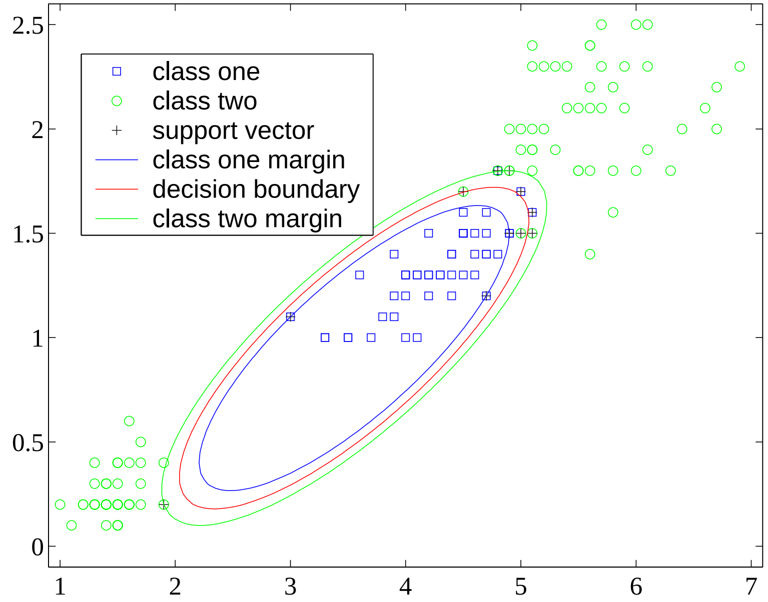
$$\hat{\boldsymbol{\alpha}}^{\mathtt{svm}} = \max_{\alpha^{\mathtt{svm}}}\left\{\sum_{i=1}^{n}\alpha_i^{\mathtt{svm}} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i^{\mathtt{svm}}\alpha_j^{\mathtt{svm}}y_iy_j\left(\mathbf{x}_i.\mathbf{x}_j\right)\right\}$$

subject to $0 \leq \alpha_i^{\mathtt{svm}} \leq C, \quad \sum_{i=1}^{m}\alpha_i^{\mathtt{svm}}y_i = 0$, $y_i \in \{-1,1\}$ indicates the class.

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i^{\mathtt{svm}}y_i\mathbf{x}_i$$

The "Kernel Trick"





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

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

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

Handling Speech data

- Speech data 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.
- Generative models are an obvious candidate:
  - HMMs and GMMs handle variable length data
  - view as "mapping" sequence to a single dimension (log-likelihood)

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

Generative Kernels

- SVMs can handle large dimensional data robustly:
  - higher dimensions - data more separable;
  - how to increase dimensionality?
- Have a generative model for each class: parameters ω_1: $\boldsymbol{\lambda}^{(1)}$ and ω_2: $\boldsymbol{\lambda}^{(2)}$
- Use a score-space:
  - add derivatives with respect to the model parameters
  - example is a log-likelihood ratio plus first derivative score-space:

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

  - dimensionality of feature-space: $1+$ `parameters` $\boldsymbol{\lambda}^{(1)}$ $+$ `parameters` $\boldsymbol{\lambda}^{(2)}$

Score-Space Metrics

- SVM training involves a distance from the decision boundary
  - need to determine appropriate distance metric
- Choose a maximally non-committal metric

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

where $\mathbf{O}_i$ and $\mathbf{O}_j$ are sequences of length T_i and T_j respectively, and

$$\mathbf{G} = \mathcal{E}\left\{(\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda}) - \boldsymbol{\mu}_\phi)\,(\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda}) - \boldsymbol{\mu}_\phi)'\right\}$$

where $\boldsymbol{\mu}_\phi = \mathcal{E}\left\{\boldsymbol{\phi}(\mathbf{O}; \boldsymbol{\lambda})\right\}$.

- In practice $\mathbf{G}$ is usually set to be a diagonal matrix

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

$$\left[\begin{array}{c}\mathbf{w}\\ w_0\end{array}\right]'\left[\begin{array}{c}\phi(\mathbf{O};\boldsymbol{\lambda})\\ 1\end{array}\right] \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 0$$

  – form of score-space and linear decision boundary
  – SVM good choice as possibly high dimensional score-space

Augmented Model Training - Binary Case (cont)

- Score-space is given by (first order derivatives)

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

  - this is the generative kernel $\phi^{11}(\mathbf{O};\boldsymbol{\lambda})$
  - 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 w_0

  - depends on both $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$

Estimating Model Parameters

- Two sets of parameters to be estimated using training data $\{\mathbf{O}_1, \ldots, \mathbf{O}_n\}$:
  - generative models (Kernel) $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}\}$
  - SVM (Lagrange multipliers) $\boldsymbol{\alpha}^{\texttt{svm}} = \{\alpha_1^{\texttt{svm}}, \ldots, \alpha_n^{\texttt{svm}}\}$
  - direction of decision boundary ($y_i \in \{-1, 1\}$ label of training example)

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

- SVM parameters trained using maximum margin training (to find $\boldsymbol{\alpha}^{\texttt{svm}}$)
- Kernel parameters may be estimated using:
  - maximum likelihood (ML) training;
  - discriminative training (e.g. maximum mutual information)
  - maximum margin (MM) training.

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

$$\frac{1}{w_1}\left(\left[\begin{array}{c}\mathbf{w}\\ w_0\end{array}\right]'\left[\begin{array}{c}\boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda})\\ 1\end{array}\right]\right) = \frac{1}{T}\log\left(\frac{P(\omega_1|\mathbf{O})}{P(\omega_2|\mathbf{O})}\right)$$

  - 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

$$\left[\begin{array}{c}1\\ \boldsymbol{\alpha}^{(1)}\\ \boldsymbol{\alpha}^{(2)}\end{array}\right] = \frac{1}{w_1}\mathbf{w} = \frac{1}{w_1}\sum_{i=1}^{n}\alpha_i^{\mathtt{svm}} y_i \mathbf{G}^{-1}\boldsymbol{\phi}(\mathbf{o}_i;\boldsymbol{\lambda})$$

Maximum Margin Kernel Estimation

- Using maximum margin training to estimate Kernel appealing:
  - optimising $\boldsymbol{\alpha}^{\texttt{svm}}$ yields local exponential parameters
  - optimising $\boldsymbol{\lambda}$ yields parameters of the base distribution
- Modified version of the standard SVM dual used:

$$\{\hat{\boldsymbol{\alpha}}^{\texttt{svm}}, \hat{\boldsymbol{\lambda}}\} = \arg\max_{\alpha^{\texttt{svm}}}\min_{\lambda}\left\{\sum_{i=1}^{n}\alpha_i^{\texttt{svm}} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i^{\texttt{svm}}\alpha_j^{\texttt{svm}}y_i y_j K(\mathbf{O}_i, \mathbf{O}_j; \boldsymbol{\lambda})\right\}$$

- Iterative optimisation required:
  - given values of $\boldsymbol{\lambda}$ perform standard SVM training
  - given values of $\boldsymbol{\alpha}^{\texttt{svm}}$ perform gradient descent optimisation of $\boldsymbol{\lambda}$

Maximum Margin Training (detail)

- Training procedure used:
  1. Initialise parameters, $\boldsymbol{\lambda}_0$, of generative model using MLE
  2. Train SVM to locate initial support vectors, $\boldsymbol{\alpha}_0^{\texttt{svm}}$
  3. Calculate initial value of objective function, $W^{(0)} = W(\boldsymbol{\lambda}_0, \boldsymbol{\alpha}_0^{\texttt{svm}})$
  4. For each iteration k:
     - (A) $\boldsymbol{\lambda}_k = \arg\min_\lambda W(\boldsymbol{\lambda}; \boldsymbol{\alpha}_{k-1}^{\texttt{svm}})$
     - (B) $\boldsymbol{\alpha}_k^{\texttt{svm}} = \arg\max_{\alpha^{\texttt{svm}}} W(\boldsymbol{\alpha}^{\texttt{svm}}; \boldsymbol{\lambda}_k)$
     - Recalculate objective function, $W^{(k)} = W(\boldsymbol{\lambda}_k, \boldsymbol{\alpha}_k^{\texttt{svm}})$

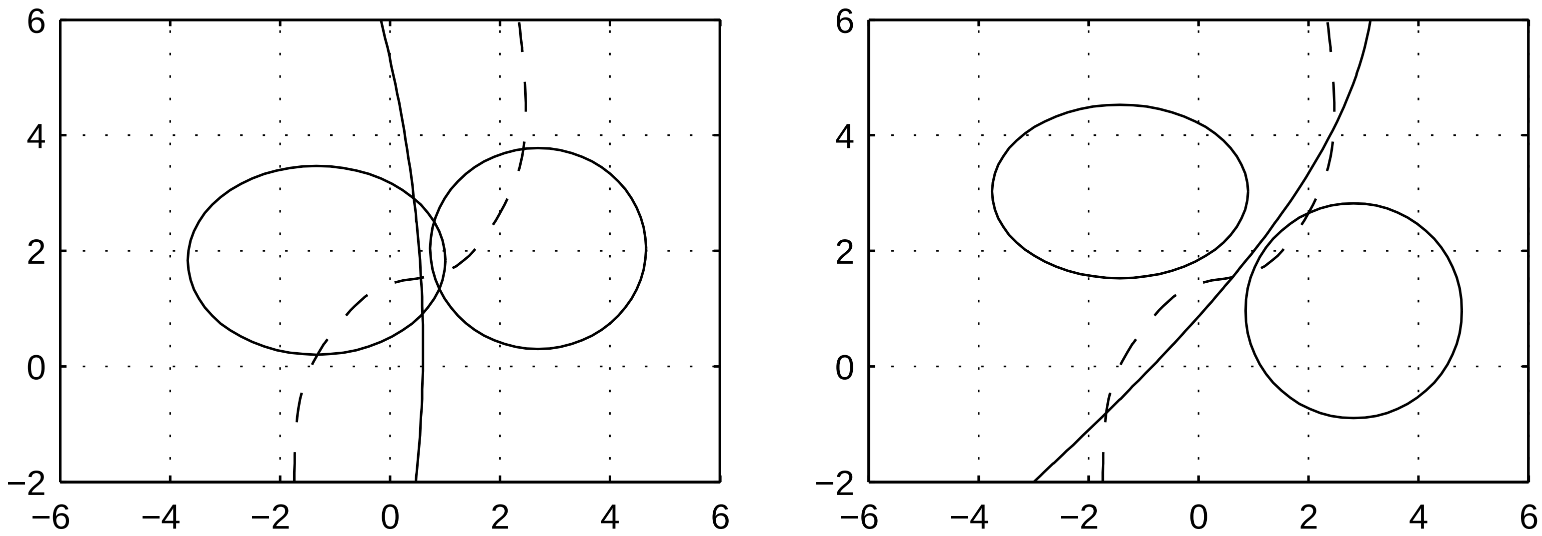
     Repeat until convergence: $|W^{(k)} - W^{(k-1)}| < \epsilon$

- (A) is a gradient descent scheme involving backing-off
  - back-off required to ensure that KKT conditions still satisfied
- (B) is standard SVM training

Maximum Margin Example

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

$$\phi(\boldsymbol{o};\boldsymbol{\lambda}) = \left[\log\left(\check{p}(\boldsymbol{o};\boldsymbol{\lambda}^{(1)})\right) - \log\left(\check{p}(\boldsymbol{o};\boldsymbol{\lambda}^{(2)})\right)\right]$$



Maximum Likelihood   Maximum Margin

- Decision boundary closer to Bayes' decision boundary (dotted line)
  - can also be obtained by optimising $\boldsymbol{\alpha}^{\mathtt{svm}}$ using $\boldsymbol{\phi}^{\mathtt{ll}}(\mathbf{O};\boldsymbol{\lambda})$ score-space ...

Exponential Family Base Distribution

- For a single component example the form of the augmented model is

$$p(\boldsymbol{o};\boldsymbol{\lambda},\boldsymbol{\alpha}) = \frac{1}{\tau}\exp\left(\boldsymbol{\lambda}'\mathbf{T}(\boldsymbol{o})\right)\exp\left(\boldsymbol{\alpha}'\mathbf{T}(\boldsymbol{o})\right) = \frac{1}{\tau}\exp\left((\boldsymbol{\alpha}+\boldsymbol{\lambda})'\mathbf{T}(\boldsymbol{o})\right)$$

  – still a member of the exponential family

- Using SVM training with generative kernel

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

  – will yield a maximum margin estimate of the exponential model
  – not true when using a model with latent variables

Valid Statistical Model?

- For a valid statistical model τ must be bounded:
  - for Gaussian covariance matrix must be positive-definite
- This places restrictions on the values of $\boldsymbol{\alpha}$
- Consider the simplest single-dimension, Gaussian base distribution
  - score-space is LLR and first derivatives of mean and variance
  - the augmented model is also Gaussian with effective variance

$$\sigma^2 = \frac{\check{\sigma}^4}{\check{\sigma}^2 - \alpha}$$

  if $\alpha \geq \check{\sigma}^2$ then the variance is negative!

- In practice this has not been an issue with the models examined here ...

Deterding Dataset

- Data from 11 vowels in British English in context of `h*d`
  - steady state portions partitioned into 6 Hamming window segments
  - linear prediction analysis to yield 10 log area parameters
  - static 10-dimensional feature vector for training/testing
- Corpus consists of
  - 48 training examples per vowel (total of 528 examples)
  - 42 test examples per vowel (total of 462 examples)
- Multi-class problem handled using set of `1-v-1` SVM classifiers
  - single pair ties resolved using pair classifier decision
  - multiple ties resolved using the GMM classifier

Deterding Data Experiments

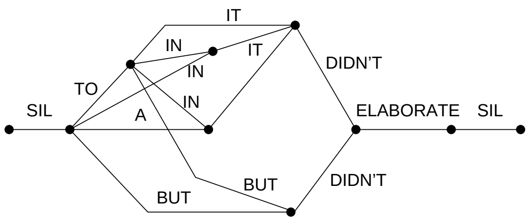
Classifier	Num. Comp.	Training (%) initial	Training (%) final	Test (%) initial	Test (%) final
GMM	1	40.0		55.8	
GMM	2	27.7		45.2	
SVM (LLR)	1	38.1	1.9	58.0	47.4
SVM (LLR)	2	26.3	0.8	48.5	38.8
SVM (LLR + $\boldsymbol{\nabla}_{\mu}$)	1	10.6	1.0	46.3	48.1

- Maximum margin training of kernel (base distribution)
  - initial - performance using ML values for $\boldsymbol{\lambda}$
  - final - performance using MM values for $\boldsymbol{\lambda}$
- Use of maximum margin training improved performance
  - but overtraining clear with maximum margin training

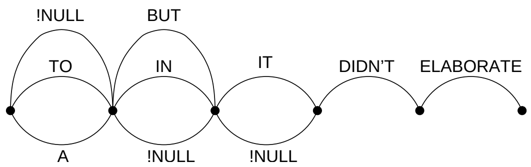
SVMs and LVCSR

- SVMs are inherently binary:
  - speech recognition has a vast number of possible classes;
  - how to map to a simple binary problem?

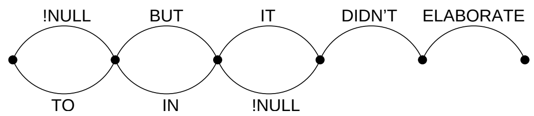
- Use pruned confusion networks:



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).

Incorporating Posterior Information

- Useful to incorporate arc log-posterior $(\mathcal{F}(\omega_1), \mathcal{F}(\omega_1))$ 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)

$$\frac{1}{T}\log\left(\frac{\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(1)},\boldsymbol{\alpha}^{(1)})}{\overline{p}(\mathbf{O};\boldsymbol{\lambda}^{(2)},\boldsymbol{\alpha}^{(2)})}\right) + b + \beta\left(\mathcal{F}(\omega_1) - \mathcal{F}(\omega_2)\right) \begin{array}{c}\omega_1\\ > \\ < \\ \omega_2\end{array} 0$$

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

$$\boldsymbol{\phi}^{\mathrm{cn}}(\mathbf{O};\boldsymbol{\lambda}) = \left[\begin{array}{c}\mathcal{F}(\omega_1) - \mathcal{F}(\omega_2)\\ \boldsymbol{\phi}(\mathbf{O};\boldsymbol{\lambda})\\ 1\end{array}\right]$$

- Incorporating in score-space requires consistency between train/test 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`:
  - bigram language model trained on `fsh2004sub`
- Perform 8-fold cross-validation on 400 hours training data:
  - matched training and test conditions
  - ML-trained Gaussian mixture model (first derivatives) score-space
  - posteriors "biased" as HMMs trained on "test" data
- Evaluation on held-out data (`eval03`)
  - 6 hours of test data
  - decoded using either LVCSR bigram or trigram
  - baseline using confusion network decoding

8-Fold Cross-Validation LVCSR Results

Word Pair (examples)	Training	CN post.	# Components 1	2	4
A/THE (8533)	ML	79.8	58.3	58.4	56.2
	SVM $\phi^{\tt ll}()$		61.1	63.0	64.7
	$+\beta$CN		79.8	80.0	80.3
	SVM $\phi^{\tt cn}()$		80.4	80.1	80.6
CAN/CAN'T (3761)	ML	78.5	81.7	86.0	88.2
	SVM $\phi^{\tt ll}()$		84.8	89.4	90.5
	$+\beta$CN		88.5	91.2	91.9
	SVM $\phi^{\tt cn}()$		89.0	91.4	91.6
KNOW/NO (4475)	ML	83.1	68.4	69.4	70.8
	SVM $\phi^{\tt ll}()$		72.1	73.6	76.6
	$+\beta$CN		84.3	84.5	85.2
	SVM $\phi^{\tt cn}()$		85.7	86.2	86.2

- Posterior score-space best approach, maximum margin distributions useful.

Evaluation Data LVCSR Results

- Baseline performance using Viterbi and Confusion Network decoding

Decoding	**Language Model**	
	bigram	**trigram**
Viterbi	34.4	30.8
Confusion Network	33.9	30.1

- Rescore common confusion pairs using 4-component and $\phi^{11}() + \beta$CN

SVM Rescoring	**#corrected/#pairs (% corrected)**	
	bigram LM	**trigram LM**
9 SVMs	44/1401 (3.1%)	41/1310 (3.1%)
15 SVMs	55/2116 (2.6%)	43/1954 (2.2%)

  – β roughly set - error rate relatively insensitive to exact value
  – less than 3% of 76157 hypothesised words rescored - more SVMs required!

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
- Support vector machines
  - use of generative kernels for dynamic data
  - maximum margin training of augmented statistical models
- Preliminary results of a large vocabulary speech recognition task
  - SVMs/Augmented models possibly useful for speech recognition