Pattern Classification

- Introduction
- Parametric classifiers
- Semi-parametric classifiers
- Dimensionality reduction
- Significance testing

Semi-Parametric Classifiers

- Mixture densities
- ML parameter estimation
- Mixture implementations
- Expectation maximization (EM)

Mixture Densities

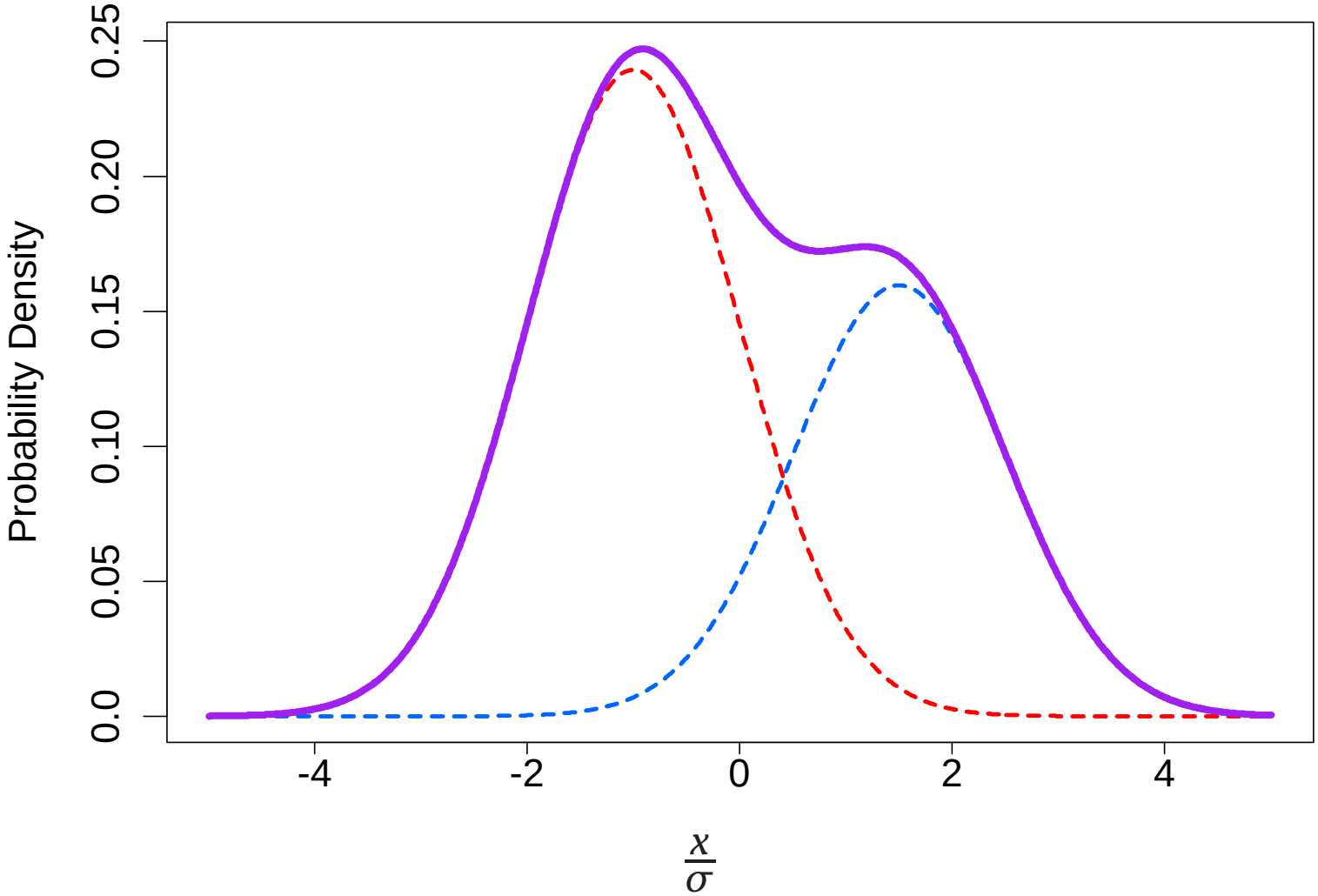
- PDF is composed of a mixture of m component densities $\{\omega_1, \ldots, \omega_m\}$:

$$p(\boldsymbol{x}) = \sum_{j=1}^{m} p(\boldsymbol{x}|\omega_j)P(\omega_j)$$

- Component PDF parameters and mixture weights $P(\omega_j)$ are typically unknown, making parameter estimation a form of unsupervised learning
- Gaussian mixtures assume Normal components:

$$p(\boldsymbol{x}|\omega_k) \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Mixture Example: One Dimension

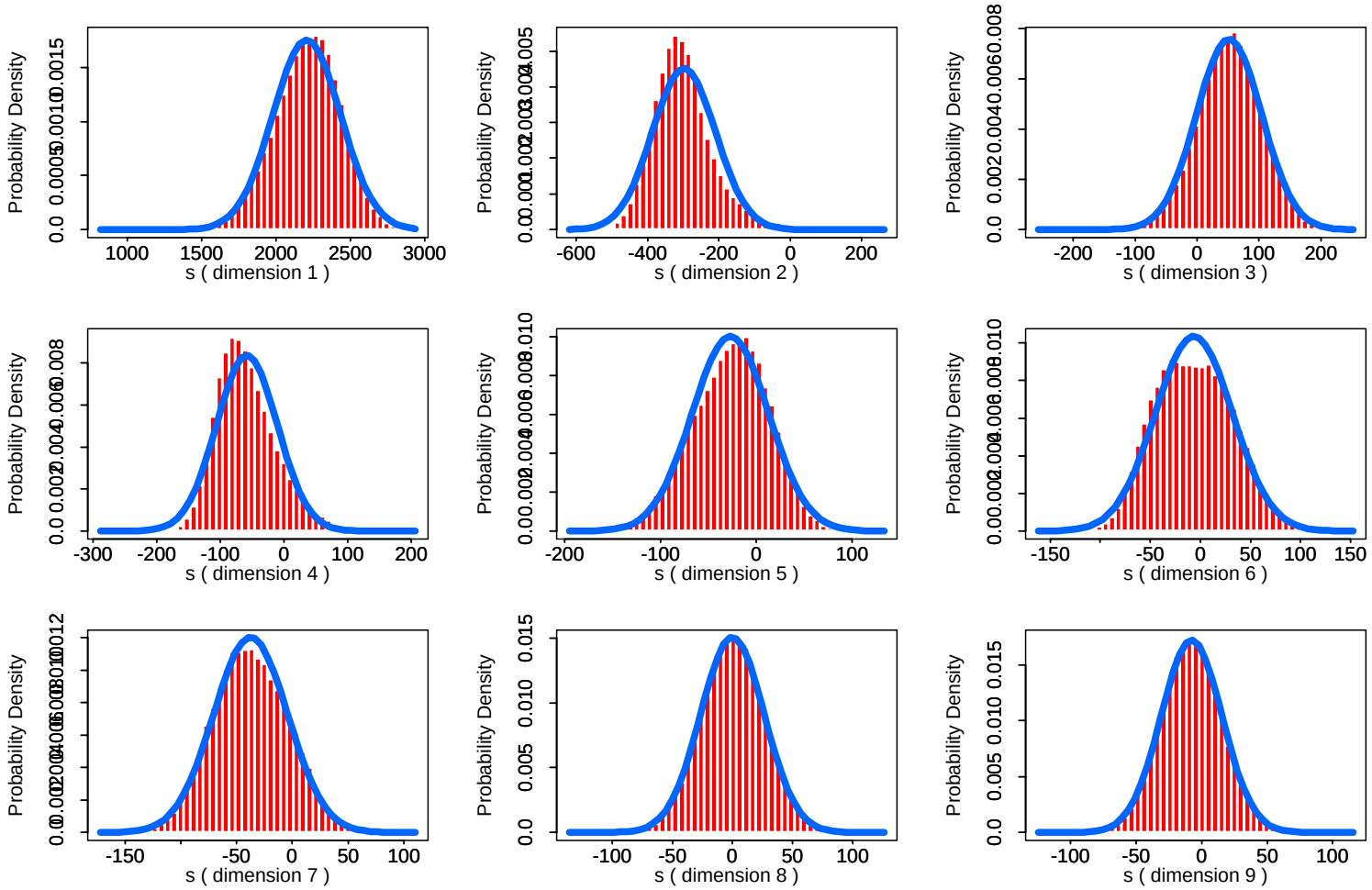


$$p(x) = 0.6p_1(x) + 0.4p_2(x)$$

$$p_1(x) \sim N(-\sigma, \sigma^2) \qquad p_2(x) \sim N(1.5\sigma, \sigma^2)$$

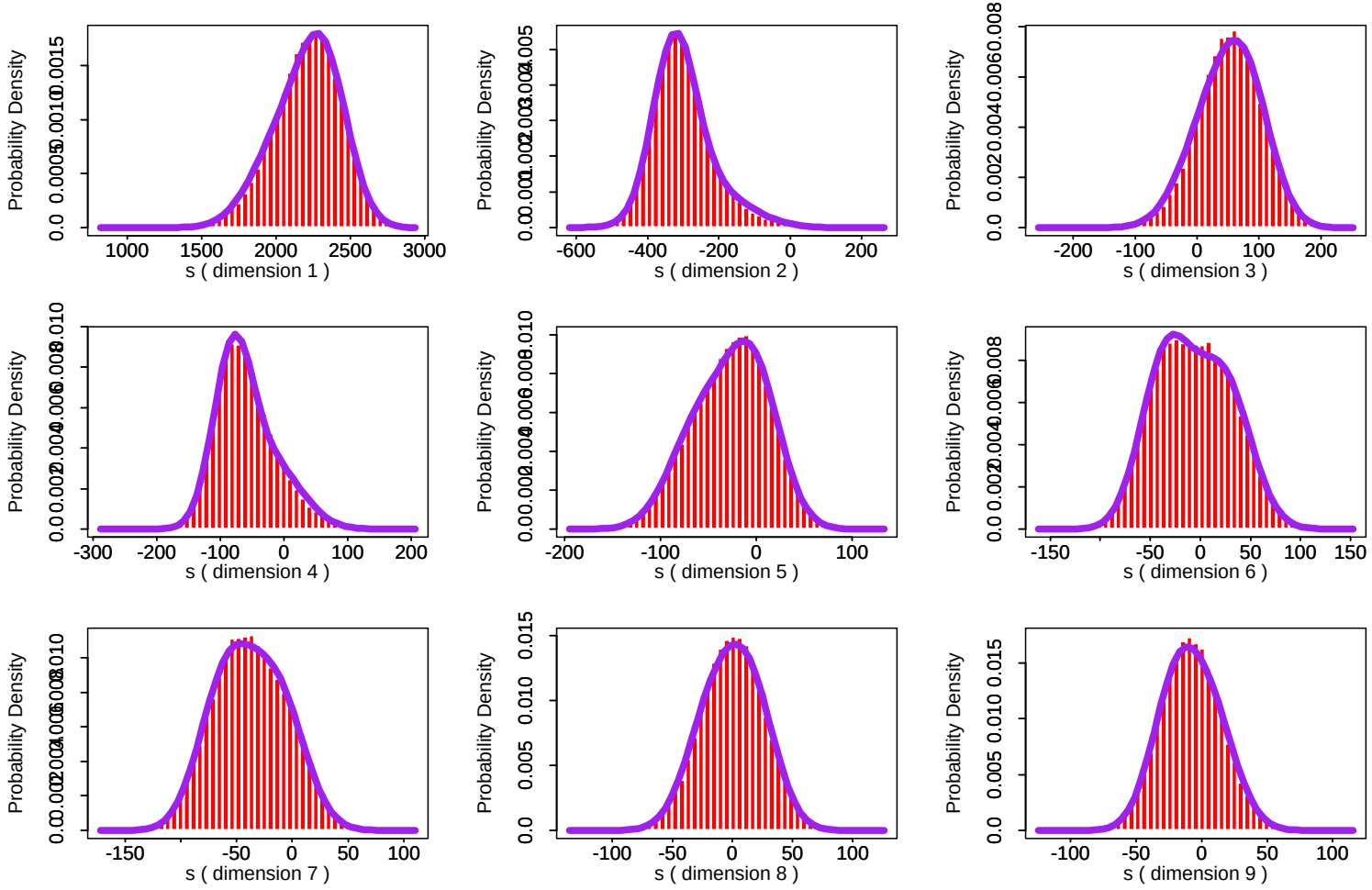
Gaussian Example

First 9 MFCC's from [s]: Gaussian PDF



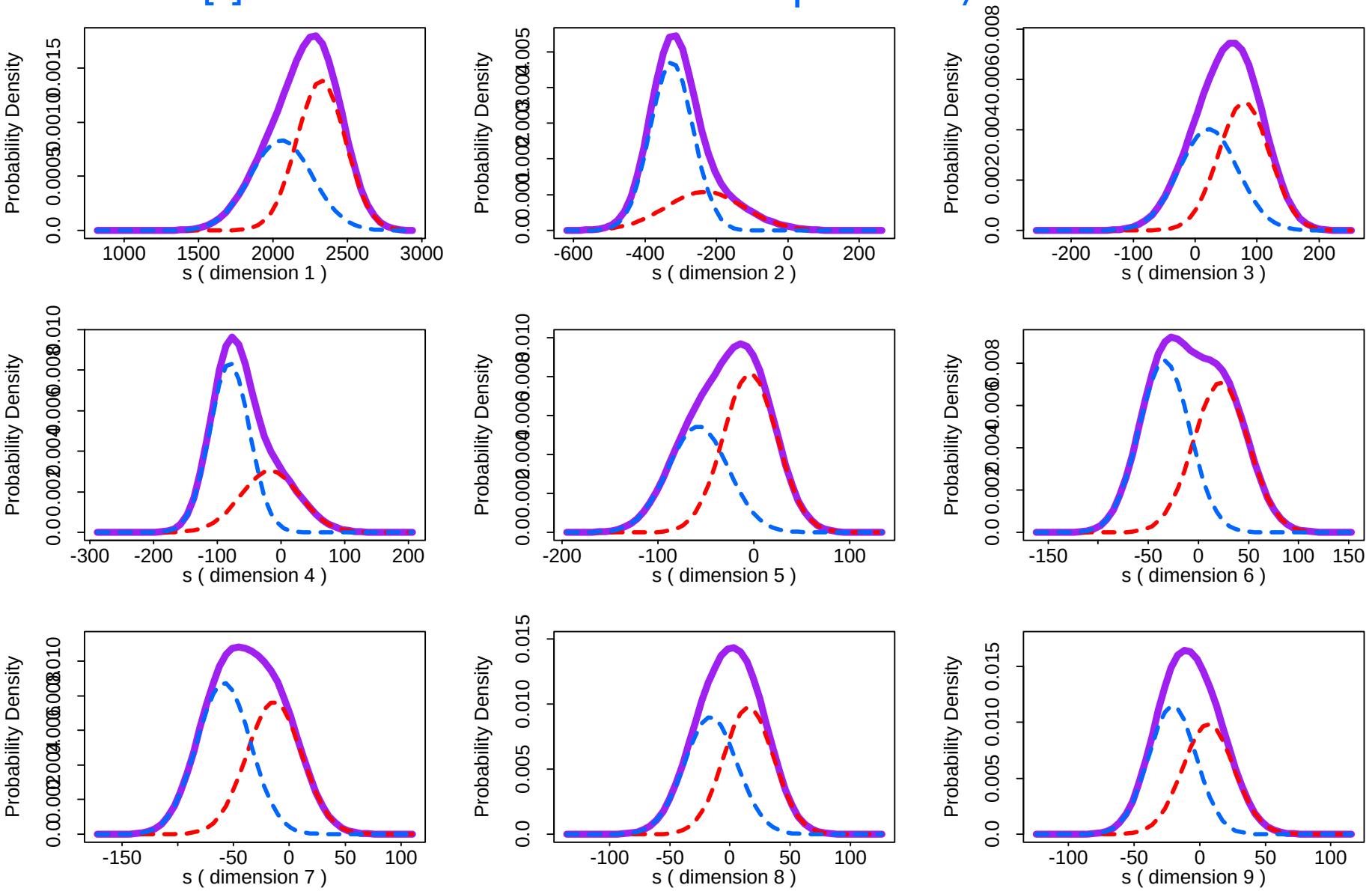
Independent Mixtures

[s]: 2 Gaussian Mixture Components/Dimension



Mixture Components

[s]: 2 Gaussian Mixture Components/Dimension



ML Parameter Estimation: 1D Gaussian Mixture Means

$$\log L(\mu_k) = \sum_{i=1}^{n} \log p(x_i) = \sum_{i=1}^{n} \log \sum_{j=1}^{m} p(x_i|\omega_j)P(\omega_j)$$

$$\frac{\partial \log L(\mu_k)}{\partial \mu_k} = \sum_i \frac{\partial}{\partial \mu_k} \log p(x_i) = \sum_i \frac{1}{p(x_i)} \frac{\partial}{\partial \mu_k} p(x_i|\omega_k)P(\omega_k)$$

$$\frac{\partial p(x_i|\omega_k)}{\partial \mu_k} = \frac{\partial}{\partial \mu_k} \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}} = p(x_i|\omega_k)\frac{(x_i-\mu_k)}{\sigma_k^2}$$

$$\frac{\partial \log L(\mu_k)}{\partial \mu_k} = \sum_i \frac{P(\omega_k)}{p(x_i)} p(x_i|\omega_k)\frac{(x_i-\mu_k)}{\sigma_k^2} = 0$$

since $\frac{p(x_i|\omega_k)P(\omega_k)}{p(x_i)} = P(\omega_k|x_i)$ $\qquad \hat{\mu}_k = \frac{\sum_i P(\omega_k|x_i)x_i}{\sum_i P(\omega_k|x_i)}$

Gaussian Mixtures: ML Parameter Estimation

The maximum likelihood solutions are of the form:

$$\hat{\boldsymbol{\mu}}_k = \frac{\frac{1}{n}\sum_i \hat{P}(\omega_k|\boldsymbol{x}_i)\boldsymbol{x}_i}{\frac{1}{n}\sum_i \hat{P}(\omega_k|\boldsymbol{x}_i)}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\frac{1}{n}\sum_i \hat{P}(\omega_k|\boldsymbol{x}_i)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^t}{\frac{1}{n}\sum_i \hat{P}(\omega_k|\boldsymbol{x}_i)}$$

$$\hat{P}(\omega_k) = \frac{1}{n}\sum_i \hat{P}(\omega_k|\boldsymbol{x}_i)$$

Gaussian Mixtures: ML Parameter Estimation

- The ML solutions are typically solved iteratively:
  - Select a set of initial estimates for $\hat{P}(\omega_k)$, $\hat{\boldsymbol{\mu}}_k$, $\hat{\boldsymbol{\Sigma}}_k$
  - Use a set of n samples to reestimate the mixture parameters until some kind of convergence is found
- Clustering procedures are often used to provide the initial parameter estimates
- Similar to K-means clustering procedure

Example: 4 Samples, 2 Densities

1. Data: $\mathcal{X} = \{x_1, x_2, x_3, x_4\} = \{2, 1, -1, -2\}$
2. Init: $p(x|\omega_1) \sim N(1, 1) \quad p(x|\omega_2) \sim N(-1, 1) \quad P(\omega_i) = 0.5$
3. Estimate:

	x_1	x_2	x_3	x_4
$P(\omega_1\|x)$	0.98	0.88	0.12	0.02
$P(\omega_2\|x)$	0.02	0.12	0.88	0.98

$$p(\mathcal{X}) \propto (e^{-0.5} + e^{-4.5})(e^{0} + e^{-2})(e^{0} + e^{-2})(e^{-0.5} + e^{-4.5})0.5^4$$

4. Recompute mixture parameters (only shown for ω_1):

$$\hat{P}(\omega_1) = \frac{.98+.88+.12+.02}{4} = 0.5$$

$$\hat{\mu}_1 = \frac{.98(2)+.88(1)+.12(-1)+.02(-2)}{.98+.88+.12+.02} = 1.34$$

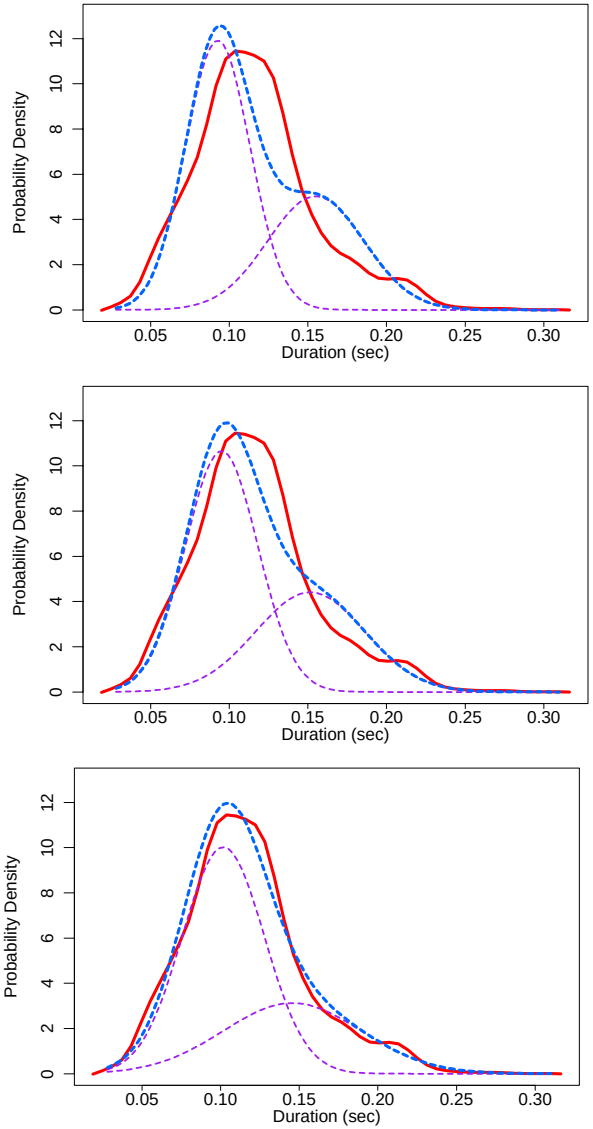
$$\hat{\sigma}_1^2 = \frac{.98(2-1.34)^2+.88(1-1.34)^2+.12(-1-1.34)^2+.02(-2-1.34)^2}{.98+.88+.12+.02} = 0.70$$

5. Repeat steps 3,4 until convergence

[s] Duration: 2 Densities

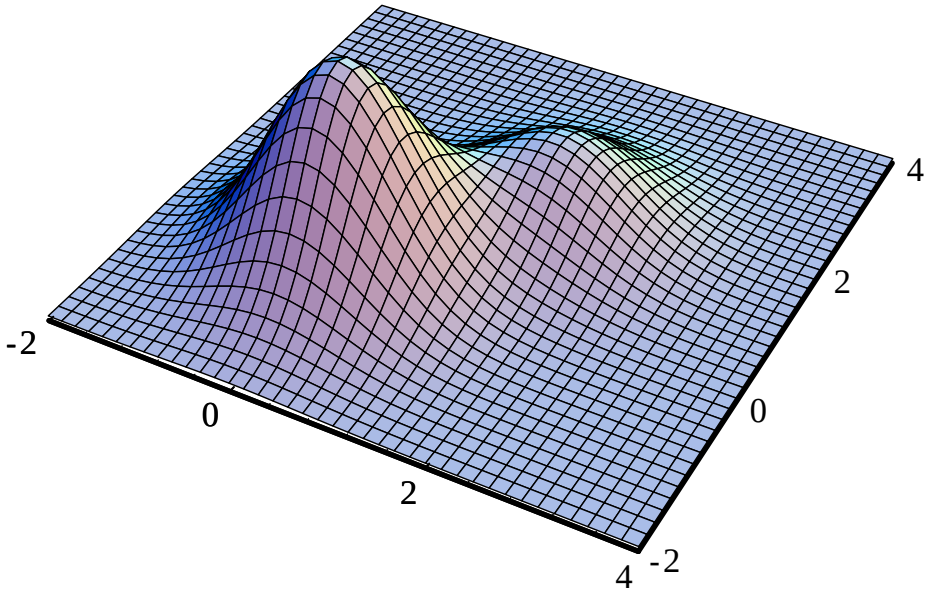
Iter	μ_1	μ_2	σ_1	σ_2
0	152	95	35	23
1	150	97	37	24
2	148	98	39	25
3	147	100	41	25
4	146	100	42	26
5	146	102	43	26
6	146	102	44	26
7	145	102	44	26

Iter	$P(\omega_1)$	$P(\omega_2)$	$\log p(X)$
0	.384	.616	2.727
1	.376	.624	2.762
2	.369	.631	2.773
3	.362	.638	2.778
4	.356	.644	2.781
5	.349	.651	2.783
6	.344	.656	2.784
7	.338	.662	2.785

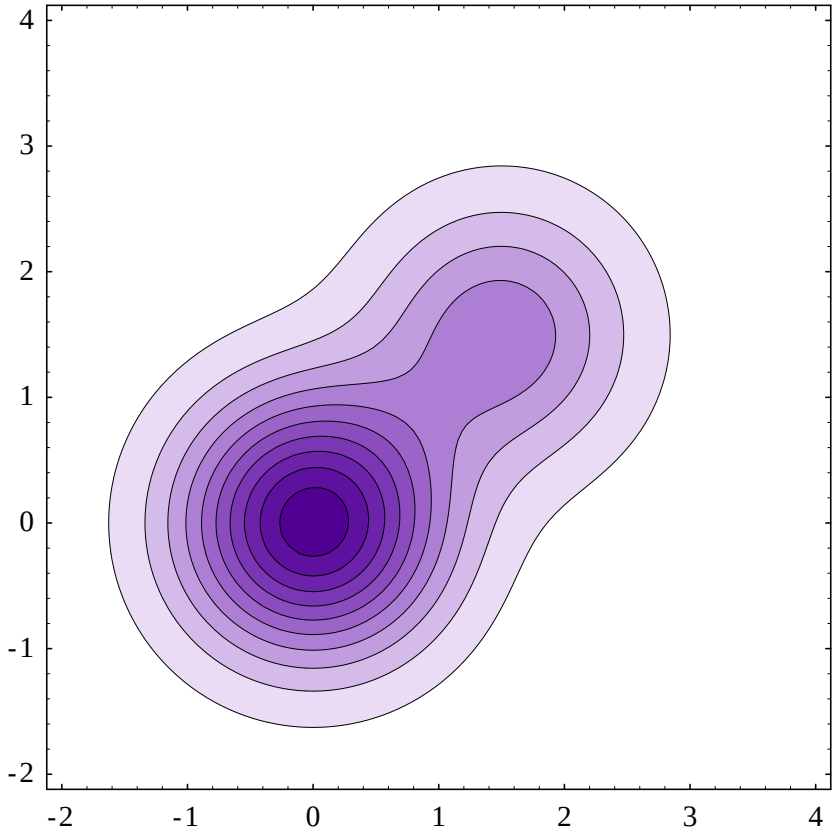


Gaussian Mixture Example: Two Dimensions

3-Dimensional PDF

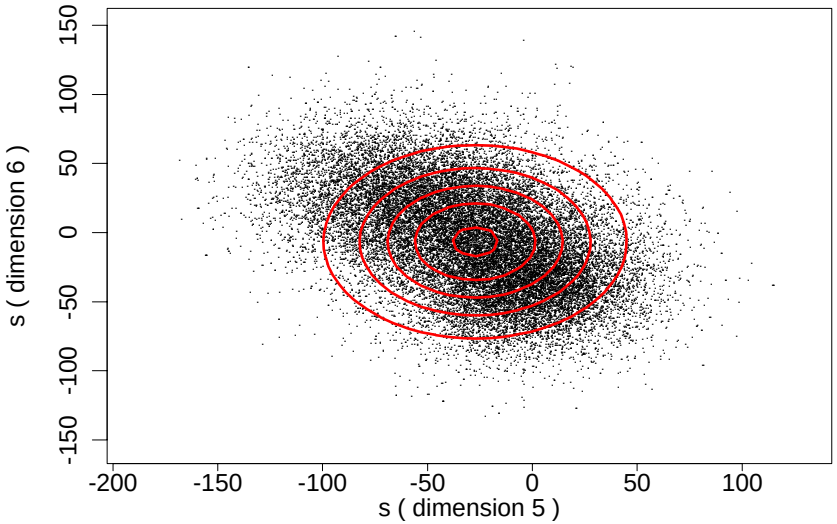


PDF Contour

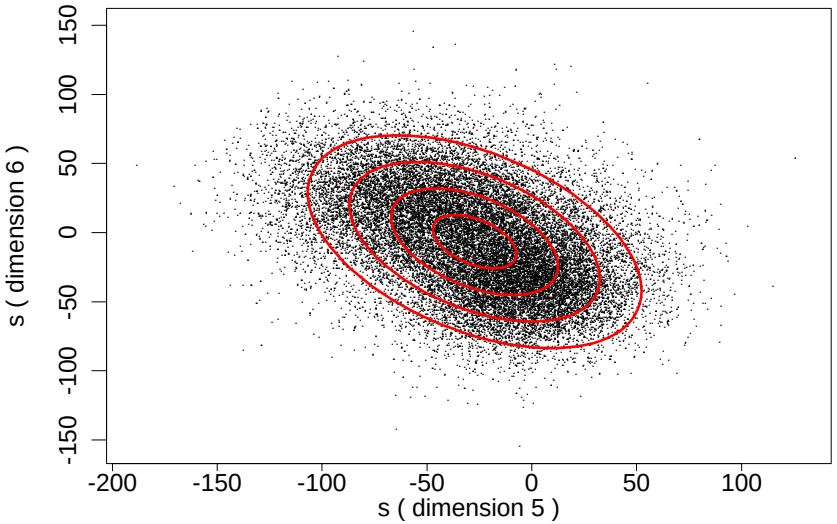


Two Dimensional Mixtures

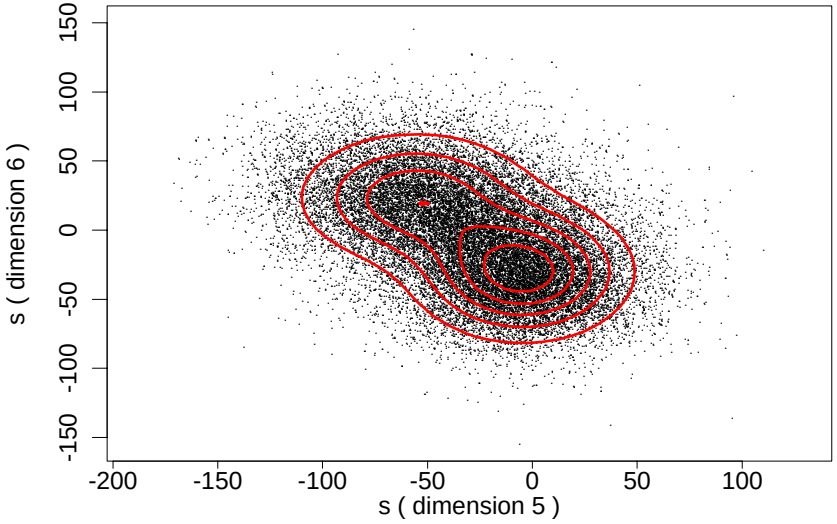
Diagonal Covariance



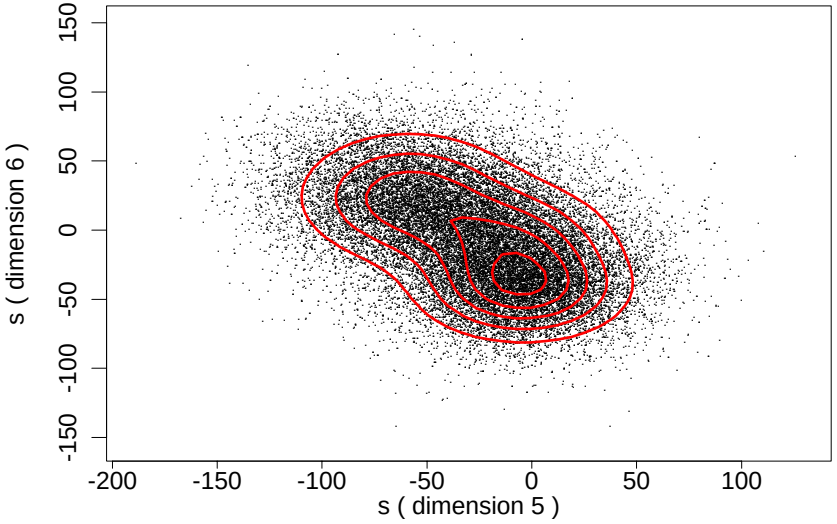
Full Covariance



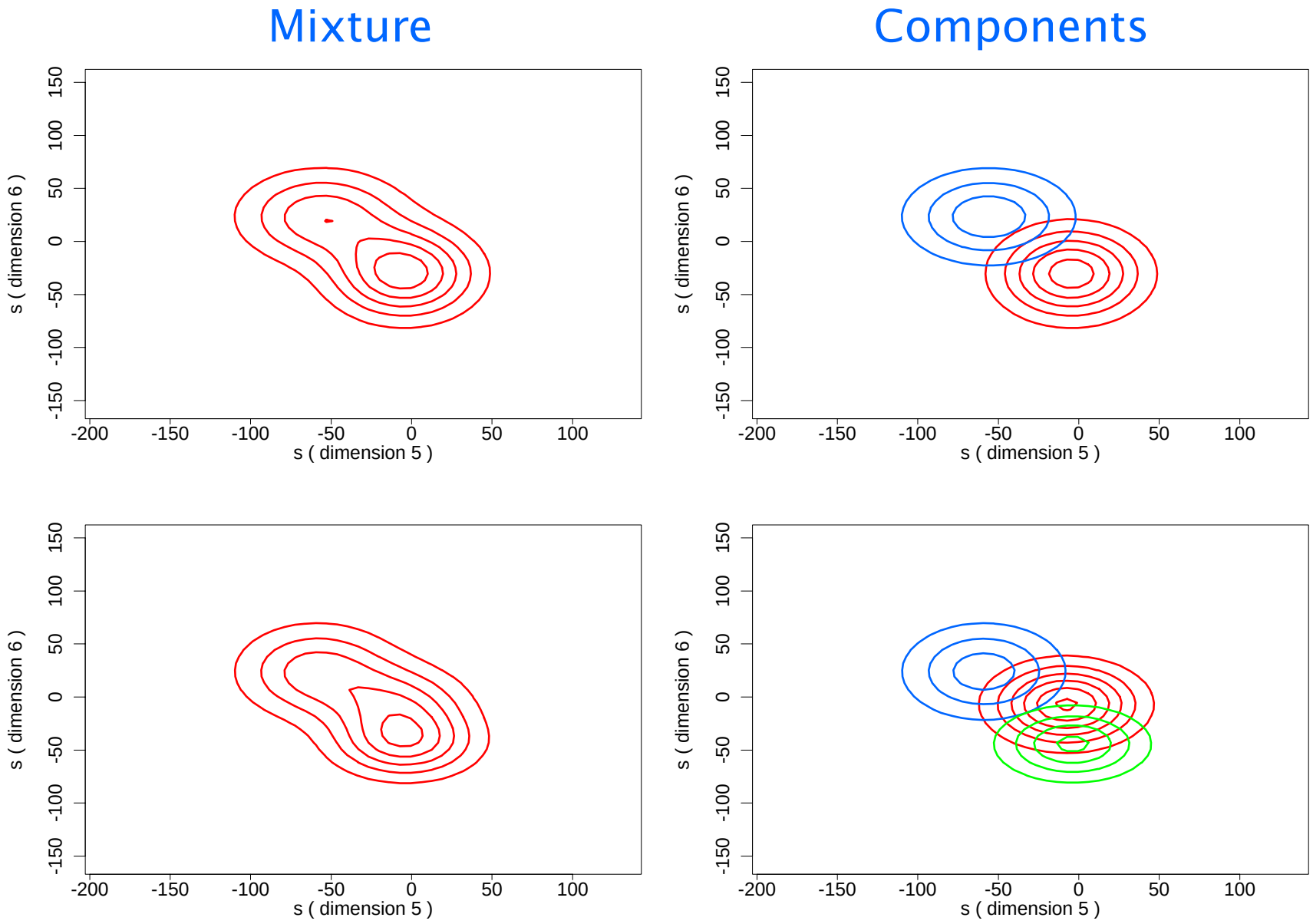
Two Mixtures



Three Mixtures



Two Dimensional Components

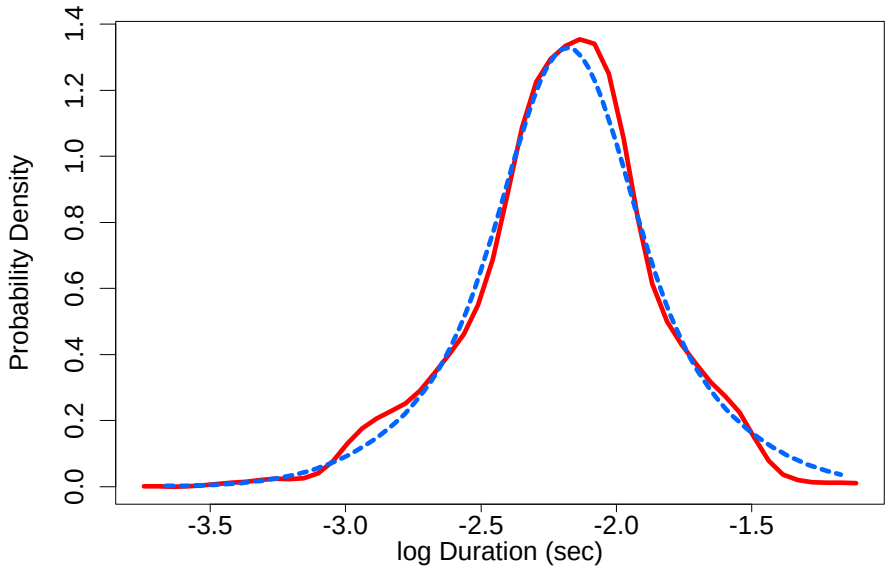


Mixture of Gaussians: Implementation Variations

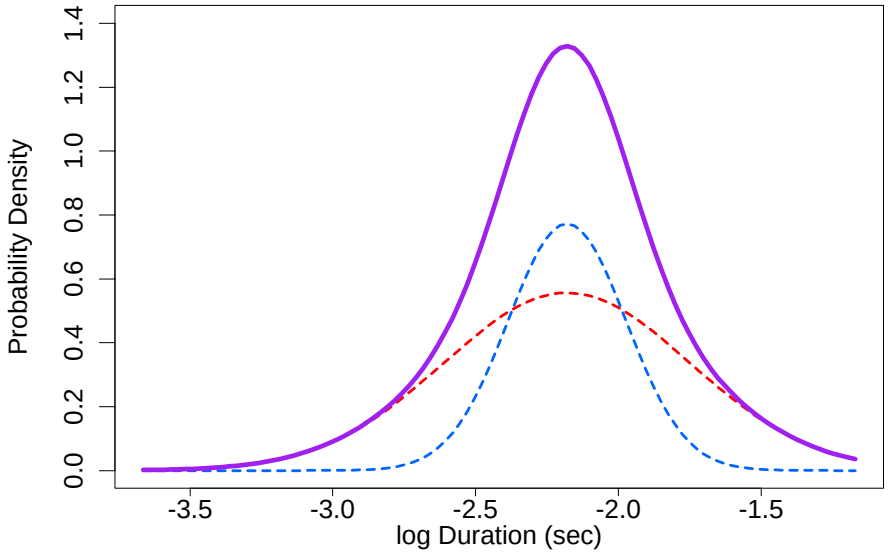
- Diagonal Gaussians are often used instead of full-covariance Gaussians
  - Can reduce the number of parameters
  - Can potentially model the underlying PDF just as well if enough components are used
- Mixture parameters are often constrained to be the same in order to reduce the number of parameters which need to be estimated
  - Richter Gaussians share the same mean in order to better model the PDF tails
  - Tied-Mixtures share the same Gaussian parameters across *all* classes. Only the mixture weights $\hat{P}(\omega_i)$ are class specific. (Also known as semi-continuous)

Richter Gaussian Mixtures

[s] Log Duration: 2 Richter Gaussians



Richter Density



Richter Components

Expectation-Maximization (EM)

- Used for determining parameters, θ, for incomplete data, $\mathcal{X} = \{\boldsymbol{x}_i\}$ (i.e., unsupervised learning problems)
- Introduces variable, $\mathcal{Z} = \{z_j\}$, to make data complete so θ can be solved using conventional ML techniques

$$\log L(\theta) = \log p(\mathcal{X}, \mathcal{Z}|\theta) = \sum_{i,j} \log p(\boldsymbol{x}_i, z_j|\theta)$$

- In reality, z_j can only be estimated by $P(z_j|\boldsymbol{x}_i, \theta)$, so we can only compute the expectation of $\log L(\theta)$

$$\mathcal{E} = E(\log L(\theta)) = \sum_i \sum_j P(z_j|\boldsymbol{x}_i, \theta) \log p(\boldsymbol{x}_i, z_j|\theta)$$

- EM solutions are computed iteratively until convergence

1. Compute the expectation of $\log L(\theta)$
2. Compute the values θ', which maximize $\mathcal{E}$

EM Parameter Estimation: 1D Gaussian Mixture Means

- Let z_i be the component id, $\{\omega_j\}$, which x_i belongs to

$$\mathcal{E} = E(\log L(\theta)) = \sum_i \sum_j P(z_j|x_i, \theta) \log p(x_i, z_j|\theta)$$

- Convert to mixture component notation:

$$\mathcal{E} = E(\log L(\mu_k)) = \sum_i \sum_j P(\omega_j|x_i) \log p(x_i, \omega_j)$$

- Differentiate with respect to μ_k:

$$\frac{\partial \mathcal{E}}{\partial \mu_k} = \sum_i P(\omega_k|x_i) \frac{\partial}{\partial \mu_k} \log p(x_i, \omega_k) = \sum_i P(\omega_k|x_i)(\frac{x_i - \mu_k}{\sigma_k^2}) = 0$$

$$\hat{\mu}_k = \frac{\sum_i P(\omega_k|x_i) x_i}{\sum_i P(\omega_k|x_i)}$$

EM Properties

- Each iteration of EM will increase the likelihood of $\mathcal{X}$

$$\log \frac{p(\mathcal{X}|\theta')}{p(\mathcal{X}|\theta)} = \sum_i \log \frac{p(\boldsymbol{x}_i|\theta')}{p(\boldsymbol{x}_i|\theta)} = \sum_i \sum_j P(z_j|\boldsymbol{x}_i, \theta) \log \frac{p(\boldsymbol{x}_i|\theta')}{p(\boldsymbol{x}_i|\theta)}$$

$$= \sum_i \sum_j P(z_j|\boldsymbol{x}_i, \theta)(\log \frac{p(\boldsymbol{x}_i|\theta')}{p(\boldsymbol{x}_i, z_j|\theta')} \frac{p(\boldsymbol{x}_i, z_j|\theta)}{p(\boldsymbol{x}_i|\theta)} + \log \frac{p(\boldsymbol{x}_i, z_j|\theta')}{p(\boldsymbol{x}_i, z_j|\theta)})$$

- Using Bayes rule and the Kullback-Liebler distance metric:

$$\frac{p(\boldsymbol{x}_i, z_j|\theta)}{p(\boldsymbol{x}_i|\theta)} = P(z_j|\boldsymbol{x}_i, \theta) \qquad \sum_j P(z_j|\boldsymbol{x}_i, \theta) \log \frac{P(z_j|\boldsymbol{x}_i, \theta)}{P(z_j|\boldsymbol{x}_i, \theta')} \geq 0$$

- Since θ' was determined to maximize $E(\log L(\theta))$:

$$\sum_i \sum_j P(z_j|\boldsymbol{x}_i, \theta) \log \frac{p(\boldsymbol{x}_i, z_j|\theta')}{p(\boldsymbol{x}_i, z_j|\theta)} \geq 0$$

- Combining these two properties: $p(\mathcal{X}|\theta') \geq p(\mathcal{X}|\theta)$

Dimensionality Reduction

- Given a training set, PDF parameter estimation becomes less robust as dimensionality increases
- Increasing dimensions can make it more difficult to obtain insights into any underlying structure
- Analytical techniques exist which can transform a sample space to a different set of dimensions
  - If original dimensions are correlated, the same information may require fewer dimensions
  - The transformed space will often have more Normal distribution than the original space
  - If the new dimensions are orthogonal, it could be easier to model the transformed space

Principal Components Analysis

- Linearly transforms d-dimensional vector, $\boldsymbol{x}$, to d' dimensional vector, $\boldsymbol{y}$, via orthonormal vectors, $\boldsymbol{W}$

$$\boldsymbol{y} = \boldsymbol{W}^t \boldsymbol{x} \qquad \boldsymbol{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{d'}\} \qquad \boldsymbol{W}^t \boldsymbol{W} = \boldsymbol{I}$$

- If $d' < d$, $\boldsymbol{x}$ can be only partially reconstructed from $\boldsymbol{y}$

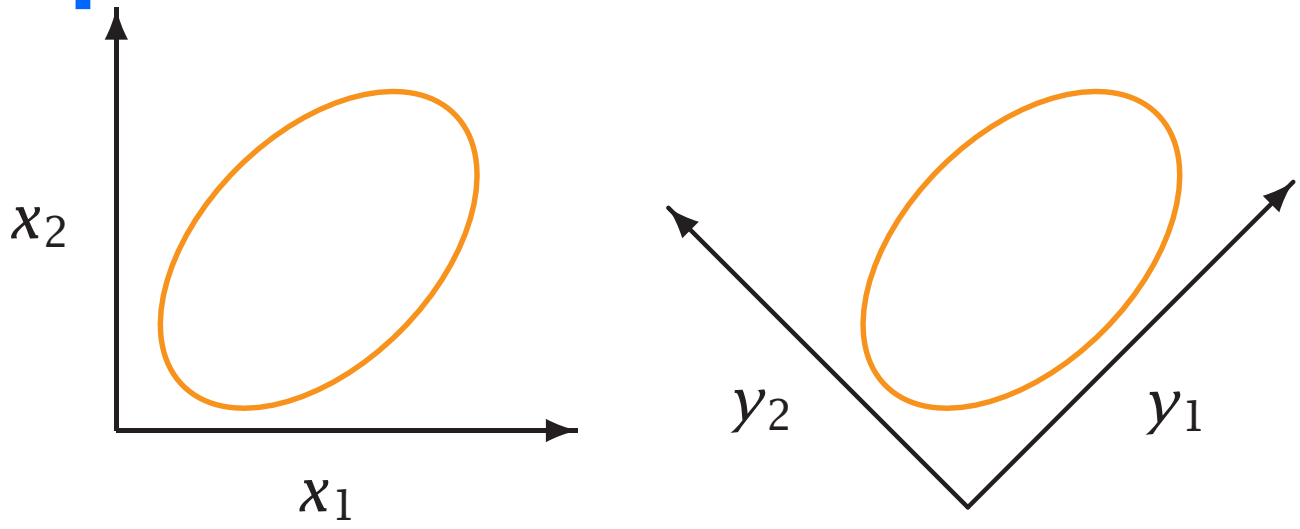
$$\hat{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{y}$$

- Principal components, $\boldsymbol{W}$, minimize the distortion, $\mathcal{D}$, between $\boldsymbol{x}$, and $\hat{\boldsymbol{x}}$, on training data, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

$$\mathcal{D} = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2$$

- Also known as Karhunen-Loève (K-L) expansion ($\boldsymbol{w}_i$'s are sinusoids for some stochastic processes)

PCA Computation



- $\boldsymbol{W}$ corresponds to the first d' eigenvectors, $\boldsymbol{P}$, of $\boldsymbol{\Sigma}$

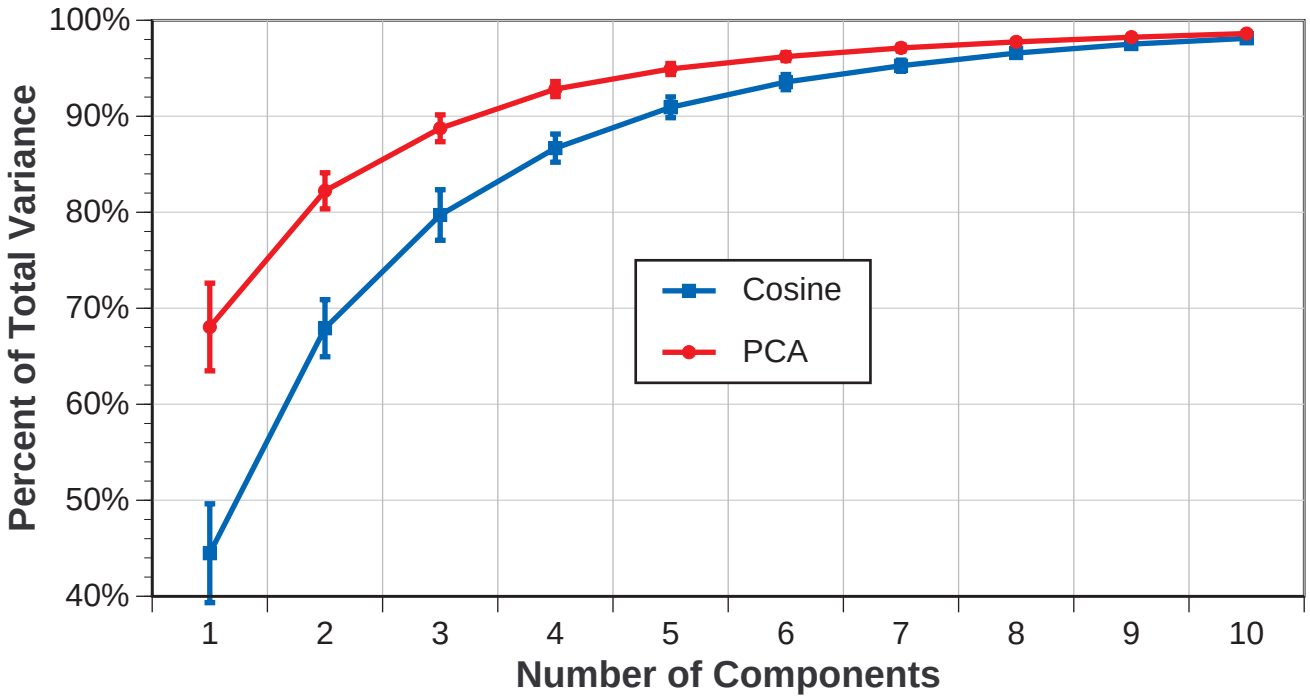
$$\boldsymbol{P} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d\} \qquad \boldsymbol{\Sigma} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^t \qquad \boldsymbol{w}_i = \boldsymbol{e}_i$$

- Full covariance structure of original space, $\boldsymbol{\Sigma}$, is transformed to a diagonal covariance structure, $\boldsymbol{\Lambda}'$
- Eigenvalues, $\{\lambda_1, \ldots, \lambda_{d'}\}$, represent the variances in $\boldsymbol{\Lambda}'$
- Axes in d'-space contain maximum amount of variance

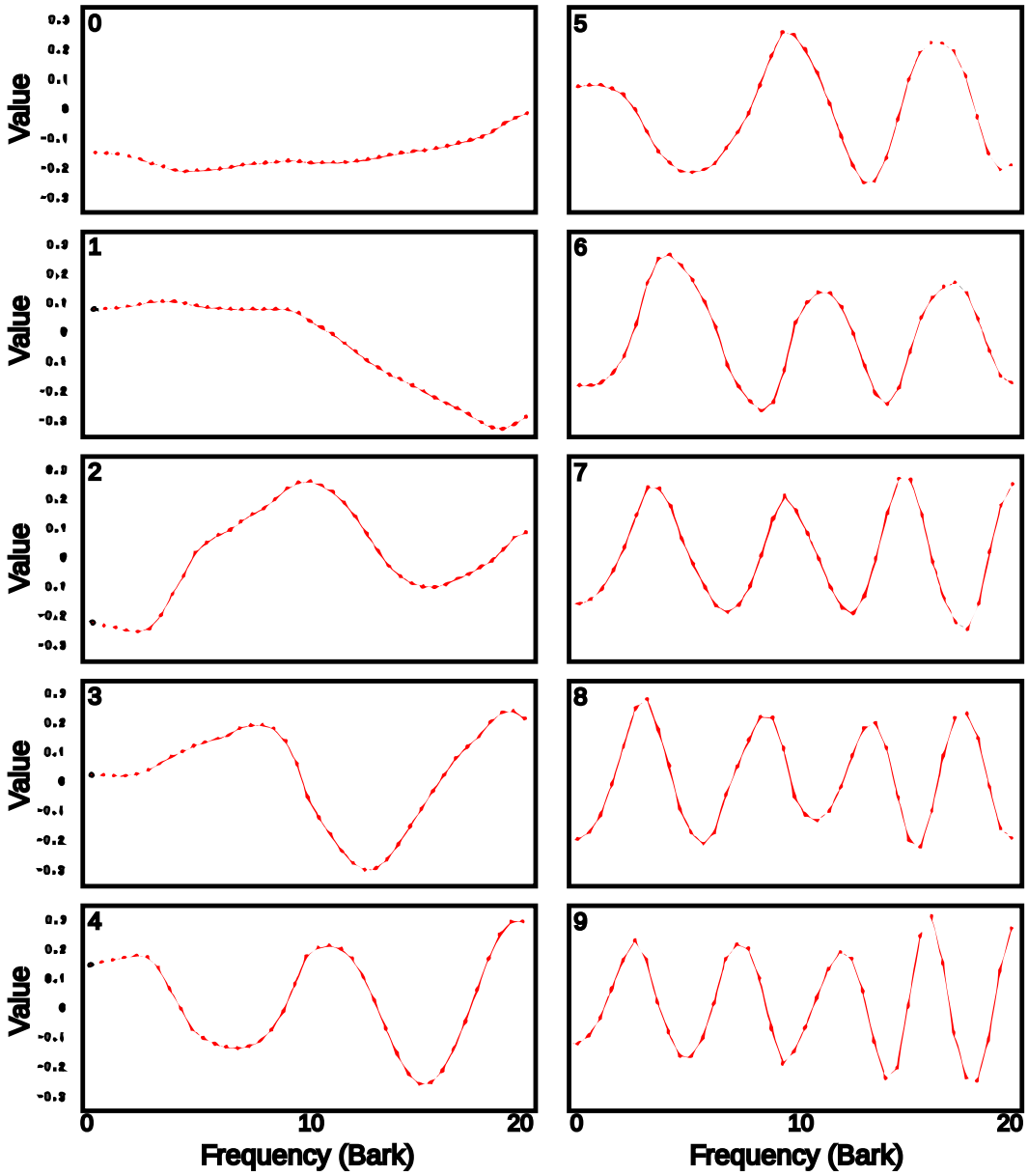
$$\mathcal{D} = \sum_{i=d'+1}^{d} \lambda_i$$

PCA Example

- Original feature vector mean rate response ($d = 40$)
- Data obtained from 100 speakers from TIMIT corpus
- First 10 components explains 98% of total variance

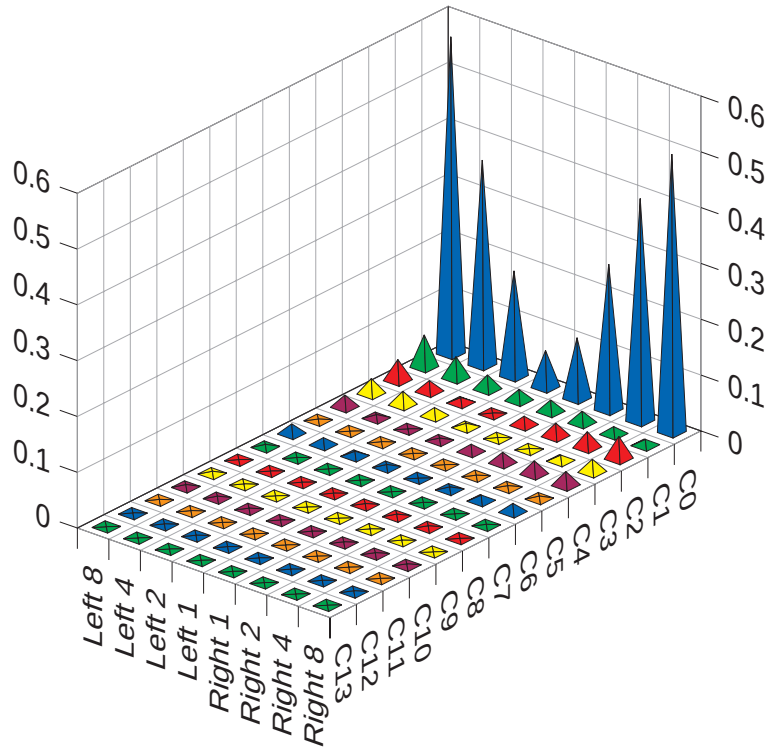


PCA Example

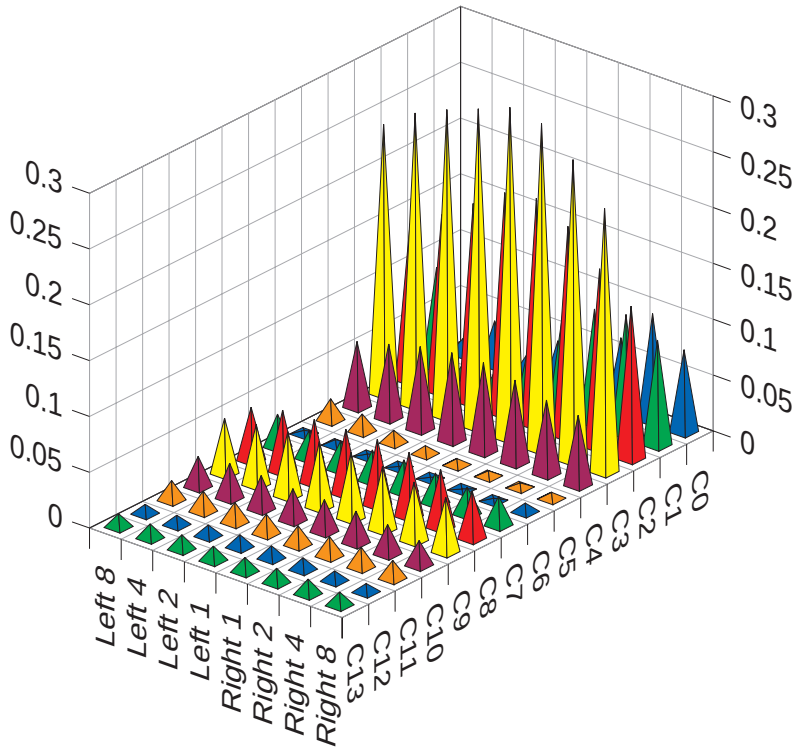


PCA for Boundary Classification

- Eight non-uniform averages from 14 MFCCs
- First 50 dimensions used for classification



Second Component



Seventh Component

PCA Issues

- PCA can be performed using
  - Covariances Σ
  - Correlation coefficients matrix $\mathcal{P}$

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} \qquad |\rho_{ij}| \leq 1$$

- $\mathcal{P}$ is usually preferred when the input dimensions have significantly different ranges
- PCA can be used to normalize or whiten original d-dimensional space to simplify subsequent processing

$$\Sigma \Longrightarrow \mathcal{P} \Longrightarrow \Lambda \Longrightarrow \boldsymbol{I}$$

- Whitening operation can be done in one step: $\boldsymbol{z} = \boldsymbol{V}^t \boldsymbol{x}$

Significance Testing

- To properly compare results from different classifier algorithms, A_1, and A_2, it is necessary to perform significance tests
  - Large differences can be insignificant for small test sets
  - Small differences can be significant for large test sets
- General significance tests evaluate the hypothesis that the probability of being correct, p_i, of both algorithms is the same
- The most powerful comparisons can be made using common train and test corpora, and common evaluation criterion
  - Results reflect differences in algorithms rather than accidental differences in test sets
  - Significance tests can be more precise when identical data are used since they can focus on tokens misclassified by only one algorithm, rather than on all tokens

McNemar's Significance Test

- When algorithms A_1 and A_2 are tested on identical data we can collapse the results into a $2x2$ matrix of counts

A_1/A_2	Correct	Incorrect
Correct	n_{00}	n_{01}
Incorrect	n_{10}	n_{11}

- To compare algorithms, we test the null hypothesis $\mathcal{H}_0$ that $p_1 = p_2$, or $n_{01} = n_{10}$, or $q = \dfrac{n_{01}}{n_{01} + n_{10}} = \dfrac{1}{2}$
- Given $\mathcal{H}_0$, the probability of observing k tokens asymmetrically classified out of $n = n_{01} + n_{10}$ has a Binomial PMF

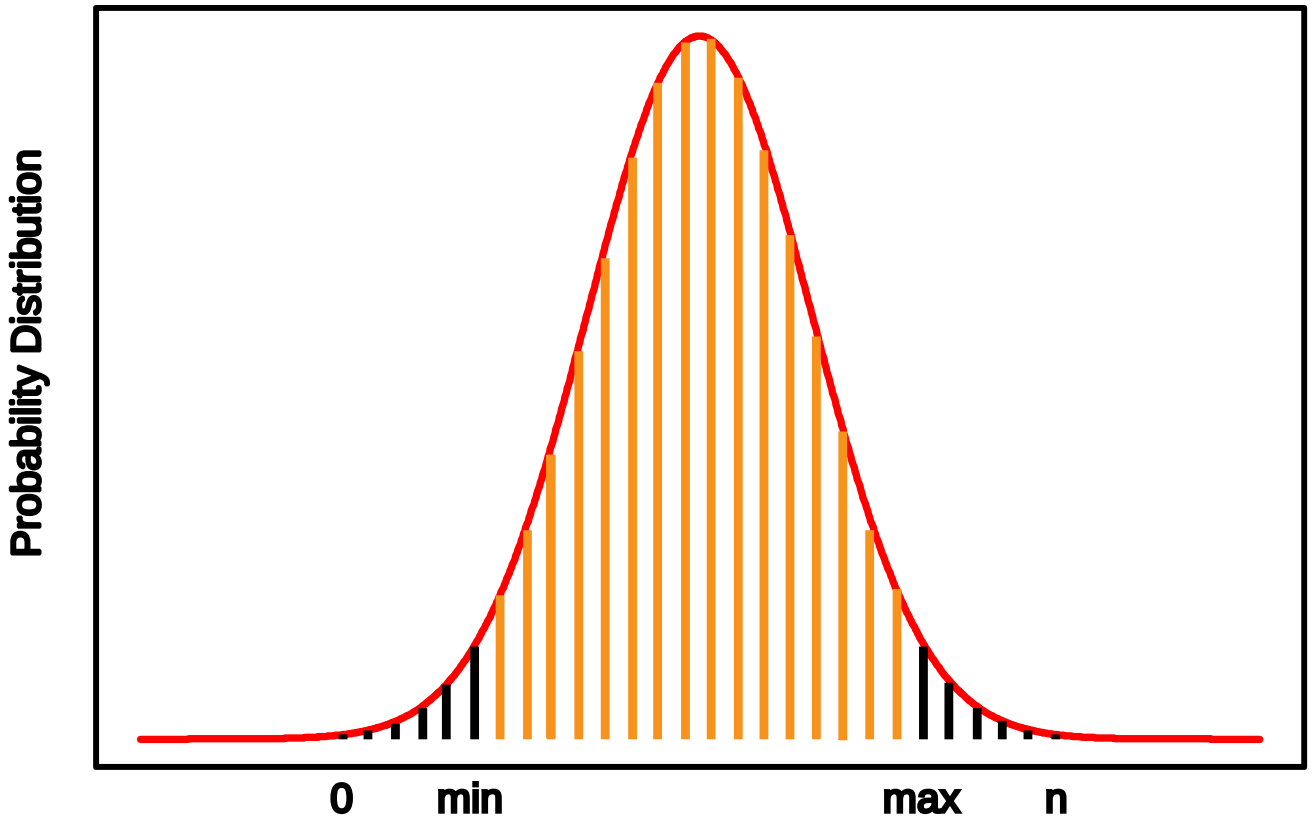
$$P(k) = \binom{n}{k} \left(\frac{1}{2}\right)^n$$

- McNemar's Test measures the probability, P, of all cases that meet or exceed the observed asymmetric distribution, and tests $P < \alpha$

McNemar's Significance Test (cont't)

- The probability, P, is computed by summing up the PMF tails

$$P = \sum_{k=0}^{l} P(k) + \sum_{k=m}^{n} P(k) \quad l = \min(n_{01}, n_{10}) \quad m = \max(n_{01}, n_{10})$$



- For large n, a Normal distribution is often assumed

Significance Test Example (Gillick and Cox, 1989)

- Common test set of 1400 tokens
- Algorithms A_1 and A_2 make 72 and 62 errors
- Are the differences significant?

	A_2	
A_1	1266	62
	72	0

$n = 134 \quad m = 72 \quad P = 0.437$

	A_2	
A_1	1325	3
	13	59

$n = 16 \quad m = 13 \quad P = 0.0213$

	A_2	
A_1	1328	0
	10	62

$n = 10 \quad m = 10 \quad P = 0.0020$

References

- Huang, Acero, and Hon, *Spoken Language Processing*, Prentice-Hall, 2001.
- Duda, Hart and Stork, *Pattern Classification*, John Wiley & Sons, 2001.
- Jelinek, *Statistical Methods for Speech Recognition.* MIT Press, 1997.
- Bishop, *Neural Networks for Pattern Recognition,* Clarendon Press, 1995.
- Gillick and Cox, Some Statistical Issues in the Comparison of Speech Recognition Algorithms, *Proc. ICASSP*, 1989.