BAYESIAN DECISION THEORY

J. Elder

CSE 4404/5327 Introduction to Machine Learning and Pattern Recognition

Credits

- Some of these slides were sourced and/or modified from:
  - Christopher Bishop, Microsoft UK
  - Simon Prince, University College London
  - Sergios Theodoridis, University of Athens & Konstantinos Koutroumbas, National Observatory of Athens

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. What are Bayes Nets?

Problems for this Meeting

- Problems 2.1-2.4
- Assigned Problem: 2.2

Bayesian Decision Theory: Topics

1. **Probability**
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. What are Bayes Nets?

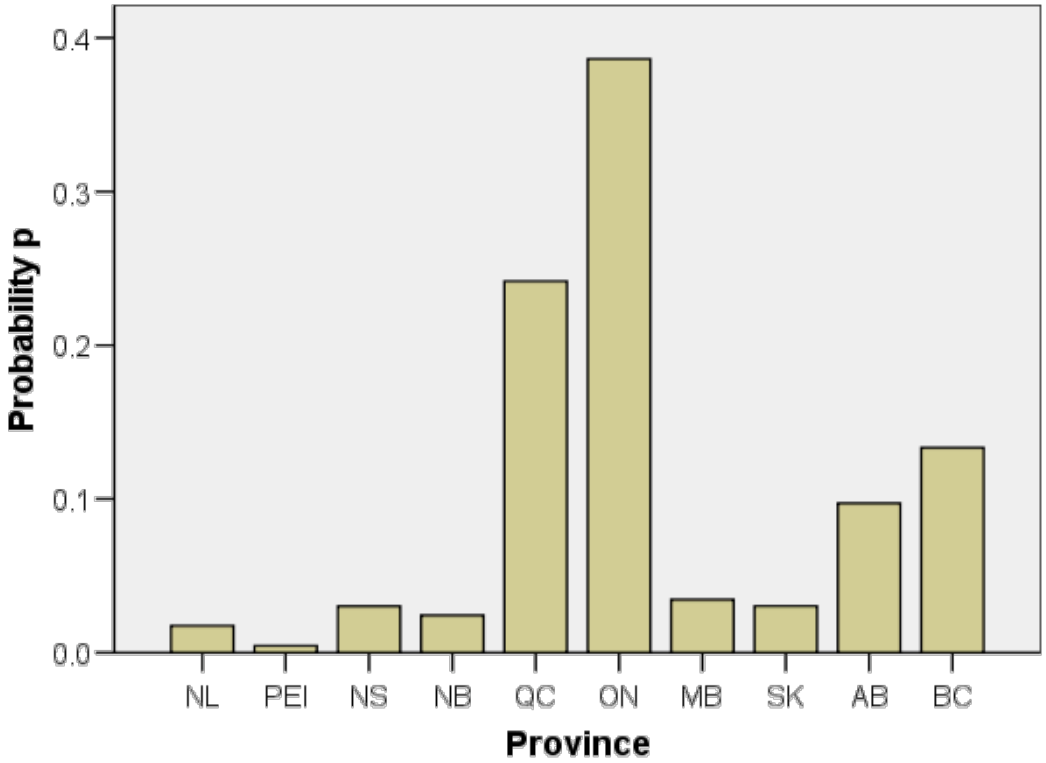
Topic 1. Probability

Random Variables

- A **random variable** is a variable whose value is uncertain.
- For example, the height of a randomly selected person in this class is a random variable – I won't know its value until the person is selected.
- Note that we are not completely uncertain about most random variables.
  - For example, we know that height will probably be in the 5'-6' range.
  - In addition, 5'6" is more likely than 5'0" or 6'0".
- The function that describes the probability of each possible value of the random variable is called a **probability distribution**.

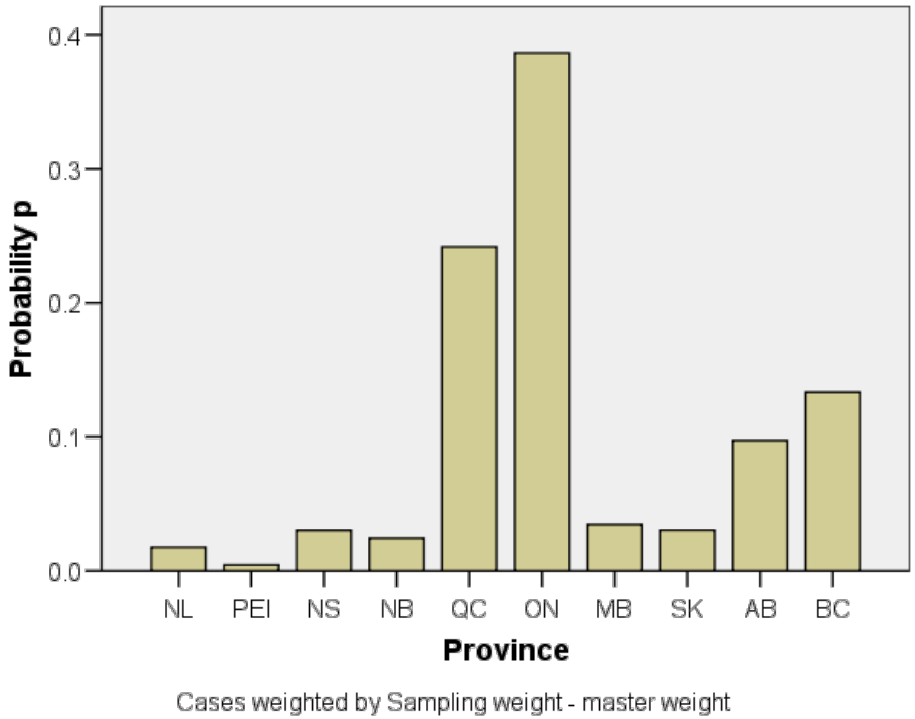
Probability Distributions

- For a **discrete** distribution, the probabilities over all possible values of the random variable must **sum** to 1.



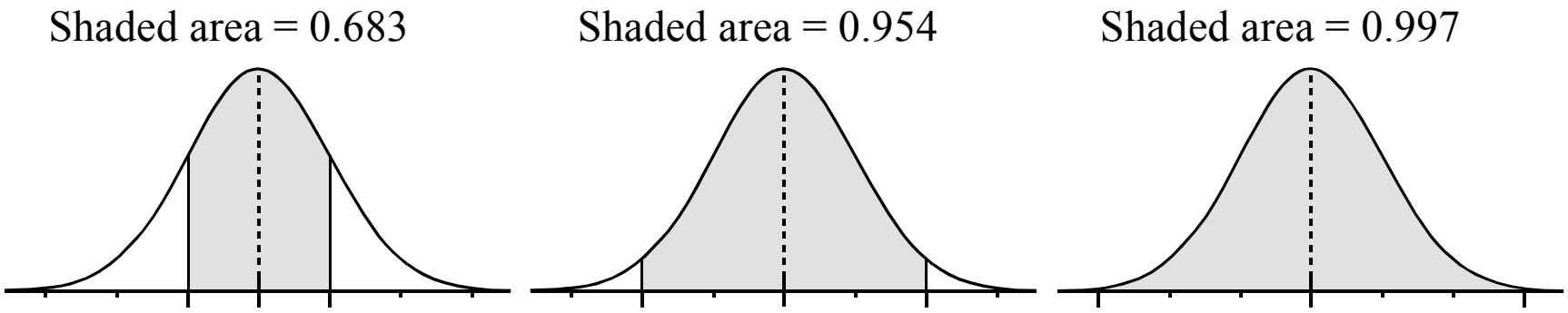
Probability Distributions

- For a **discrete** distribution, we can talk about the probability of a particular score occurring, e.g., p(Province = Ontario) = 0.36.
- We can also talk about the probability of any one of a subset of scores occurring, e.g., p(Province = Ontario or Quebec) = 0.50.
- In general, we refer to these occurrences as **events**.



Probability Distributions

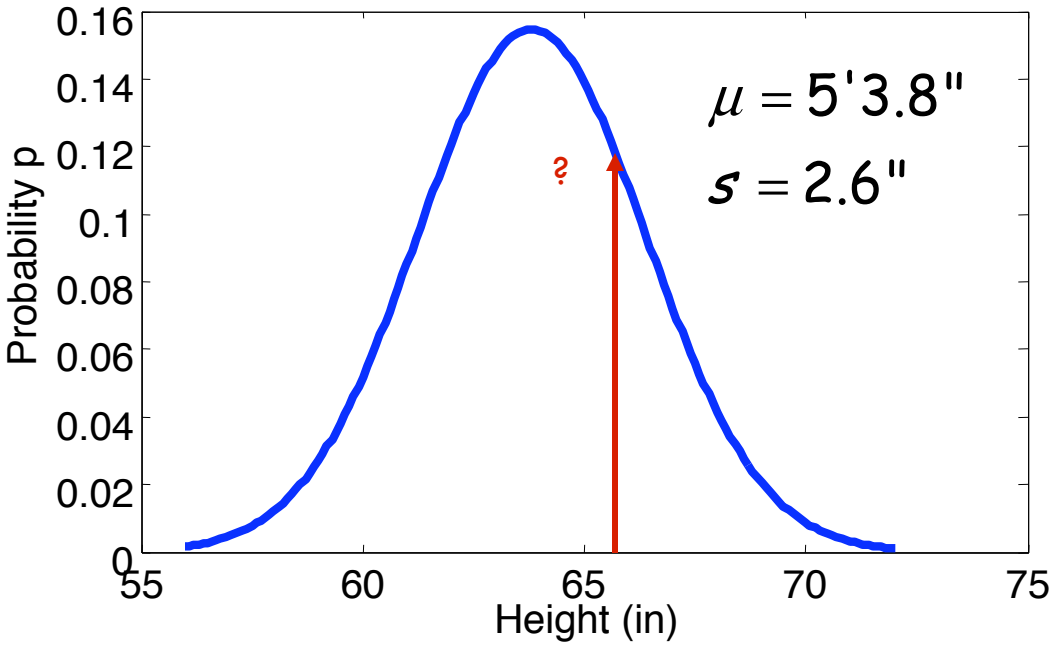
- For a **continuous** distribution, the probabilities over all possible values of the random variable must **integrate** to 1 (i.e., the area under the curve must be 1).
- Note that the height of a continuous distribution can exceed 1!



Continuous Distributions

- For continuous distributions, it **does not** make sense to talk about the probability of an exact score.
  - e.g., what is the probability that your height is exactly 65.485948467… inches?

Normal Approximation to probability distribution for height of Canadian females
(parameters from General Social Survey, 1991)

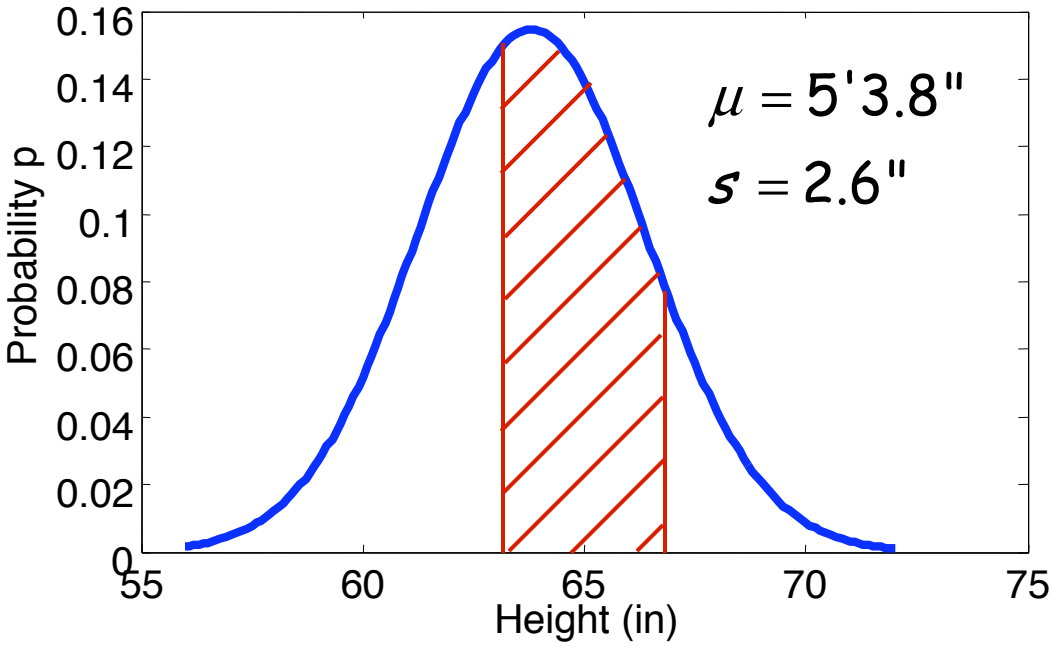


Continuous Distributions

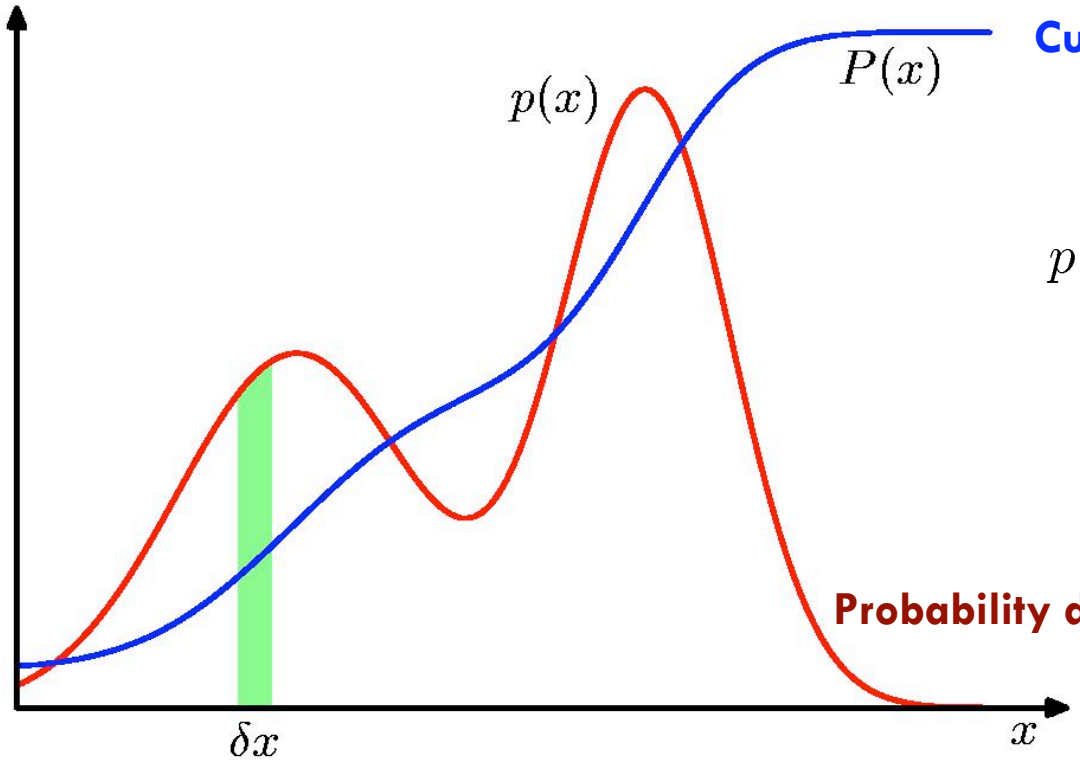
- It **does** make sense to talk about the probability of observing a score that falls within a certain range
  - e.g., what is the probability that you are between 5'3" and 5'7"?
  - e.g., what is the probability that you are less than 5'10"?

} Valid events

Normal Approximation to probability distribution for height of Canadian females
(parameters from General Social Survey, 1991)



Probability Densities



Cumulative distribution (CDF)

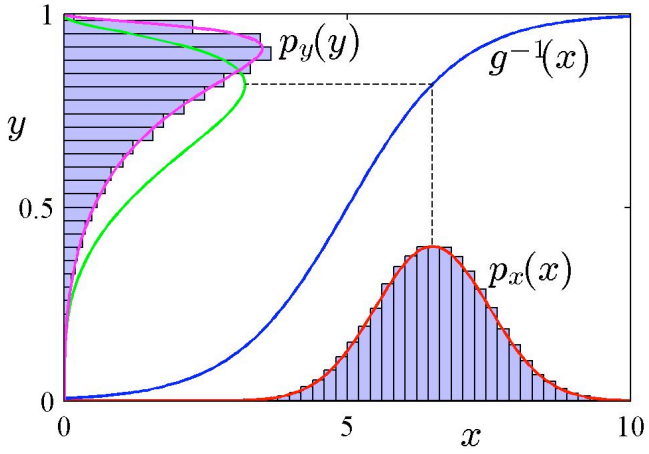
$$p(x \in (a,b)) = \int_a^b p(x)\,\mathrm{d}x$$

$$P(z) = \int_{-\infty}^{z} p(x)\,\mathrm{d}x$$

Probability density (PDF)

$$p(x) \geqslant 0 \qquad \int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1$$

Transformed Densities



Observations falling within $(x + \delta x)$ tranform to the range $(y + \delta y)$

$$\rightarrow p_x(x)|\delta x| = p_y(y)|\delta y|$$

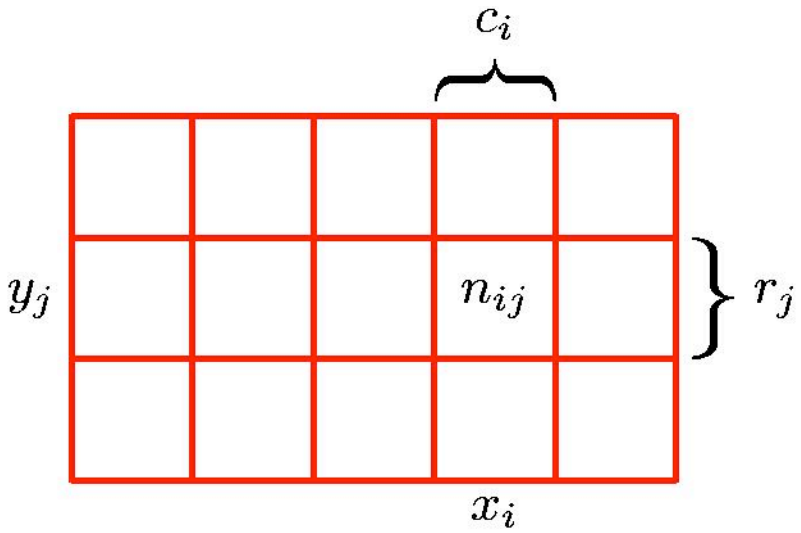
$$\rightarrow p_y(y) \simeq p_x(x)\left|\frac{\delta x}{\delta y}\right|$$

Note that in general, $\delta y \neq \delta x$.

Rather, $\frac{\delta y}{\delta x} \rightarrow \frac{dy}{dx}$ as $\delta x \rightarrow 0$.

Thus $p_y(y) = p_x(x)\left|\frac{dx}{dy}\right|$

Joint Distributions



Marginal Probability

$$p(X = x_i) = \frac{c_i}{N}.$$

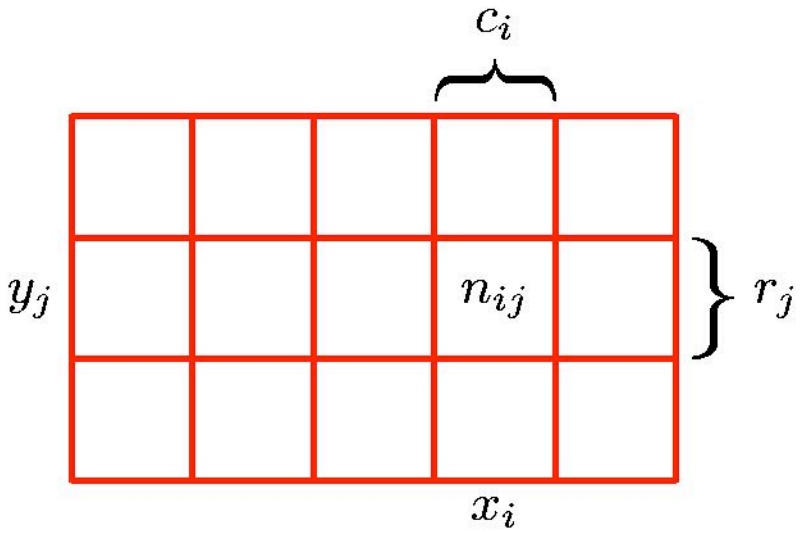
Joint Probability

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

Conditional Probability

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

Joint Distributions



Sum Rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N}\sum_{j=1}^{L} n_{ij}$$

$$= \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

Product Rule

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N}$$

$$= p(Y = y_j | X = x_i) p(X = x_i)$$

Joint Distributions: The Rules of Probability

- Sum Rule $$p(X) = \sum_Y p(X, Y)$$
- Product Rule $$p(X, Y) = p(Y|X)p(X)$$

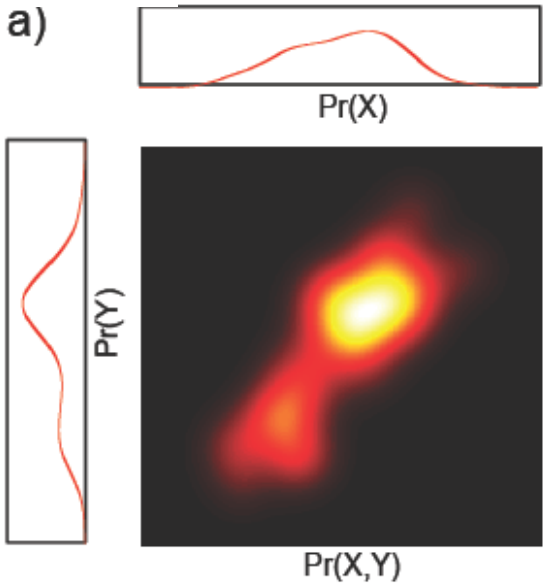
Marginalization

We can recover probability distribution of any variable in a joint distribution by integrating (or summing) over the other variables

$$Pr(X) = \int Pr(X,Y)dY$$

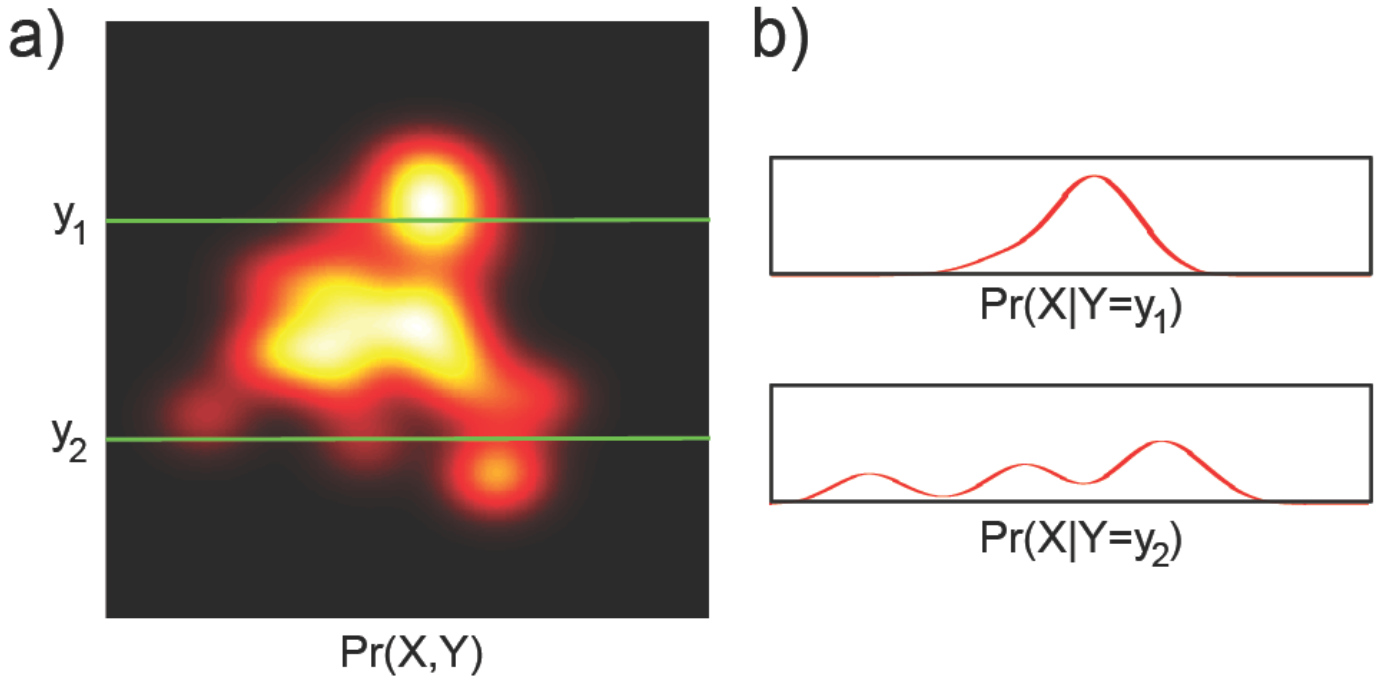
$$Pr(Y) = \int Pr(X,Y)dX$$

$$Pr(X,Y) = \sum_W \sum_Z Pr(W,X,Y,Z)$$



Conditional Probability

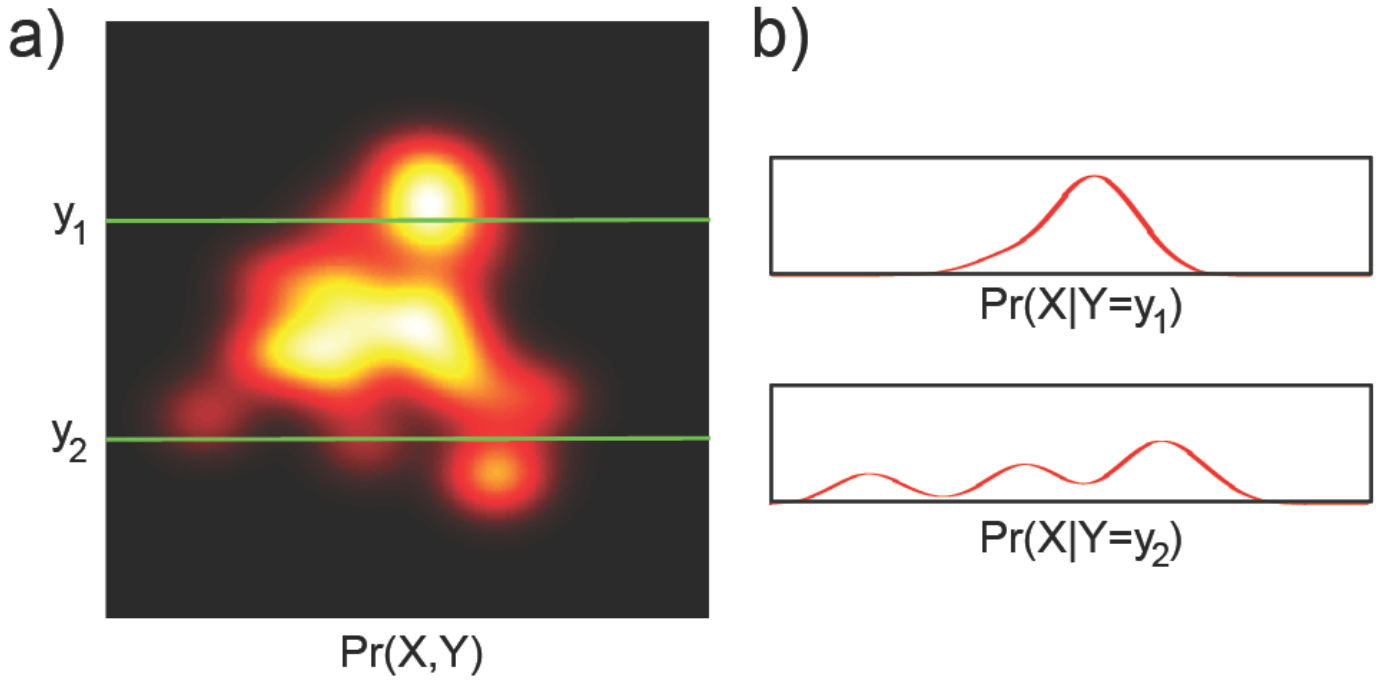
- Conditional probability of X given that Y=y* is relative propensity of variable X to take different outcomes given that Y is fixed to be equal to y*
- Written as Pr(X | Y=y*)



Conditional Probability

- Conditional probability can be extracted from joint probability
- Extract appropriate slice and normalize

$$Pr(X|Y=y^*) = \frac{Pr(X,Y=y^*)}{\int (Pr(X,Y=y^*)dX} = \frac{Pr(X,Y=y^*)}{Pr(Y=y^*)}$$



Conditional Probability

$$Pr(X|Y=y^*) = \frac{Pr(X,Y=y^*)}{\int (Pr(X,Y=y^*)dX} = \frac{Pr(X,Y=y^*)}{Pr(Y=y^*)}$$

- More usually written in compact form

$$Pr(X|Y) = \frac{Pr(X,Y)}{Pr(Y)}$$

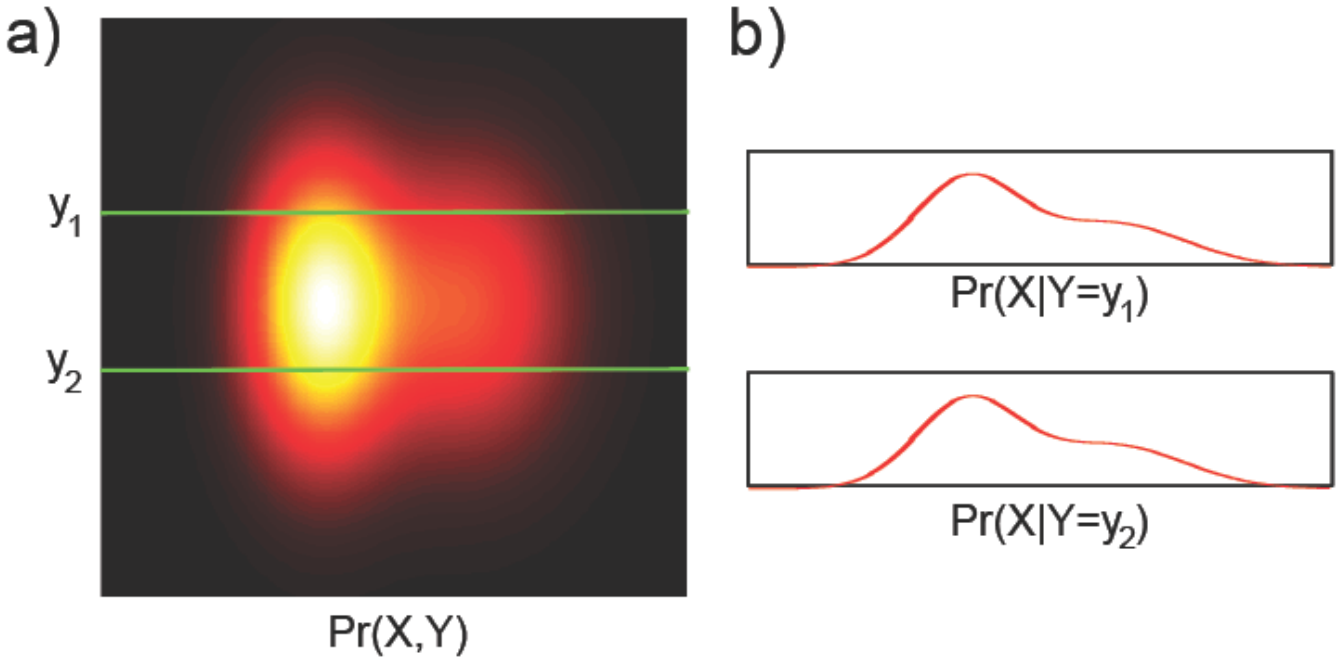
- Can be re-arranged to give

$$Pr(X,Y) = Pr(X|Y)Pr(Y)$$

Independence

- If two variables X and Y are independent then variable X tells us nothing about variable Y (and vice-versa)

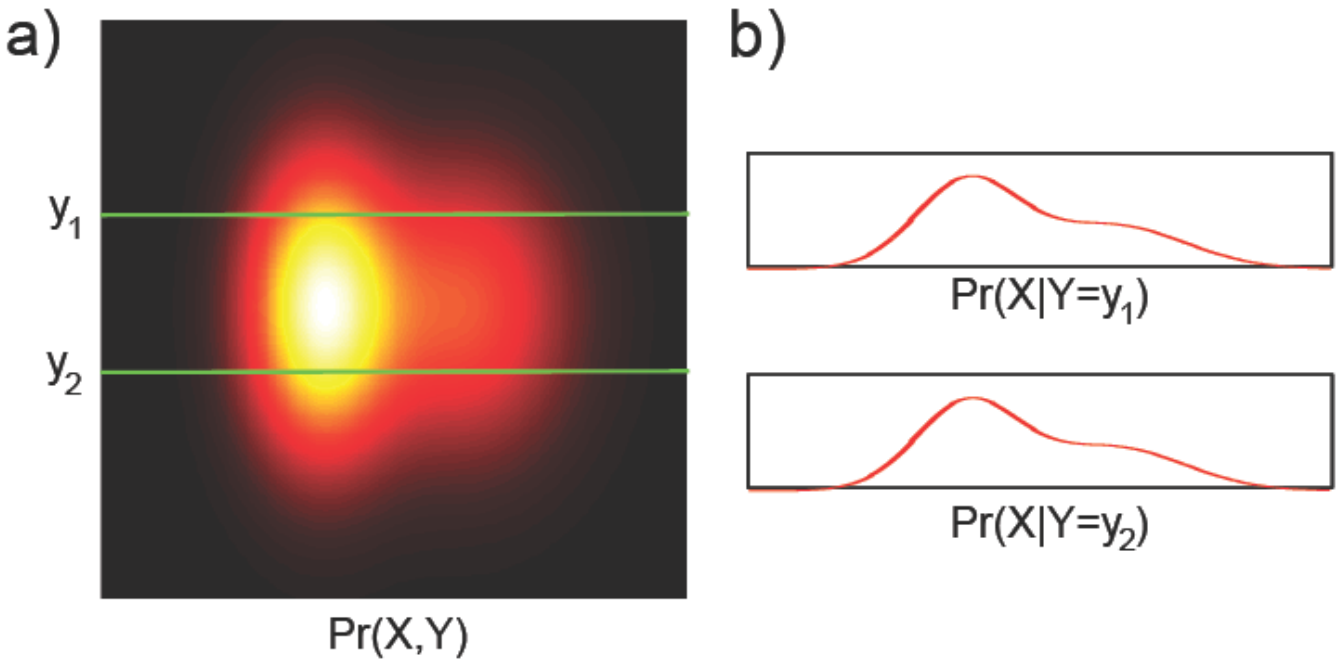
$$Pr(X|Y) = Pr(X)$$
$$Pr(Y|X) = Pr(Y)$$



Independence

- When variables are independent, the joint factorizes into a product of the marginals:

$$\begin{aligned} Pr(X,Y) &= Pr(X|Y)Pr(Y) \\ &= Pr(X)Pr(Y) \end{aligned}$$



Bayes' Rule

From before:

$$Pr(X,Y) = Pr(X|Y)Pr(Y)$$
$$Pr(X,Y) = Pr(Y|X)Pr(X)$$

Combining:

$$Pr(Y|X)Pr(X) = Pr(X|Y)Pr(Y)$$

Re-arranging:

$$Pr(Y|X) = \frac{Pr(X|Y)Pr(Y)}{Pr(X)}$$
$$= \frac{Pr(X|Y)Pr(Y)}{\int Pr(X,Y)dY}$$
$$= \frac{Pr(X|Y)Pr(Y)}{\int Pr(X|Y)Pr(Y)dY}$$

Bayes' Rule Terminology

Likelihood – propensity for observing a certain value of X given a certain value of Y

Prior – what we know about y before seeing x

$$Pr(Y|X) = \frac{Pr(X|Y)Pr(Y)}{Pr(X)}$$

Posterior – what we know about y after seeing x

Evidence –a constant to ensure that the left hand side is a valid distribution

End of Lecture 2

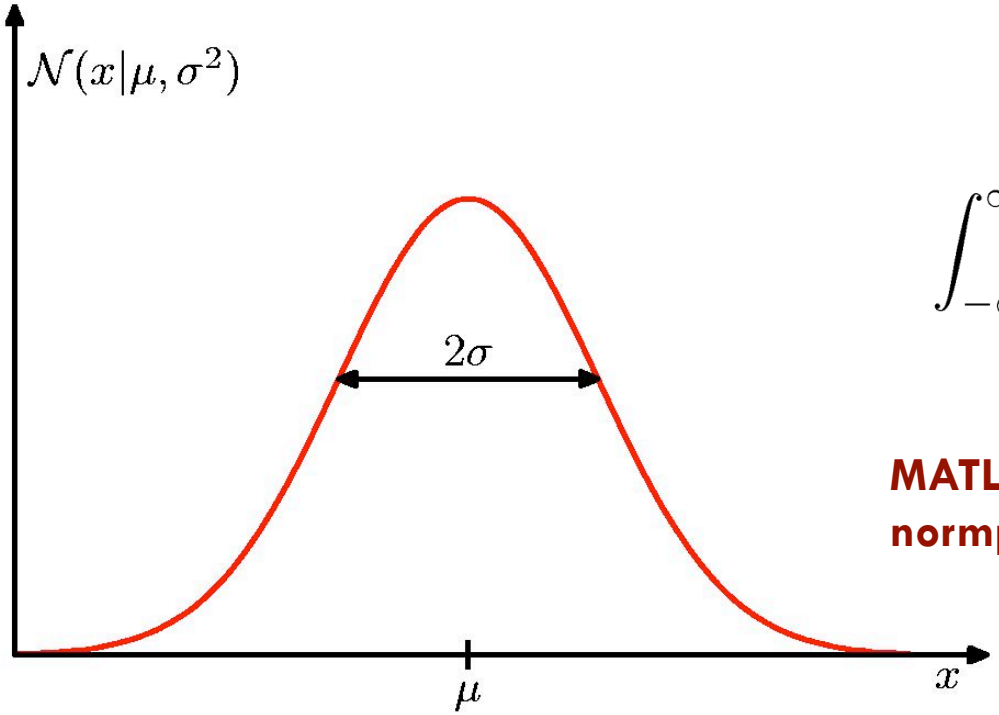
Bayesian Decision Theory: Topics

1. Probability
2. **The Univariate Normal Distribution**
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. What are Bayes Nets?

Topic 2. The Univariate Normal Distribution

The Gaussian Distribution

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



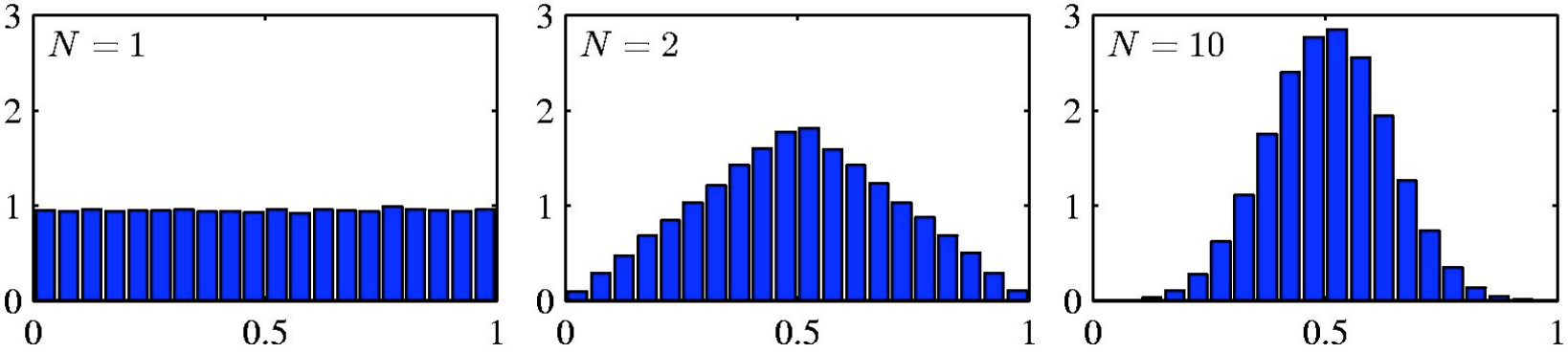
$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

MATLAB Statistics Toolbox Function:
normpdf(x,mu,sigma)

Central Limit Theorem

- The distribution of the sum of N i.i.d. random variables becomes increasingly Gaussian as N grows.
- Example: N uniform [0,1] random variables.



Expectations

$$\mathbb{E}[f] = \sum_x p(x) f(x)$$

$$\mathbb{E}[f] = \int p(x) f(x) \, \mathrm{d}x$$

$$\mathbb{E}_x[f|y] = \sum_x p(x|y) f(x)$$

Conditional Expectation (discrete)

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

Approximate Expectation (discrete and continuous)

Variances and Covariances

$$\text{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

$$\begin{aligned}\text{cov}[x, y] &= \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y]\end{aligned}$$

$$\begin{aligned}\text{cov}[\mathbf{x}, \mathbf{y}] &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^{\mathrm{T}} - \mathbb{E}[\mathbf{y}^{\mathrm{T}}]\}\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^{\mathrm{T}}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^{\mathrm{T}}]\end{aligned}$$

Gaussian Mean and Variance

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) x \, \mathrm{d}x = \mu$$

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) x^2 \, \mathrm{d}x = \mu^2 + \sigma^2$$

$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. **Bayesian Classifiers**
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. What are Bayes Nets?

Topic 3. Bayesian Classifiers

Bayesian Classification

- Input feature vectors

$$\mathbf{x} = \left[x_1, x_2, \ldots, x_l \right]^T$$

- Assign the pattern represented by feature vector **x** to the **most probable** of the available classes

$$\omega_1, \omega_2, \ldots, \omega_M$$

That is, $\mathbf{x} \rightarrow \omega_i : P(\omega_i \mid \mathbf{x})$ is maximum.

Posterior

Bayesian Classification

- Computation of posterior probabilities
  - Assume known
    - Prior probabilities

$$P(\omega_1), P(\omega_2) \ldots, P(\omega_M)$$

    - Likelihoods

$$p(\mathbf{x} \mid \omega_i), \qquad i = 1,2,\ldots,M$$

Bayes' Rule for Classification

$$p(\omega_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_i) p(\omega_i)}{p(\mathbf{x})},$$

where

$$p(\mathbf{x}) = \sum_{i=1}^{M} p(\mathbf{x} \mid \omega_i) p(\omega_i)$$

M=2 Classes

- Given **x** classify it according to the rule

$$\text{If } P(\omega_1|\mathbf{x}) > P(\omega_2|\mathbf{x}) \rightarrow \omega_1$$
$$\text{If } P(\omega_2|\mathbf{x}) > P(\omega_1|\mathbf{x}) \rightarrow \omega_2$$

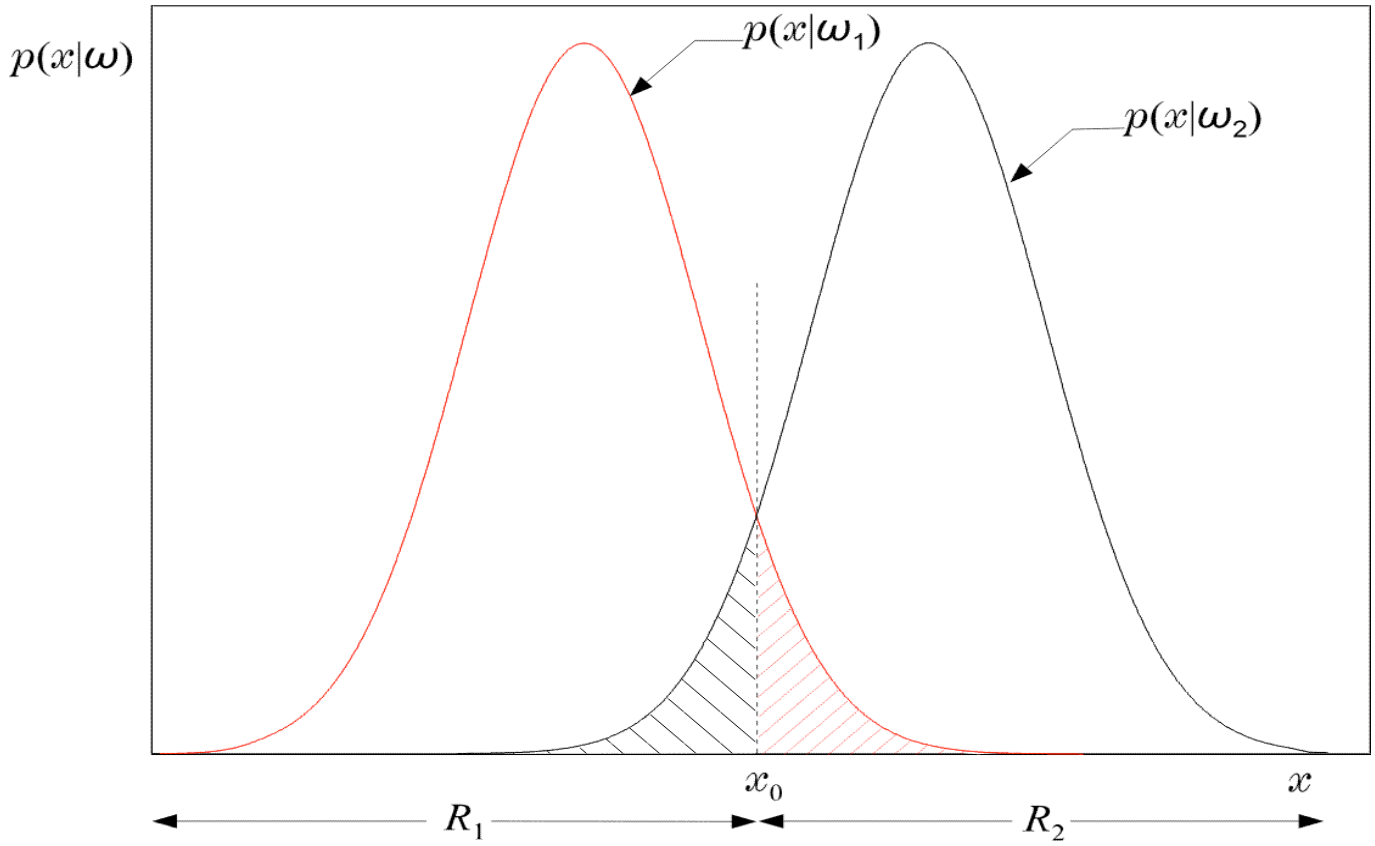
- Equivalently: classify **x** according to the rule

$$\text{If } p(\mathbf{x}|\omega_1)P(\omega_1) > p(\mathbf{x}|\omega_2)P(\omega_2) \rightarrow \omega_1$$
$$\text{If } p(\mathbf{x}|\omega_2)P(\omega_2) > p(\mathbf{x}|\omega_1)P(\omega_1) \rightarrow \omega_2$$

- For equiprobable classes the test becomes

$$\text{If } p(\mathbf{x}|\omega_1) > p(\mathbf{x}|\omega_2) \rightarrow \omega_1$$
$$\text{If } p(\mathbf{x}|\omega_2) > p(\mathbf{x}|\omega_1) \rightarrow \omega_2$$

Example: Equiprobable Classes



$$R_1(\rightarrow \omega_1) \text{ and } R_2(\rightarrow \omega_2)$$

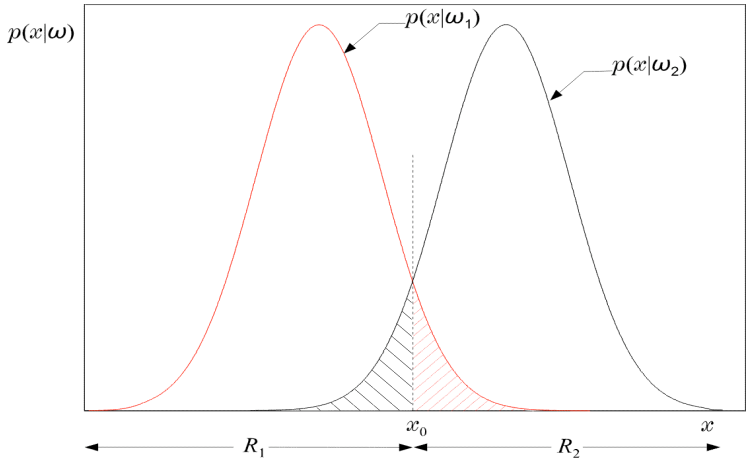
Example: Equiprobable Classes

- Probability of error
  - The black and red shaded areas represent

$$P(\text{error} \mid \omega_2) = \int_{-\infty}^{x_0} p(x|\omega_2)dx \quad \text{and} \quad P(\text{error} \mid \omega_1) = \int_{x_0}^{\infty} p(x|\omega_1)dx$$

  - Thus

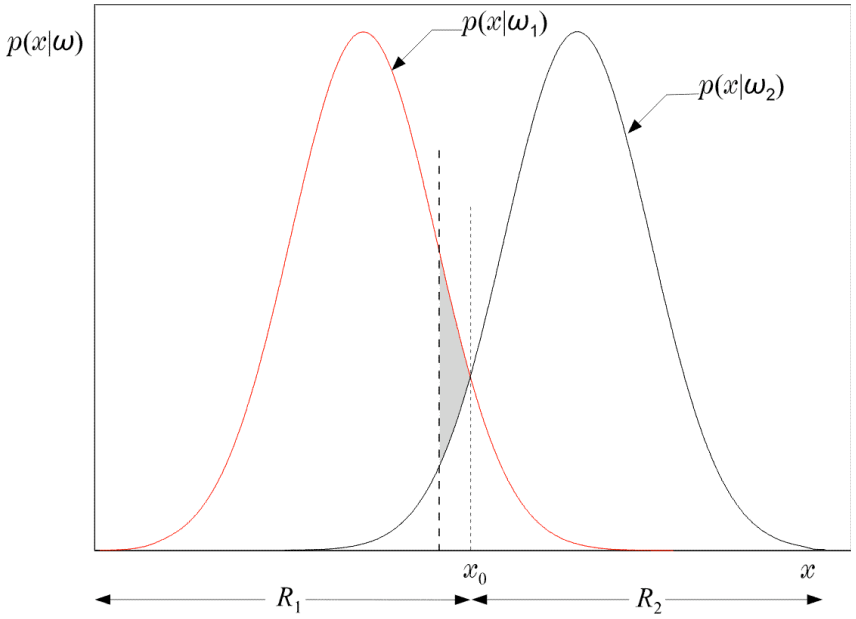
$$\begin{aligned} P_e &\triangleq P(\text{error}) \\ &= P(\omega_2)P(\text{error}|\omega_2) + P(\omega_1)P(\text{error}|\omega_1) \\ &= \frac{1}{2}\int_{-\infty}^{x_0} p(x|\omega_2)dx + \frac{1}{2}\int_{x_0}^{+\infty} p(x|\omega_1)dx \end{aligned}$$



- **Bayesian classifier is OPTIMAL: it minimizes the classification error probability**

Example: Equiprobable Classes

- To see this, observe that shifting the threshold increases the error rate for one class of patterns more than it decreases the error rate for the other class.



The General Case

- In general, for M classes and unequal priors, the decision rule

$$P(\omega_i \mid \mathbf{x}) > P(\omega_j \mid \mathbf{x}) \quad \forall j \neq i \quad \rightarrow \omega_i$$

minimizes the expected error rate.

Types of Error

- Minimizing the expected error rate is a pretty reasonable goal.
- However, it is not always the best thing to do.
- Example:
  - You are designing a pedestrian detection algorithm for an autonomous navigation system.
  - Your algorithm must decide whether there is a pedestrian crossing the street.
  - There are two possible types of error:
    - False positive: there is no pedestrian, but the system thinks there is.
    - Miss: there is a pedestrian, but the system thinks there is not.
  - Should you give equal weight to these 2 types of error?

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. **Minimizing Risk**
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. What are Bayes Nets?

Topic 4. Minimizing Risk

The Loss Matrix

- To deal with this problem, instead of minimizing error rate, we minimize something called the **risk**.
- First, we define the **loss matrix L**, which quantifies the cost of making each type of error.
- Element λ_{ij} of the loss matrix specifies the cost of deciding class *j* when in fact the input is of class *i*.
- Typically, we set $\lambda_{ii}=0$ for all i.
- Thus a typical loss matrix for the M = 2 case would have the form

$$L = \begin{bmatrix} 0 & \lambda_{12} \\ \lambda_{21} & 0 \end{bmatrix}$$

Risk

- Given a loss function, we can now define the risk associated with each class k as:

$$r_k = \sum_{i=1}^{M} \lambda_{ki} \int_{R_i} p(\mathbf{x} \mid \omega_k) d\mathbf{x}$$

Probability we will decide Class ω_i given pattern from Class ω_k

- where R_i is the region of the input space where we will decide ω_i.

Minimizing Risk

- Now the goal is to minimize the expected risk ***r***, given by

$$r = \sum_{k=1}^{M} r_k P(\omega_k)$$

Minimizing Risk

$$r = \sum_{k=1}^{M} r_k P(\omega_k) \quad \text{where} \quad r_k = \sum_{i=1}^{M} \lambda_{ki} \int_{R_i} p(\mathbf{x} \mid \omega_k) d\mathbf{x}$$

- We need to select the decision regions R_i to minimize the risk r.
- Note that the set of R_i are disjoint and exhaustive.
- Thus we can minimize the risk by ensuring that each input $\mathbf{x}$ falls in the region R_i that minimizes the expected loss for that particular input, i.e.,

Letting $l_i = \sum_{k=1}^{M} \lambda_{ki} p(\mathbf{x} \mid \omega_k) P(\omega_k)$,

we select the partioning regions such that

$\mathbf{x} \in R_i$ if $l_i < l_j \quad \forall j \neq i$

Example: M=2

- For the 2-class case:

$$l_1 = \lambda_{11} p(\mathbf{x} \mid \omega_1) P(\omega_1) + \lambda_{21} p(\mathbf{x} \mid \omega_2) P(\omega_2)$$

and

$$l_2 = \lambda_{12} p(\mathbf{x} \mid \omega_1) P(\omega_1) + \lambda_{22} p(\mathbf{x} \mid \omega_2) P(\omega_2)$$

- Thus we assign $\mathbf{x}$ to ω_1 if

$$(\lambda_{21} - \lambda_{22}) p(\mathbf{x} \mid \omega_2) P(\omega_2) < (\lambda_{12} - \lambda_{11}) p(\mathbf{x} \mid \omega_1) P(\omega_1)$$

- i.e., if

Likelihood Ratio Test

$$\frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} > \frac{P(\omega_2)(\lambda_{21} - \lambda_{22})}{P(\omega_1)(\lambda_{12} - \lambda_{11})}.$$

Likelihood Ratio Test

$$\frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} \overset{?}{>} \frac{P(\omega_2)(\lambda_{21} - \lambda_{22})}{P(\omega_1)(\lambda_{12} - \lambda_{11})}.$$

- Typically, the loss for a correct decision is 0. Thus the likelihood ratio test becomes

$$\frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} \overset{?}{>} \frac{P(\omega_2)\lambda_{21}}{P(\omega_1)\lambda_{12}}.$$

- In the case of equal priors and equal loss functions, the test reduces to

$$\frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} \overset{?}{>} 1.$$

Example

- Consider a one-dimensional input space, with features generated by normal distributions with identical variance:

$$p(x|\omega_1) \sim N(\mu_1, \sigma^2)$$
$$p(x|\omega_2) \sim N(\mu_2, \sigma^2)$$

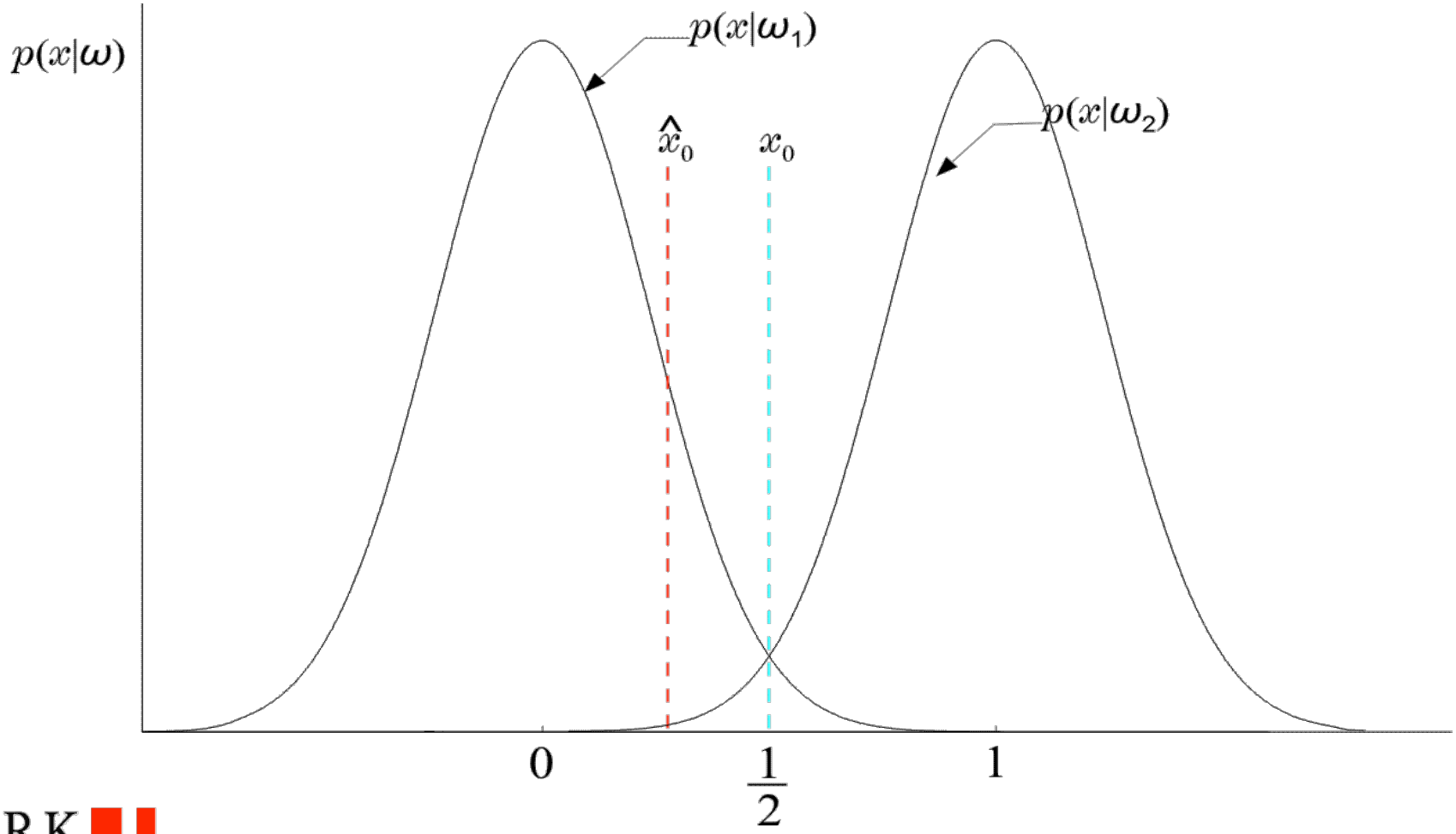
where $\mu_1 = 0, \mu_2 = 1,$ and $\sigma^2 = \frac{1}{2}$

- Let's assume equiprobable classes, and higher loss for errors on Class 2, specifically:

$$\lambda_{21} = 1, \ \lambda_{12} = \frac{1}{2}.$$

Results

- The threshold has shifted to the left – why?

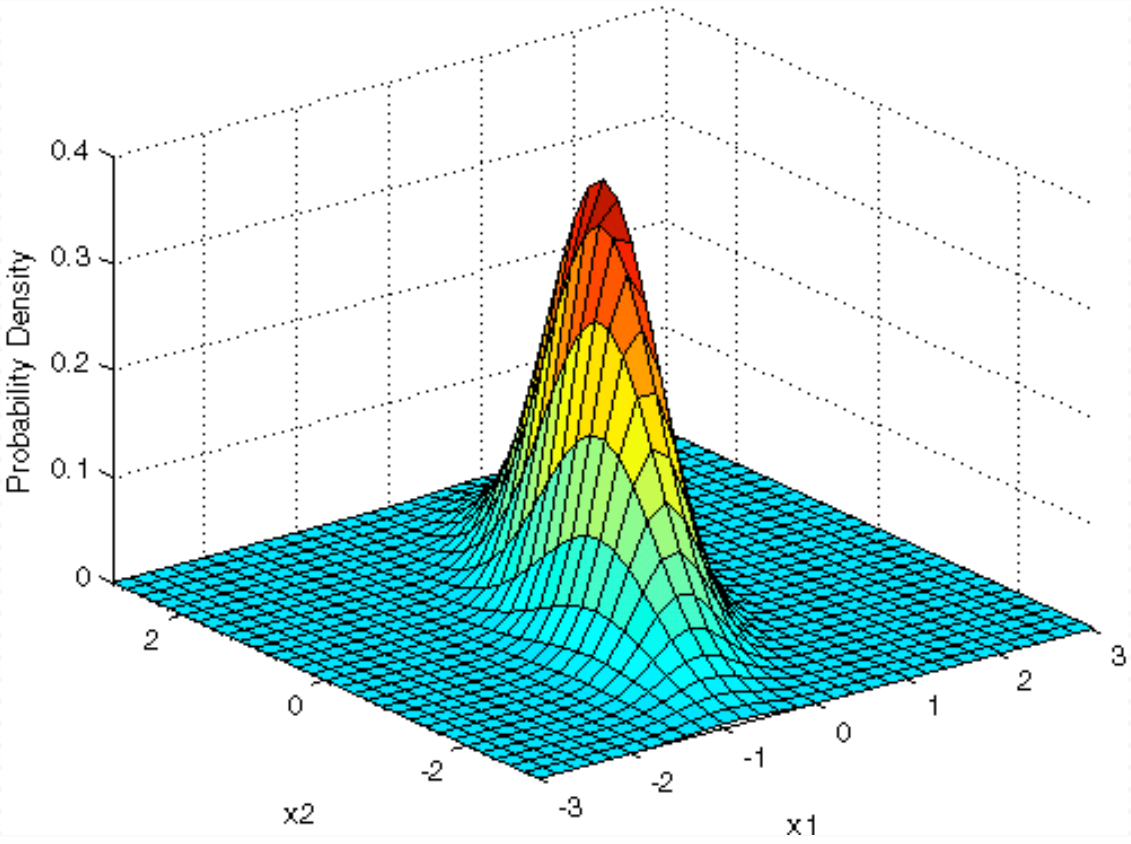


End of Lecture 3

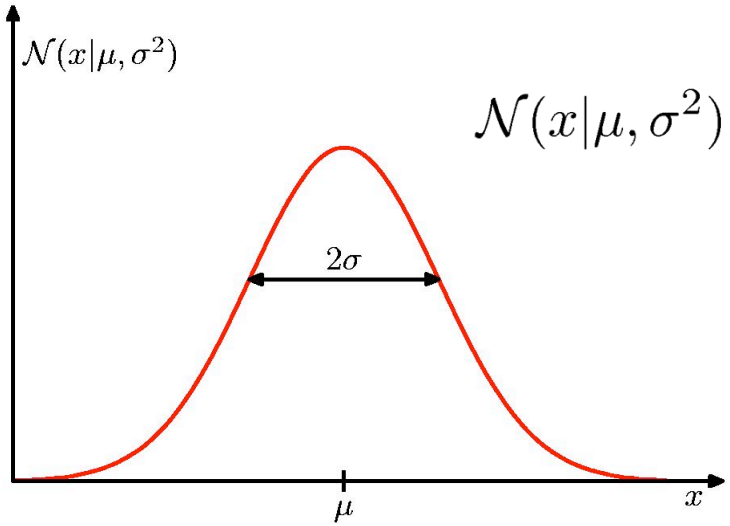
Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. **The Multivariate Normal Distribution**
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. What are Bayes Nets?

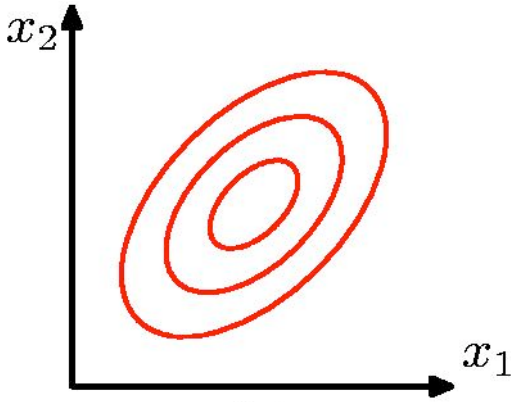
Topic 5 The Multivariate Normal Distribution



The Multivariate Gaussian



$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



MATLAB Statistics Toolbox Function:
mvnpdf(x,mu,sigma)

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Geometry of the Multivariate Gaussian

$$\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$ where $\Delta \equiv$ Mahalanobis distance from μ to x

MATLAB Statistics Toolbox Function: mahal(x,y)

Eigenvector equation: $\Sigma u_i = \lambda_i u_i$

where $(\mathbf{u}_i, \lambda_i)$ are the *i*th eigenvector and eigenvalue of Σ.

Note that Σ real and symmetric $\rightarrow$ λ_i real.

See Appendix B for a review of matrices and eigenvectors.

Geometry of the Multivariate Gaussian

$$\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

Δ = Mahalanobis distance from μ to x

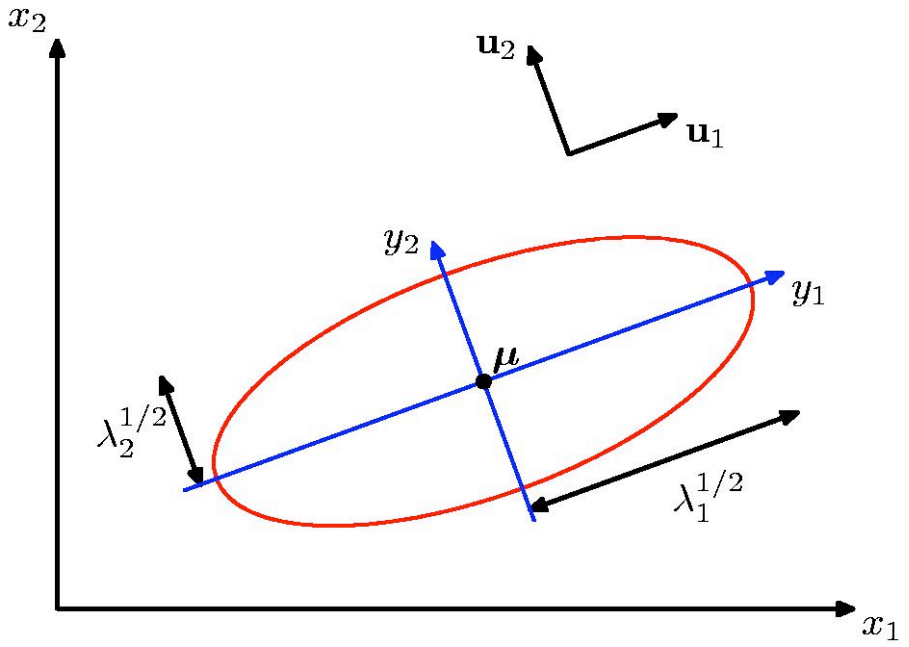
$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{D}\frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}$$

where $(\mathbf{u}_i, \lambda_i)$ are the ith eigenvector and eigenvalue of Σ.

$$\Delta^2 = \sum_{i=1}^{D}\frac{y_i^2}{\lambda_i}$$

$$y_i = \mathbf{u}_i^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu})$$

or $\mathbf{y} = \mathbf{U}(\mathbf{x} - \boldsymbol{\mu})$



Moments of the Multivariate Gaussian

$$\begin{aligned}\mathbb{E}[\mathbf{x}] &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}\mathbf{x}\,\mathrm{d}\mathbf{x} \\ &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}\mathbf{z}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{z}\right\}(\mathbf{z}+\boldsymbol{\mu})\,\mathrm{d}\mathbf{z}\end{aligned}$$

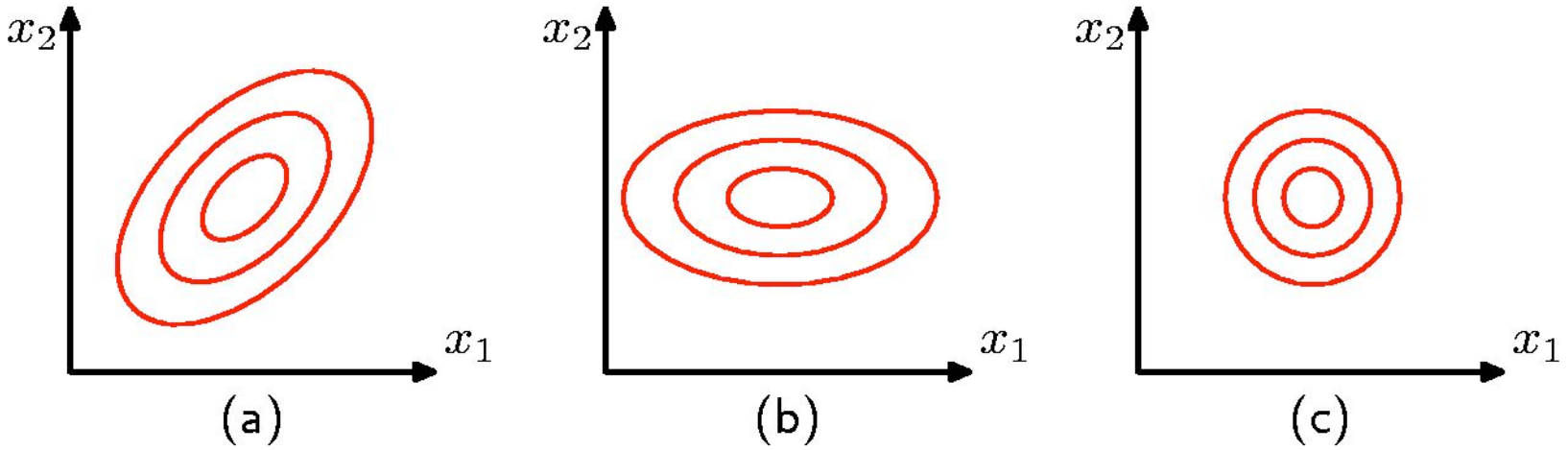
thanks to anti-symmetry of z

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$$

Moments of the Multivariate Gaussian

$$\mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \boldsymbol{\Sigma}$$

$$\mathrm{cov}[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^{\mathrm{T}}\right] = \boldsymbol{\Sigma}$$

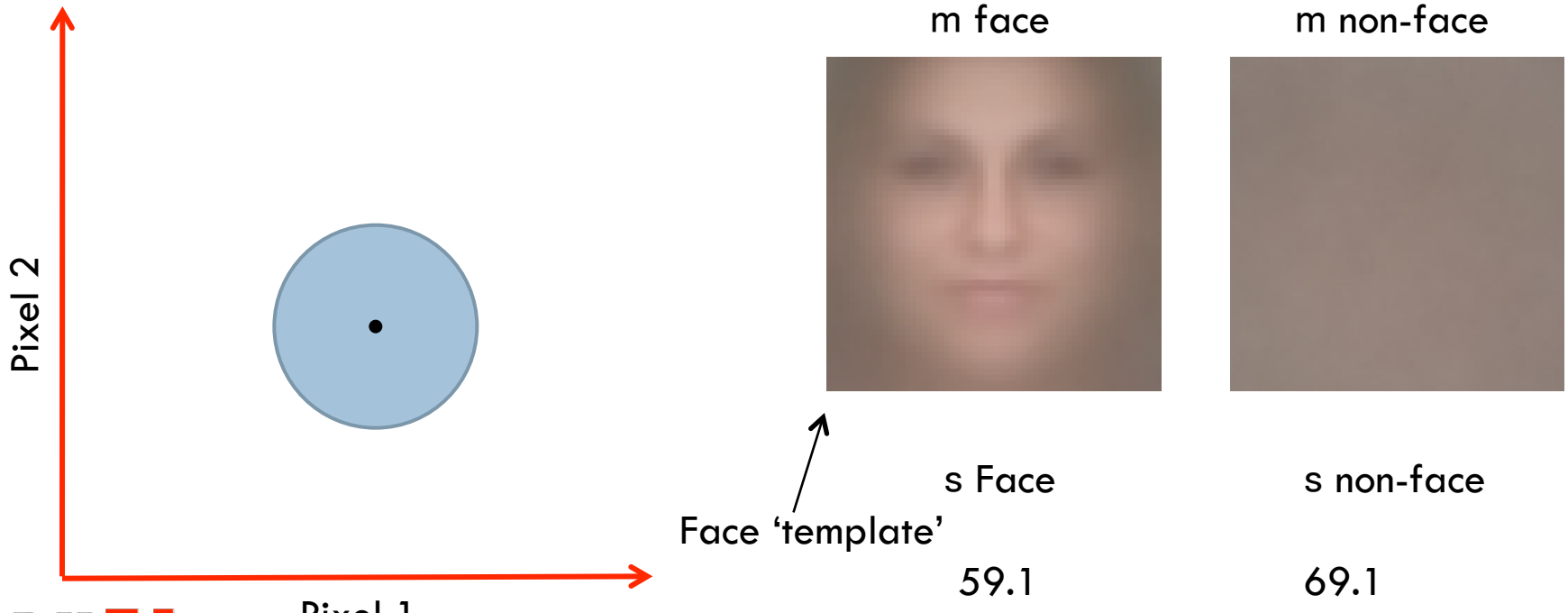


5.1 Application: Face Detection

Model # 1: Gaussian, uniform covariance

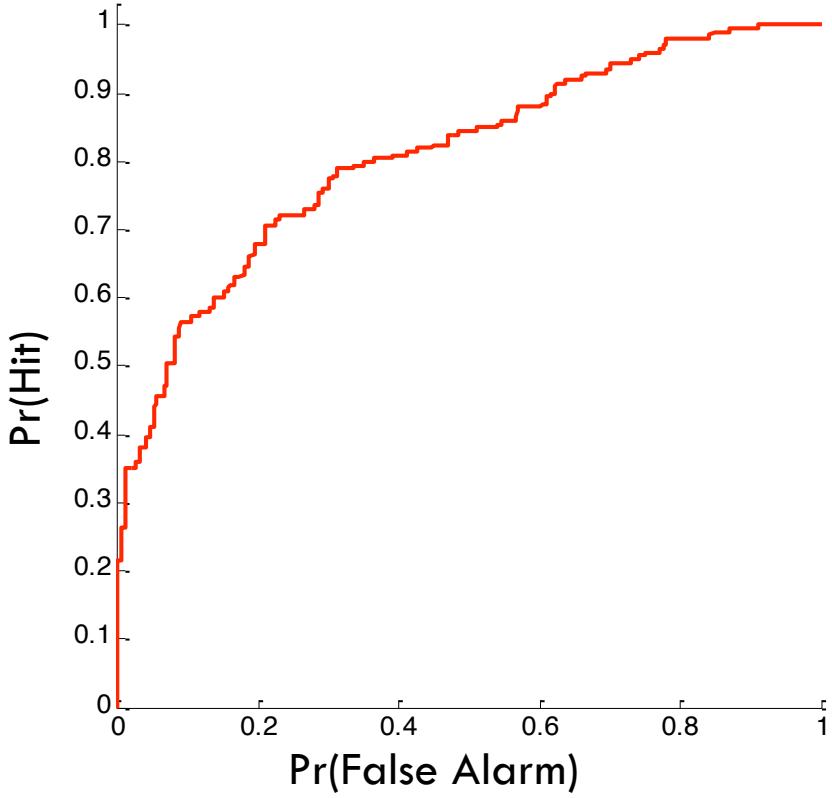
$$Pr(\mathbf{x}|\text{face}) = \frac{1}{(2\pi)^{d/2}|\Sigma|}\exp\left\{-0.5(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

Fit model using maximum likelihood criterion



Model 1 Results

Results based on 200 cropped faces and 200 non-faces from the same database.

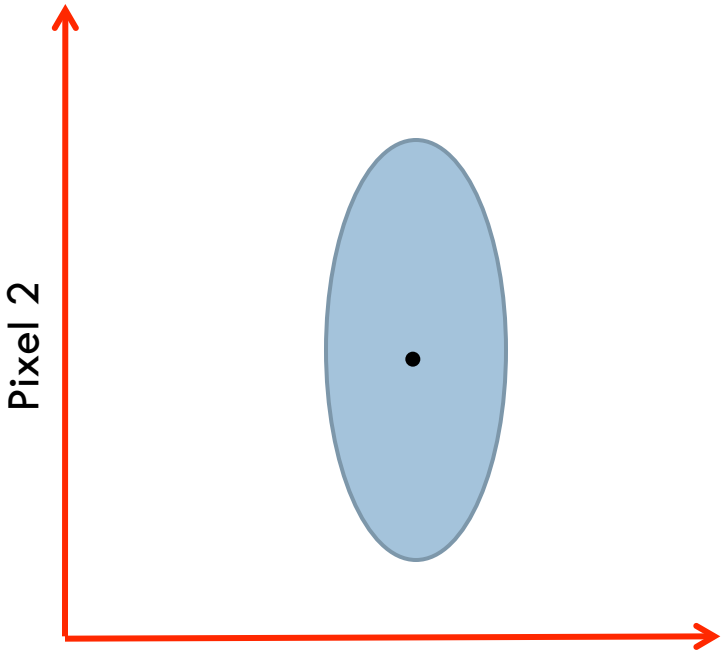


How does this work with a real image?

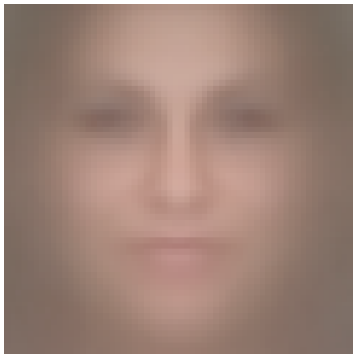
Model # 2: Gaussian, diagonal covariance

$$Pr(\mathbf{x}|\text{face}) = \frac{1}{(2\pi)^{d/2}|\Sigma|}\exp\left\{-0.5(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

Fit model using maximum likelihood criterion

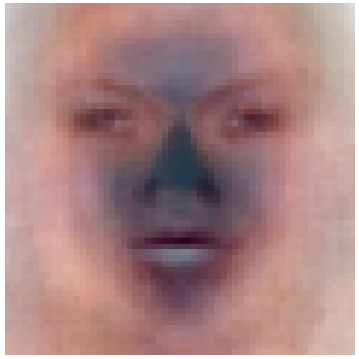


m face

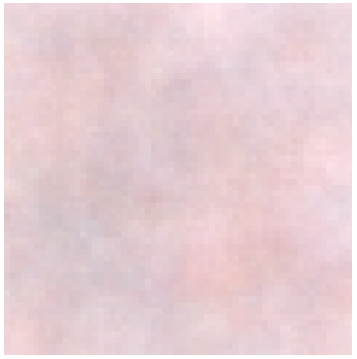


m non-face

s Face

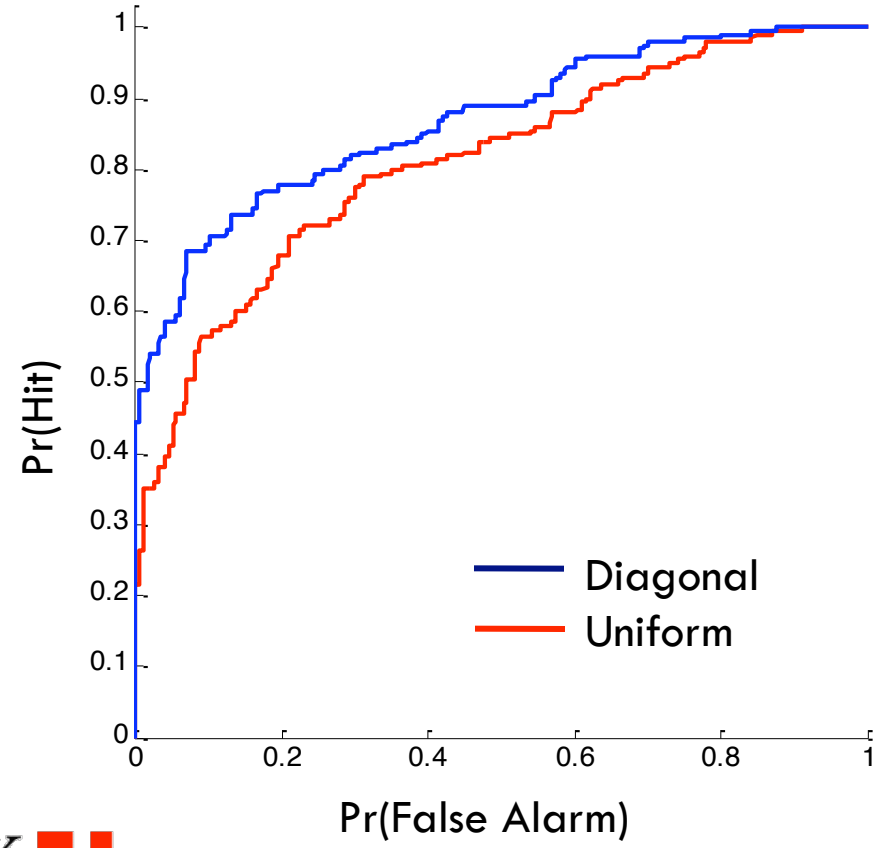


s non-face



Model 2 Results

Results based on 200 cropped faces and 200 non-faces from the same database.

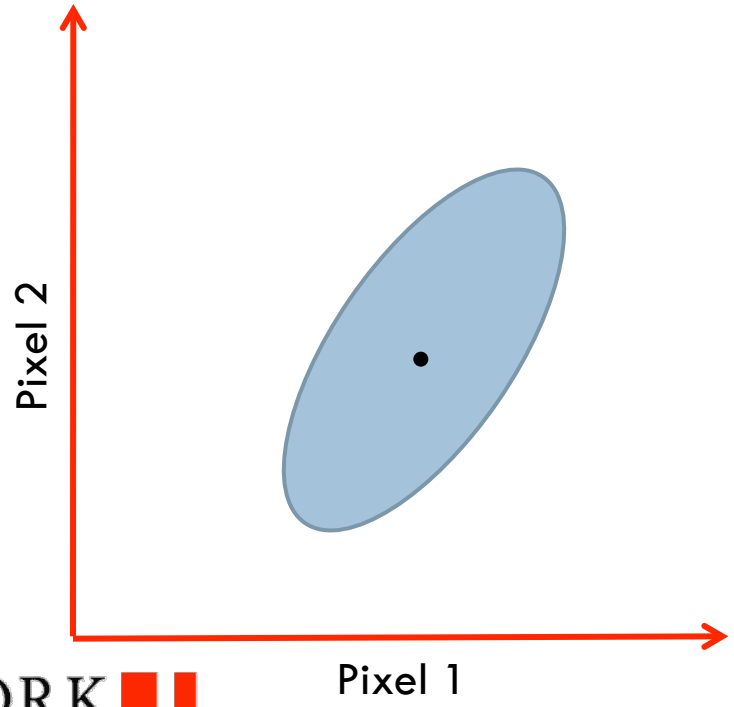


More sophisticated model unsurprisingly classifies new faces and non-faces better.

Model # 3: Gaussian, full covariance

$$Pr(\mathbf{x}|\text{face}) = \frac{1}{(2\pi)^{d/2}|\Sigma|}\exp\left\{-0.5(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

Fit model using maximum likelihood criterion



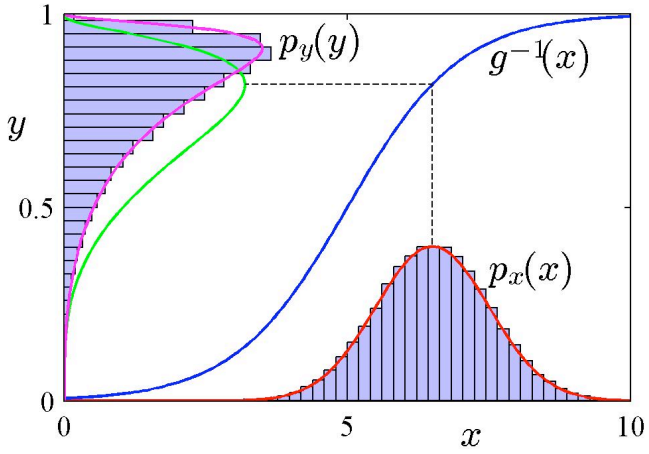
PROBLEM: we cannot fit this model. We don't have enough data to estimate the full covariance matrix.

N=800 training images
D=10800 dimensions

Total number of measured numbers = ND = 800x10,800 = 8,640,000

Total number of parameters in cov matrix = (D+1)D/2 = (10,800+1)x10,800/2 = 58,325,400

Transformed Densities Revisited



Observations falling within $(x + \delta x)$ tranform to the range $(y + \delta y)$

$$\rightarrow p_x(x)|\delta x| = p_y(y)|\delta y|$$

$$\rightarrow p_y(y) \simeq p_x(x)\left|\frac{\delta x}{\delta y}\right|$$

Note that in general, $\delta y \neq \delta x$.

Rather, $\frac{\delta y}{\delta x} \rightarrow \frac{dy}{dx}$ as $\delta x \rightarrow 0$.

Thus $p_y(y) = p_x(x)\left|\frac{dx}{dy}\right|$

Problems for this week

- Problems 2.7 – 2.17, 2.19 – 2.21, 2.23 – 2.27 are all good!
  - At least do problem 2.14. We will talk about this Monday. (Hopefully one of you will present a solution!)
- Also, MATLAB exercises up to 1.4.4 are good.
  - At least do Exercise 1.4.2. We will talk about this next week.

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. **Decision Boundaries in Higher Dimensions**
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. What are Bayes Nets?

Topic 6.
Decision Boundaries in Higher Dimensions

Decision Surfaces

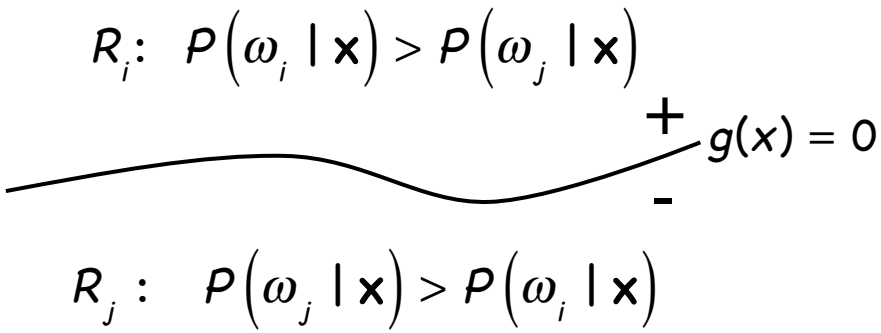
- If decision regions R_i and R_j are contiguous, define

$$g(\mathbf{x}) \equiv P(\omega_i \mid \mathbf{x}) - P(\omega_j \mid \mathbf{x})$$

- Then the decision surface

$$g(x) = 0$$

separates the two decision regions. $g(x)$ is positive on one side and negative on the other.



Discriminant Functions

- If $f(.)$ monotonic, the rule remains the same if we use:

$$\underline{x} \rightarrow \omega_i \text{ if: } f(P(\omega_i | \underline{x})) > f(P(\omega_j | \underline{x})) \quad \forall i \neq j$$

- $g_i(\mathbf{x}) \equiv f(P(\omega_i | \mathbf{x}))$ is a **discriminant function**
- In general, discriminant functions can be defined in other ways, independent of Bayes.
- In theory this will lead to a suboptimal solution
- However, non-Bayesian classifiers can have significant advantages:
  - Often a full Bayesian treatment is intractable or computationally prohibitive.
  - Approximations made in a Bayesian treatment may lead to errors avoided by non-Bayesian methods.

End of Lecture 4

Multivariate Normal Likelihoods

- Multivariate Gaussian pdf

$$p(\underline{x}|\omega_i) = \frac{1}{(2\pi)^{\frac{\ell}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma_i^{-1}(\underline{x}-\underline{\mu}_i)\right)$$

$$\underline{\mu}_i = E\left[\underline{x}\right]$$

$$\Sigma_i = E\left[(\underline{x}-\underline{\mu}_i)(\underline{x}-\underline{\mu}_i)^T\right]$$

called the covariance matrix

Logarithmic Discriminant Function

$$p(\underline{x}|\omega_i) = \frac{1}{(2\pi)^{\frac{\ell}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma_i^{-1}(\underline{x}-\underline{\mu}_i)\right)$$

- ln(·) is monotonic. Define:

$$g_i(\underline{x}) = \ln\left(p\left(\underline{x}\mid\omega_i\right)P\left(\omega_i\right)\right) = \ln p\left(\underline{x}\mid\omega_i\right) + \ln P(\omega_i)$$

$$= -\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma_i^{-1}(\underline{x}-\underline{\mu}_i) + \ln P(\omega_i) + C_i$$

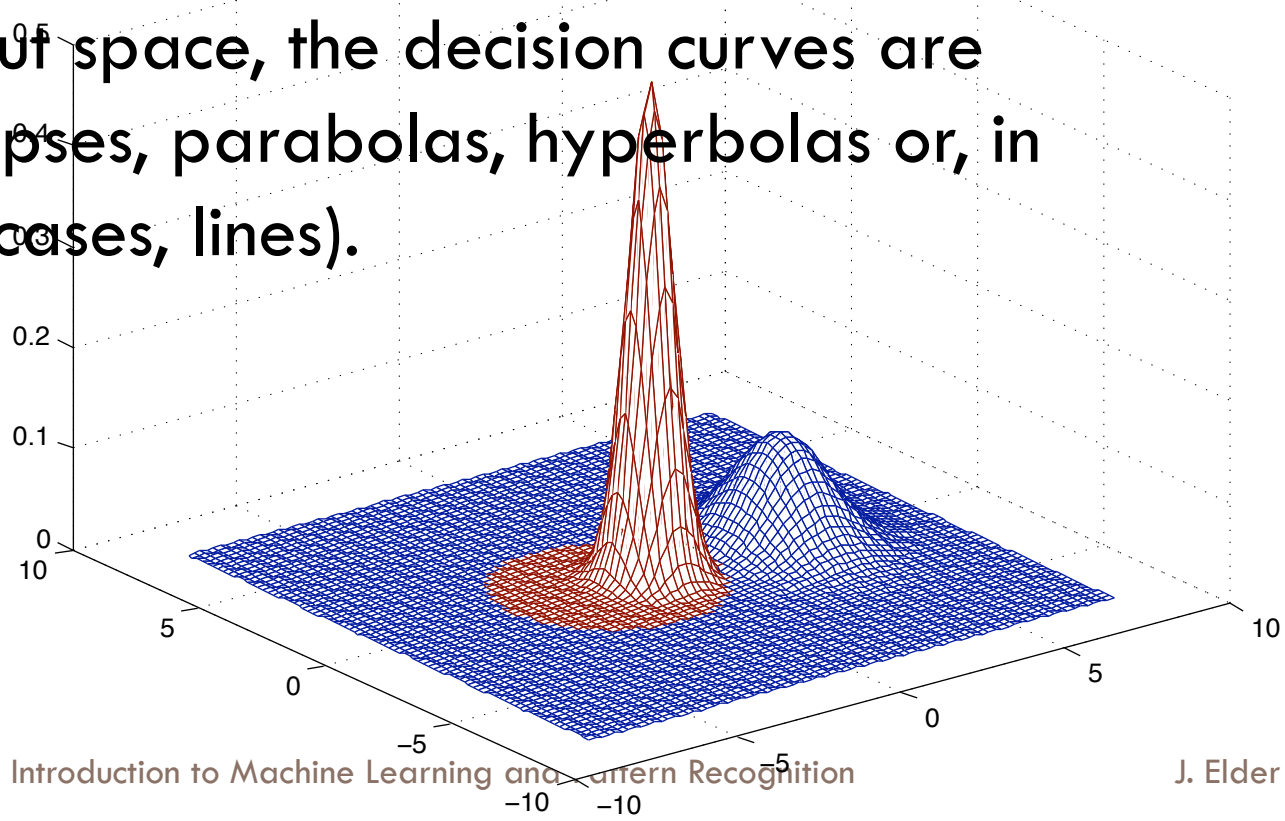
where

$$C_i = -\frac{\ell}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i|$$

Quadratic Classifiers

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) + \ln P(\omega_i) + C_i$$

- Thus the decision surface has a quadratic form.
- For a 2D input space, the decision curves are quadrics (ellipses, parabolas, hyperbolas or, in degenerate cases, lines).



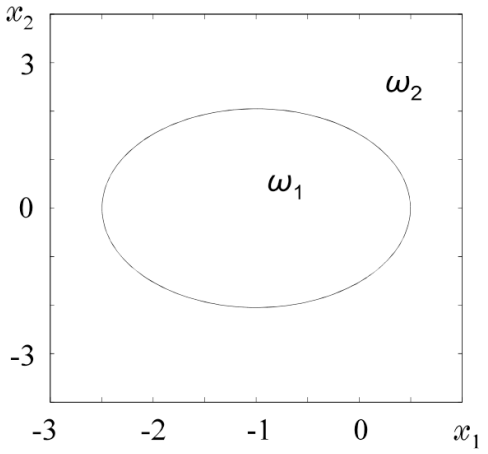
Example: Isotropic Likelihoods

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) + \ln P(\omega_i) + C_i$$

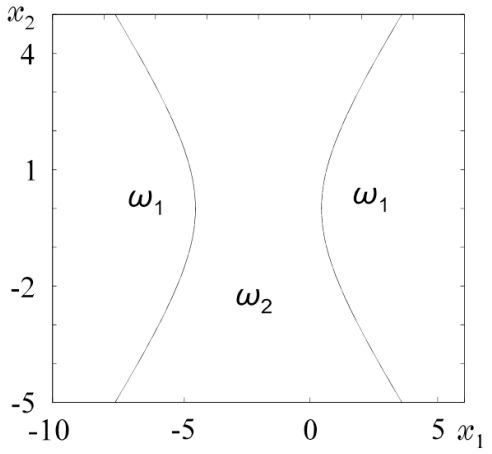
- Suppose that the two likelihoods are both isotropic, but with different means and variances. Then

$$g_i(\mathbf{x}) = -\frac{1}{2\sigma_i^2}(x_1^2 + x_2^2) + \frac{1}{\sigma_i^2}(\mu_{i1}x_1 + \mu_{i2}x_2) - \frac{1}{2\sigma_i^2}(\mu_{i1}^2 + \mu_{i2}^2) + \ln\left(P\left(\omega_i\right)\right) + C_i$$

- And $g_i(\underline{x}) - g_j(\underline{x}) = 0$ will be a quadratic equation in 2 variables.



(a)



(b)

Equal Covariances

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) + \ln P(\omega_i) + C_i$$

- The quadratic term of the decision boundary is given by

$$\frac{1}{2}\mathbf{x}^T\left(\Sigma_j^{-1} - \Sigma_i^{-1}\right)\mathbf{x}$$

- Thus if the covariance matrices of the two likelihoods are identical, the decision boundary is linear.

Linear Classifier

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma^{-1}(\underline{x} - \underline{\mu}_i) + \ln P(\omega_i) + C_i$$

- In this case, we can drop the quadratic terms and express the discriminant function in linear form:

$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{io}$$

$$\underline{w}_i = \Sigma^{-1}\underline{\mu}_i$$

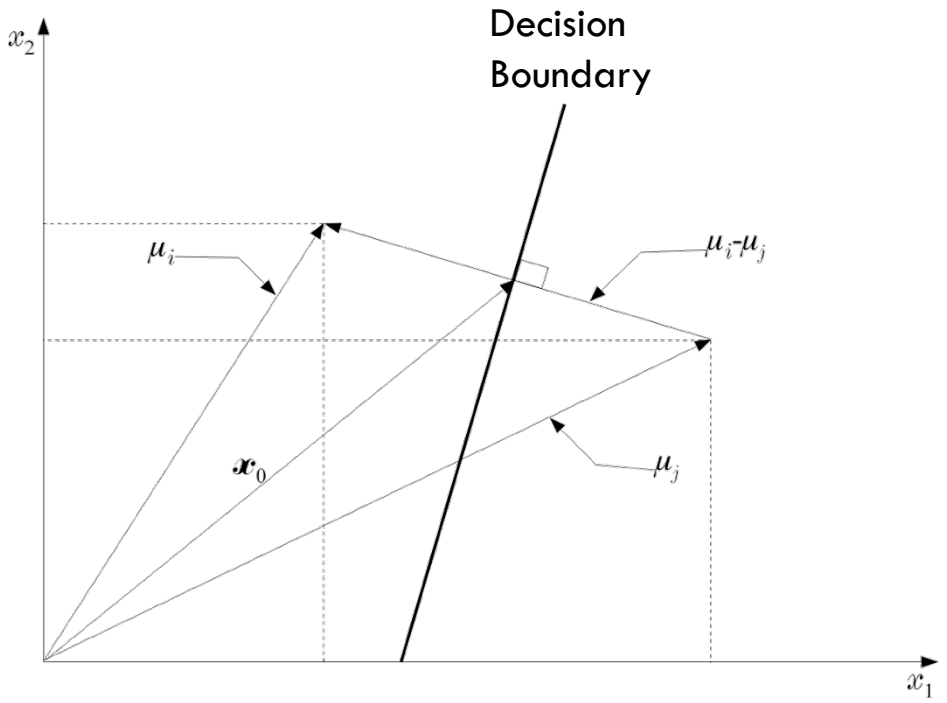
$$w_{i0} = \ln P(\omega_i) - \frac{1}{2}\underline{\mu}_i^T \Sigma^{-1}\underline{\mu}_i$$

Example 1: Isotropic, Identical Variance

$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{io}$$

$$\underline{w}_i = \Sigma^{-1}\underline{\mu}_i$$

$$w_{i0} = \ln P(\omega_i) - \frac{1}{2}\underline{\mu}_i^T\Sigma^{-1}\underline{\mu}_i$$



$\Sigma = \sigma^2 I$. Then

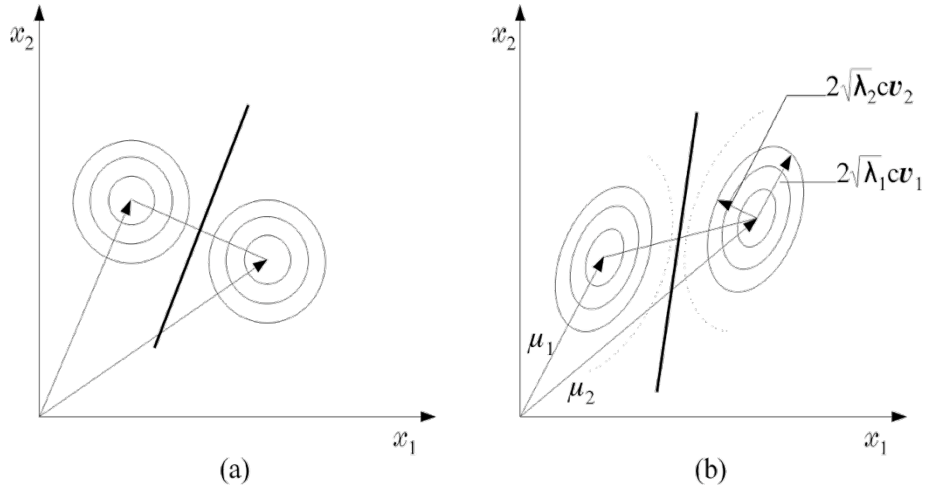
$\underline{w}^T(\underline{x} - \underline{x}_o) = 0$, where

$\underline{w} = \underline{\mu}_i - \underline{\mu}_j$, and

$$\underline{x}_o = \frac{1}{2}(\underline{\mu}_i + \underline{\mu}_j) - \sigma^2 \ln\frac{P(\omega_i)}{P(\omega_j)}\frac{\underline{\mu}_i - \underline{\mu}_j}{\left\|\underline{\mu}_i - \underline{\mu}_j\right\|^2}$$

Example 2: Equal Covariance

$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{io}$$
$$\underline{w}_i = \Sigma^{-1}\underline{\mu}_i$$
$$w_{i0} = \ln P(\omega_i) - \frac{1}{2}\underline{\mu}_i^T\Sigma^{-1}\underline{\mu}_i$$



$$g_{ij}(\underline{x}) = \underline{w}^T(\underline{x} - \underline{x}_0) = 0 \quad \text{where}$$

$$\underline{w} = \Sigma^{-1}(\underline{\mu}_i - \underline{\mu}_j),$$

$$\underline{x}_0 = \frac{1}{2}(\underline{\mu}_i + \underline{\mu}_j) - \ln\left(\frac{P(\omega_i)}{P(\omega_j)}\right)\frac{\underline{\mu}_i - \underline{\mu}_j}{\left\|\underline{\mu}_i - \underline{\mu}_j\right\|^2_{\Sigma^{-1}}},$$

and

$$\left\|\underline{x}\right\|_{\Sigma^{-1}} \equiv (\underline{x}^T\Sigma^{-1}\underline{x})^{\frac{1}{2}}$$

Minimum Distance Classifiers

- If the two likelihoods have identical covariance AND the two classes are equiprobable, the discrimination function simplifies:

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) + \ln P(\omega_i) + C_i$$

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma^{-1}(\underline{x} - \underline{\mu}_i)$$

Isotropic Case

- In the isotropic case,

$$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma^{-1}(\underline{x} - \underline{\mu}_i) = -\frac{1}{2\sigma^2}\left\|\underline{x} - \underline{\mu}_i\right\|^2$$

- Thus the Bayesian classifier simply assigns the class that minimizes the Euclidean distance d_e between the observed feature vector and the class mean.

$$d_e = \left\|\underline{x} - \underline{\mu}_i\right\|$$

General Case: Mahalanobis Distance

- To deal with anisotropic distributions, we simply classify according to the Mahalanobis distance, defined as

$$d_m = g_i(\underline{x}) = \left((\underline{x} - \underline{\mu}_i)^T \Sigma^{-1} (\underline{x} - \underline{\mu}_i)\right)^{1/2}$$

- Since the covariance matrix is symmetric, it can be represented as $\Sigma = \Phi \Lambda \Phi^T$

  where the columns v_i of Φ are the eigenvectors of Σ

  and where

  Λ is a diagonal matrix whose diagonal elements λ_i

  are the corresponding eigenvalues.

- Then we have

$$d_m^2 = (\underline{x} - \underline{\mu}_i)^T \Phi^T \Lambda^{-1} \Phi (\underline{x} - \underline{\mu}_i)$$

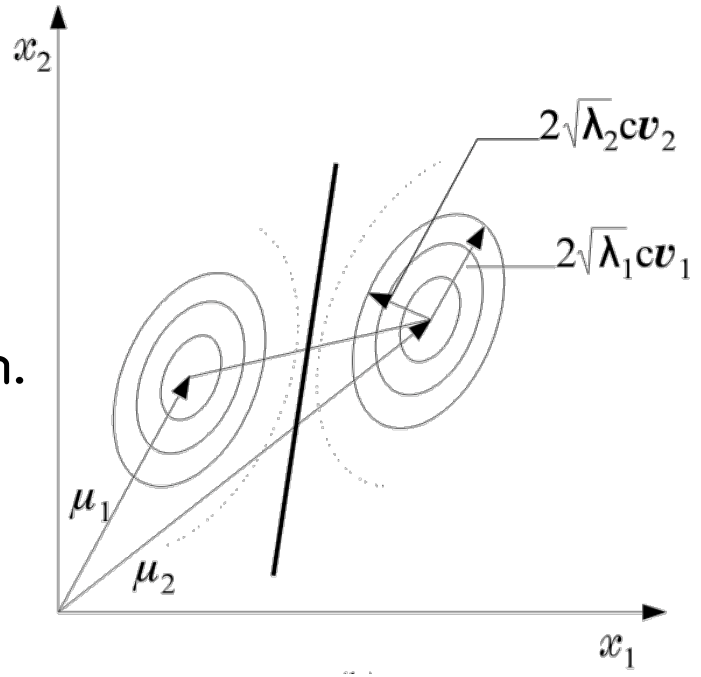
General Case: Mahalanobis Distance

$$d_m^2 = (\underline{x} - \underline{\mu}_i)^T \Phi^T \Lambda^{-1} \Phi (\underline{x} - \underline{\mu}_i)$$

Let $\mathbf{x}' = \Phi^T \mathbf{x}$. Then the coordinates of $\mathbf{x}'$ are the projections of $\mathbf{x}$ onto the eigenvectors of Σ, and we have:

$$\frac{(x_1' - \mu_{i1}')^2}{\lambda_1} + \cdots + \frac{(x_l' - \mu_{il}')^2}{\lambda_l} = d_m^2$$

Thus the curves of constant Mahalanobis distance c have ellipsoidal form.



Example:

Given ω_1, ω_2 : $P(\omega_1) = P(\omega_2)$ and $p(\underline{x}|\omega_1) = N(\underline{\mu}_1, \Sigma)$, $p(\underline{x}|\omega_2) = N(\underline{\mu}_2, \Sigma)$,

$$\underline{\mu}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \underline{\mu}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1.1 & 0.3 \\ 0.3 & 1.9 \end{bmatrix}$$

classify the vector $\underline{x} = \begin{bmatrix} 1.0 \\ 2.2 \end{bmatrix}$ using Bayesian classification:

- $\Sigma^{-1} = \begin{bmatrix} 0.95 & -0.15 \\ -0.15 & 0.55 \end{bmatrix}$
- Compute Mahalanobis d_m from μ_1, μ_2 :

$$d^2_{m,1} = \begin{bmatrix} 1.0, & 2.2 \end{bmatrix} \Sigma^{-1} \begin{bmatrix} 1.0 \\ 2.2 \end{bmatrix} = 2.952, \; d^2_{m,2} = \begin{bmatrix} -2.0, & -0.8 \end{bmatrix} \Sigma^{-1} \begin{bmatrix} -2.0 \\ -0.8 \end{bmatrix} = 3.672$$

- Classify $\underline{x} \rightarrow \omega_1$. Observe that $d_{E,2} < d_{E,1}$

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. **Parameter Estimation**
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. What are Bayes Nets?

Topic 7. Parameter Estimation

Topic 7.1 Maximum Likelihood Estimation

Maximum Likelihood Parameter Estimation

Suppose we believe input vectors $\underline{x}$ are distributed as $p(\underline{x}) \equiv p(\underline{x};\underline{\theta})$, where $\underline{\theta}$ is an unknown parameter.

Given independent training input vectors $X = \left\{\underline{x}_1, \underline{x}_2, \ldots \underline{x}_N\right\}$ we want to compute the maximum likelihood estimate $\underline{\theta}_{ML}$ for $\underline{\theta}$.

Since the input vectors are independent, we have

$$p(X;\underline{\theta}) \equiv p(\underline{x}_1, \underline{x}_2, \ldots \underline{x}_N;\underline{\theta}) = \prod_{k=1}^{N} p(\underline{x}_k;\underline{\theta})$$

Maximum Likelihood Parameter Estimation

$$p(X;\underline{\theta}) = \prod_{k=1}^{N} p(\underline{x}_k;\underline{\theta})$$

$$\text{Let } L(\underline{\theta}) \equiv \ln p(X;\underline{\theta}) = \sum_{k=1}^{N} \ln p(\underline{x}_k;\underline{\theta})$$

The general method is to take the derivative of L with respect to $\underline{\theta}$, set it to 0 and solve for $\underline{\theta}$:

$$\hat{\underline{\theta}}_{ML}: \quad \frac{\partial L(\underline{\theta})}{\partial(\underline{\theta})} = \sum_{k=1}^{N} \frac{\partial \ln p(\underline{x}_k;\underline{\theta})}{\partial(\underline{\theta})} = \underline{0}$$

Properties of the Maximum Likelihood Estimator

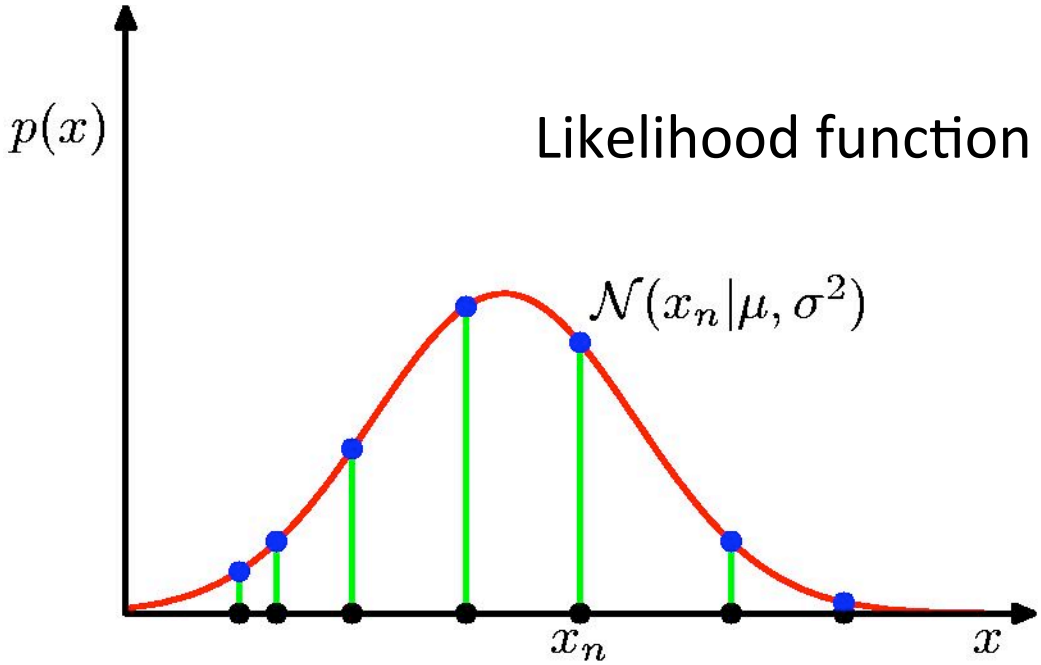
Let $\underline{\theta}_0$ be the true value of the unknown parameter vector.

Then

$\underline{\theta}_{ML}$ is asymptotically unbiased: $\lim_{N\to\infty} E[\underline{\theta}_{ML}] = \underline{\theta}_0$

$\underline{\theta}_{ML}$ is asymptotically consistent: $\lim_{N\to\infty} E\left\|\hat{\underline{\theta}}_{ML} - \underline{\theta}_0\right\|^2 = 0$

Example: Univariate Normal



$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu,\sigma^2\right)$$

Example: Univariate Normal

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

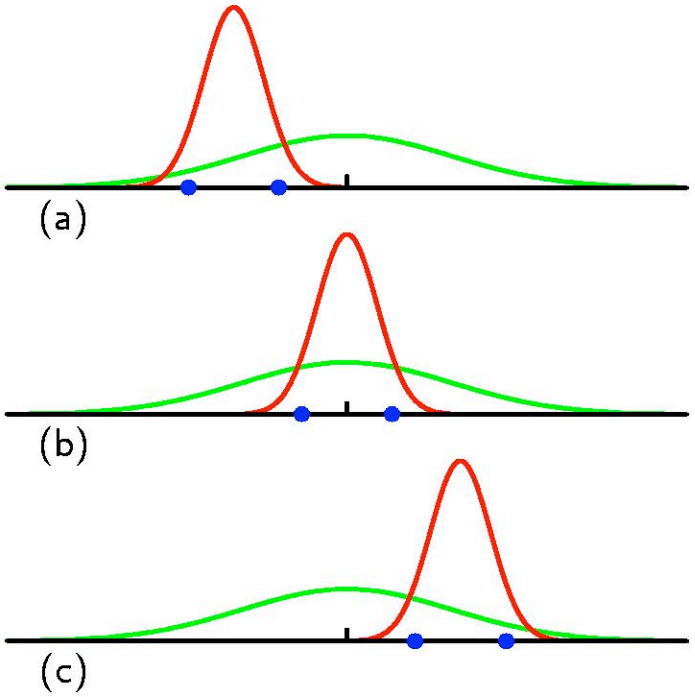
$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}x_n \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{\mathrm{ML}})^2$$

Example: Univariate Normal

$$\mathbb{E}[\mu_{\mathrm{ML}}] = \mu$$

$$\mathbb{E}[\sigma^2_{\mathrm{ML}}] = \left(\frac{N-1}{N}\right)\sigma^2$$

$$\begin{aligned}\widetilde{\sigma}^2 &= \frac{N}{N-1}\sigma^2_{\mathrm{ML}} \\ &= \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2\end{aligned}$$



Thus σ_{ML} is biased (although asymptotically unbiased).

End of Lecture 5

Example: Multivariate Normal

- Given i.i.d. data $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^{\mathrm{T}}$, the log likelihood function is given by

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$$

Maximum Likelihood for the Gaussian

- Set the derivative of the log likelihood function to zero,

$$\frac{\partial}{\partial \boldsymbol{\mu}} \ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}) = 0$$

- and solve to obtain

$$\boldsymbol{\mu}_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n.$$

- One can also show that

$$\boldsymbol{\Sigma}_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathrm{T}}.$$

Recall: If $\mathbf{x}$ and $\mathbf{a}$ are vectors, then $\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{x}^t\mathbf{a}\right) = \frac{\partial}{\partial \mathbf{x}}\left(\mathbf{a}^t\mathbf{x}\right) = \mathbf{a}$

Topic 7.1 Bayesian Parameter Estimation

Bayesian Inference for the Gaussian (Univariate Case)

- Assume σ^2 is known. Given i.i.d. data $\mathbf{x} = \{x_1, \ldots, x_N\}$, the likelihood function for μ is given by

$$p(\mathbf{x}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2\right\}.$$

- This has a Gaussian shape as a function of μ (but it is *not* a distribution over μ).

Bayesian Inference for the Gaussian (Univariate Case)

- Combined with a Gaussian prior over μ,

$$p(\mu) = \mathcal{N}\left(\mu|\mu_0, \sigma_0^2\right).$$

- this gives the posterior

$$p(\mu|\mathbf{x}) \propto p(\mathbf{x}|\mu)p(\mu).$$

- Completing the square over μ, we see that

$$p(\mu|\mathbf{x}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$$

Bayesian Inference for the Gaussian

- … where

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2+\sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2+\sigma^2}\mu_{\mathrm{ML}}, \qquad \mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}.$$

Shortcut: $p(\mu \mid X)$ has the form $C\exp(-\Delta^2)$.

Get Δ^2 in form $a\mu^2 - 2b\mu + c = a(\mu - b/a)^2 + \text{const}$ and identify

$\mu_N = b/a$

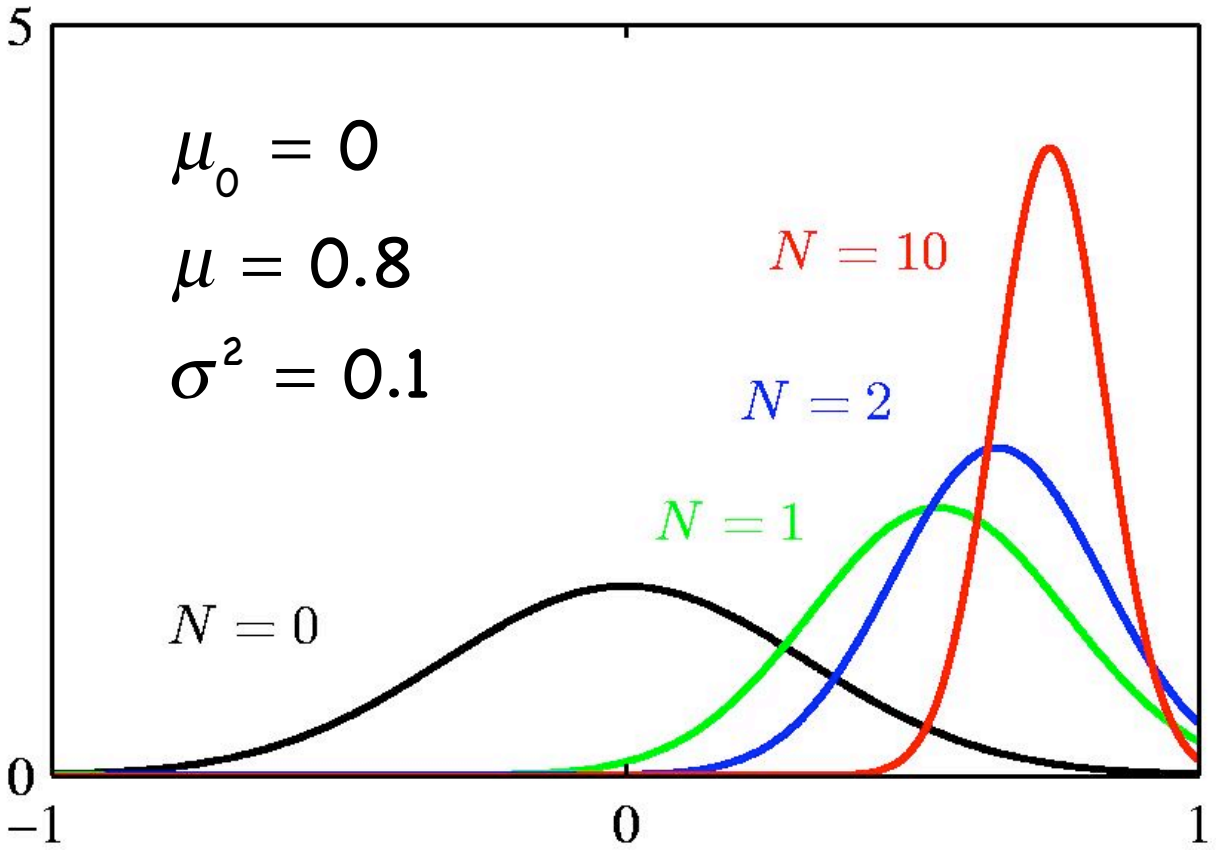
$$\frac{1}{\sigma_N^2} = a$$

- Note:

	$N=0$	$N \to \infty$
μ_N	μ_0	μ_{ML}
σ_N^2	σ_0^2	0

Bayesian Inference for the Gaussian

- Example: $p(\mu|\mathbf{x}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$



Maximum a Posteriori (MAP) Estimation

$$p(\mu|\mathbf{x}) = \mathcal{N}\left(\mu|\mu_N, \sigma_N^2\right)$$

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{\mathrm{ML}}, \qquad \mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}.$$

In MAP estimation, we use the value of μ that maximizes the posterior $p(\mu \mid X)$:

$\mu_{MAP} = \mu_N$.

Full Bayesian Parameter Estimation

- In both ML and MAP, we use the training data **X** to estimate a specific value for the unknown parameter vector $\underline{\theta}$, and then use that value for subsequent inference on new observations **x**: $p\left(\mathbf{x} \mid \underline{\theta}\right)$
- These methods are suboptimal, because in fact we are always uncertain about the exact value of $\underline{\theta}$, and to be optimal we should take into account the possibility that $\underline{\theta}$ assumes other values.

Full Bayesian Parameter Estimation

- In full Bayesian parameter estimation, we do not estimate a specific value for $\underline{\theta}$.
- Instead, we compute the posterior over $\underline{\theta}$, and then integrate it out when computing $p(\mathbf{x} \mid \mathbf{X})$:

$$p(\underline{x}|X) = \int p(\underline{x}|\underline{\theta})p(\underline{\theta}|X)d\underline{\theta}$$

$$p(\underline{\theta}|X) = \frac{p(X|\underline{\theta})p(\underline{\theta})}{p(X)} = \frac{p(X|\underline{\theta})p(\underline{\theta})}{\int p(X|\underline{\theta})p(\underline{\theta})d\underline{\theta}}$$

$$p(X|\underline{\theta}) = \prod_{k=1}^{N} p(\underline{x}_k|\underline{\theta})$$

Example: Univariate Normal with Unknown Mean

Consider again the case $p(\underline{x}|\mu) \sim N(\mu, \sigma)$ where σ is known and $\mu \sim N(\mu_0, \sigma_0)$

We showed that $p(\mu|X) \sim N(\mu_N, \sigma_N^2)$, where

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{\mathrm{ML}}, \qquad \mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}.$$

In the MAP approach, we approximate $p(\underline{x}|\underline{X}) \sim N(\mu_N, \sigma^2)$

In the full Bayesian approach, we calculate $p(\underline{x}|X) = \int p(\underline{x} \mid \mu) p(\mu|X)\, d\mu$

which can be shown to yield $p(\underline{x}|X) \sim N(\mu_N, \sigma^2 + \sigma_N^2)$

Hints for Exercise 1.4.2

- Here are some MATLAB functions you may find useful in solving Exercise 1.4.2
  - mnrnd
  - mvnrnd
  - mvnpdf
  - mean
  - cov
  - squeeze
  - sum
  - repmat
  - inv
  - min
  - max
  - zeros
  - ones

Problem 1.4.2


```
function pe=pr142
%Exercise 1.4.2 from PR Matlab Manual

m(:,1)=[0 0 0]';
m(:,2)=[1 2 2]';
m(:,3)=[3 3 4]';
S=[0.8 0.2 0.1;0.2 0.8 0.2; 0.1 0.2 0.8];
N=1000;

%Part 1
ntrain=mnrnd(N,ones(3,1)/3); %Number of training pts generated by each
class
ntest=mnrnd(N,ones(3,1)/3); %Number of test pts generated by each class
test=[];
mml=zeros(3,3);
Smli=zeros(3,3,3);
for i=1:3
    train=mvnrnd(m(:,i),S,ntrain(i)); %training vectors from class i
    test=[test;mvnrnd(m(:,i),S,ntest(i))]; %test vectors from class i

    mml(i,:)=mean(train); %ML estimate of mean for class i
    Smli(i,:,:)=ntest(i)*cov(train,1);%weighted ML estimate of
covariance for class i
end

Sml=squeeze(mean(Smli)/N); %ML estimate of common covariance

%Part 2:  Euclidean distance
for i=1:3
    dsq(:,i)=sum((test-repmat(mml(i,:),N,1)).^2,2);
end
[m,idx(:,1)]=min(dsq');

%Part 3: Mahalanobis distance
for i=1:3
    y=test-repmat(mml(i,:),N,1);
    dsq(:,i)=sum((y*inv(Sml)).*y,2);
end
[m,idx(:,2)]=min(dsq');

%Part 4:  Maximum likelihood classifier
for i=1:3
    p(:,i)=mvnpdf(test,mml(i,:),Sml);
end
[m,idx(:,3)]=max(p');

%Ground truth classes
idxgt=[ones(ntest(1),1);2*ones(ntest(2),1);3*ones(ntest(3),1)];

for i=1:3
    pe(i)=mean(idx(:,i)~=idxgt); %Error rate for class i
end
```

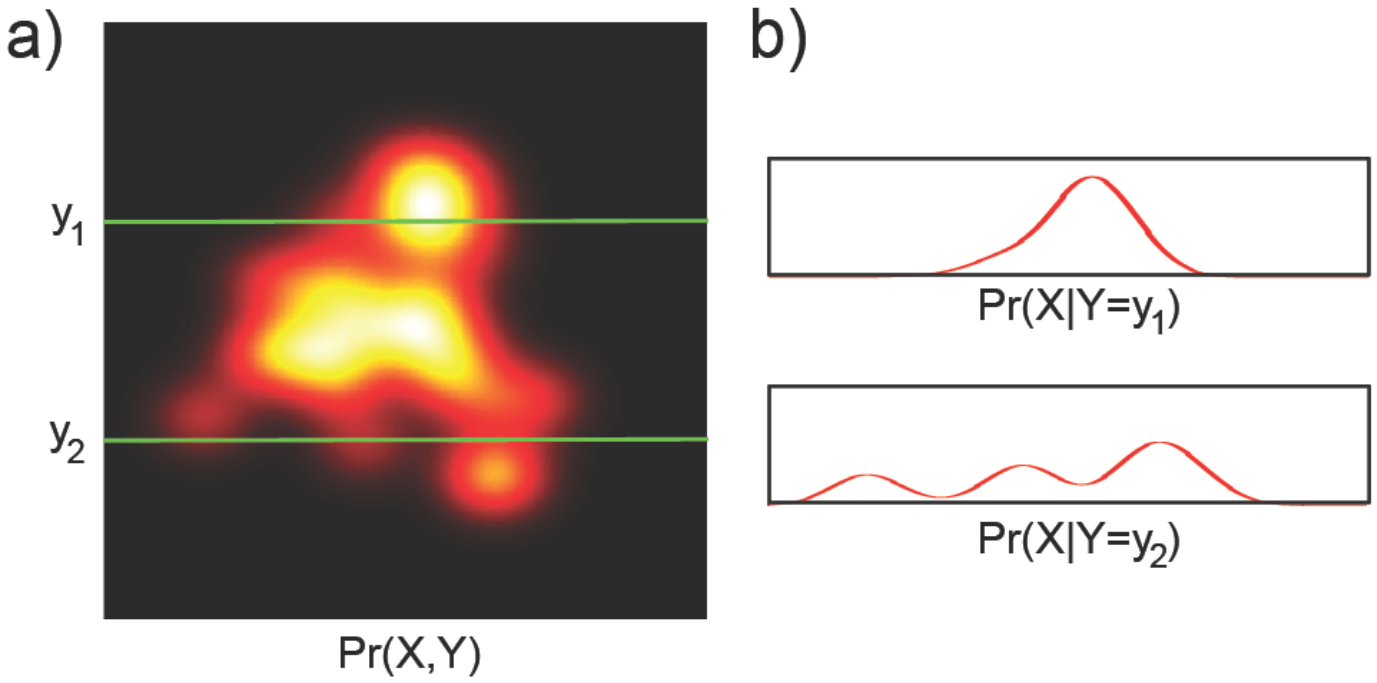
Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. **Mixture Models and EM**
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. What are Bayes Nets?

Topic 8 Mixture Models and EM

Motivation

- What do we do if a distribution is not well-approximated by a standard parametric model?



8.1 Intuition

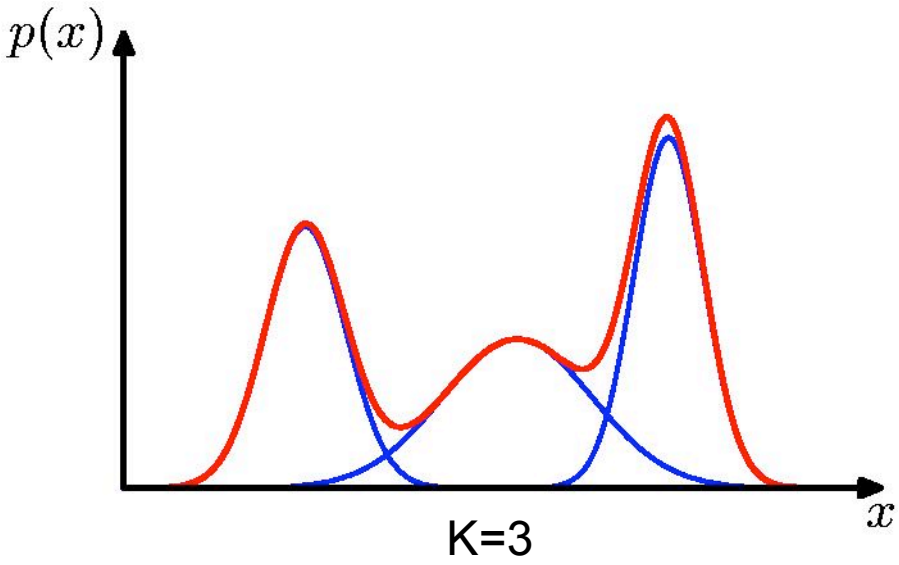
Mixtures of Gaussians

- Combine simple models into a complex model:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

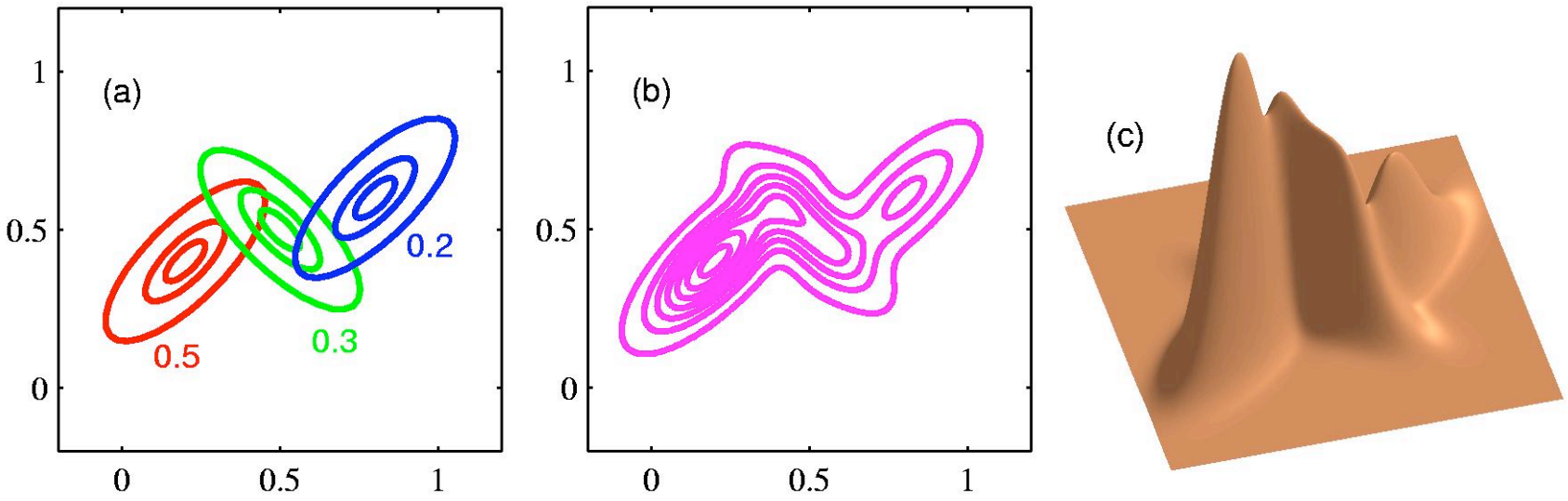
Component

Mixing coefficient



$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$

Mixtures of Gaussians



Mixtures of Gaussians

- Determining parameters μ, σ and π using maximum log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

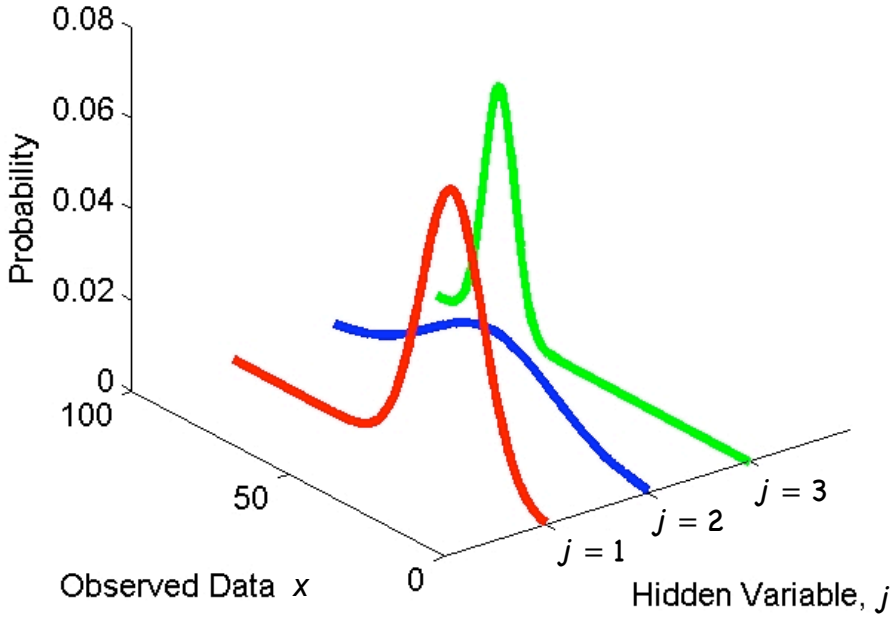
Log of a sum; no closed form maximum.

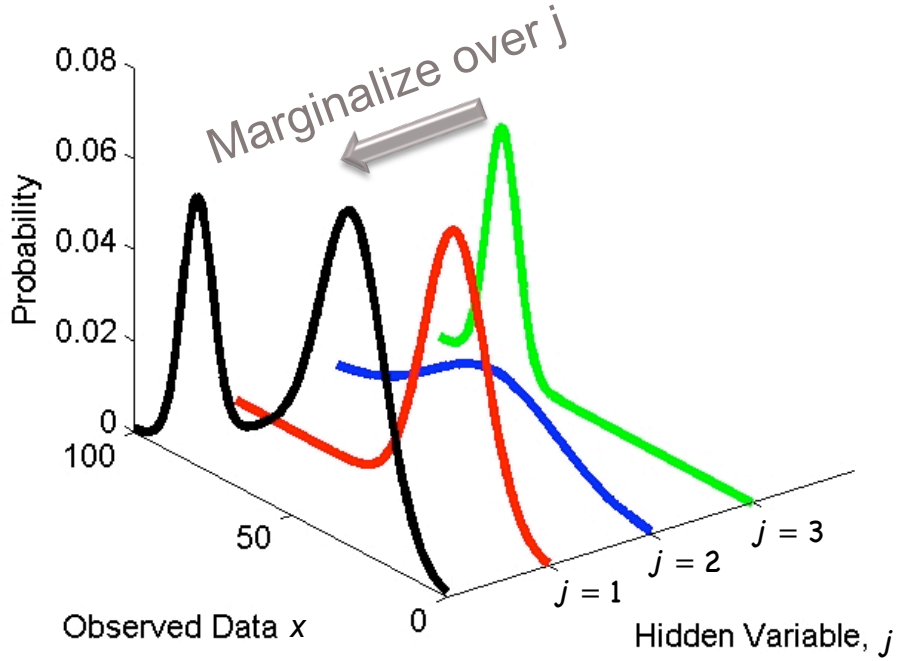
- Solution: use standard, iterative, numeric optimization methods or the ***expectation maximization*** algorithm.

End of Lecture 6

Hidden Variable Interpretation

$$p\left(x \mid P_1,\ldots P_J,\mu_1,\ldots\mu_J,\sigma_1,\ldots\sigma_J\right)=\sum_{j=1}^{J}P_j\mathcal{N}\left(x;\mu_j,\sigma_j^2\right)=\sum_{j=1}^{J}p(j)p\left(x \mid j\right)$$





Hidden Variable Interpretation

$$p\left(x \mid P_1,\ldots P_J,\mu_1,\ldots\mu_J,\sigma_1,\ldots\sigma_J\right)=\sum_{j=1}^{J}P_j\mathcal{N}\left(x;\mu_j,\sigma_j^2\right)=\sum_{j=1}^{J}p(j)p\left(x \mid j\right)$$

ASSUMPTIONS

- for each training datum x_i there is a hidden variable j_i.
- j_i represents which Gaussian x_i came from
- hence j_i takes discrete values

OUR GOAL:

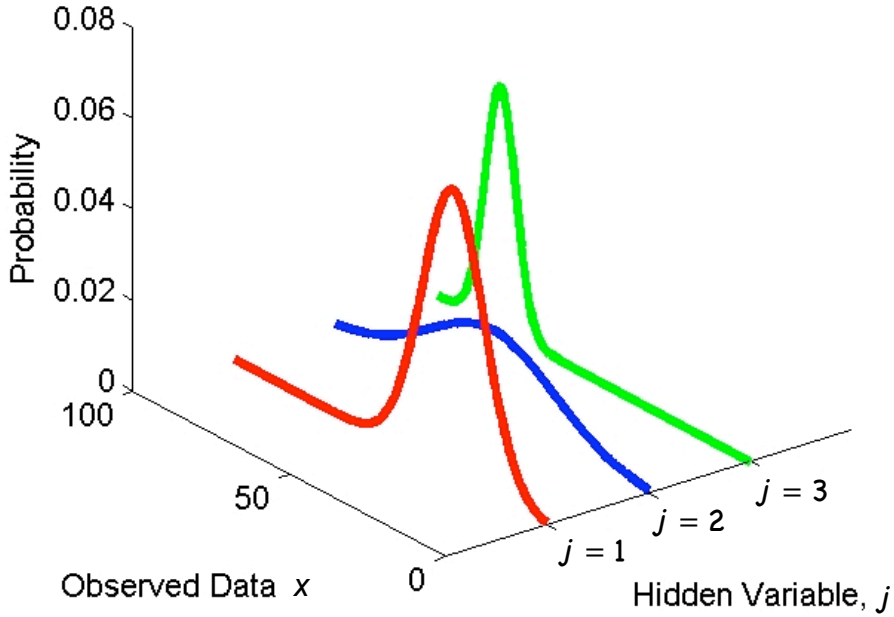
To estimate the parameters θ:

The means μ_j

The covariances Σ_j

The weights (mixing coefficients) P_j

for all J components of the model.



THING TO NOTICE:

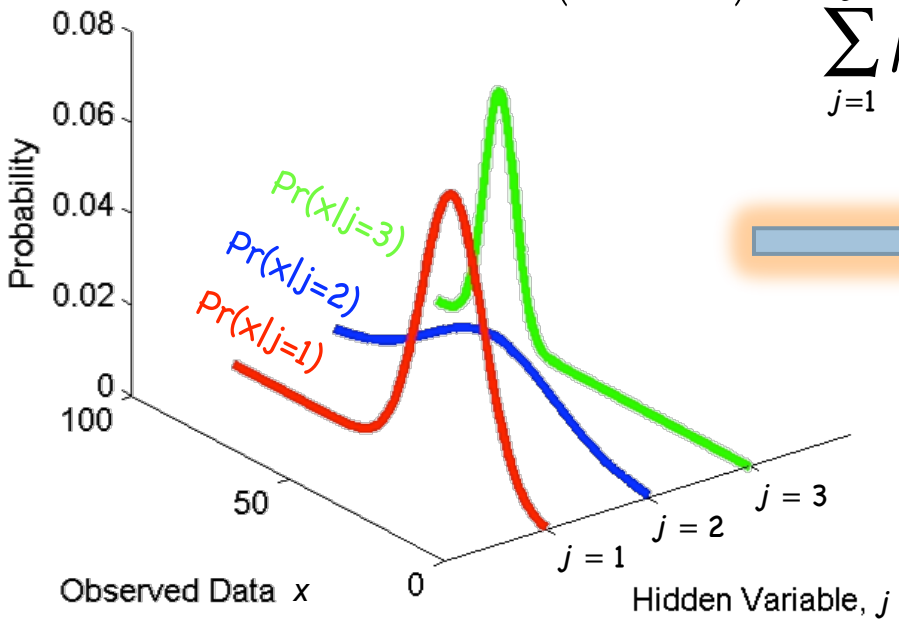
If we knew the hidden variables j_i for the training data it would be easy to estimate parameters θ – just estimate individual Gaussians separately.

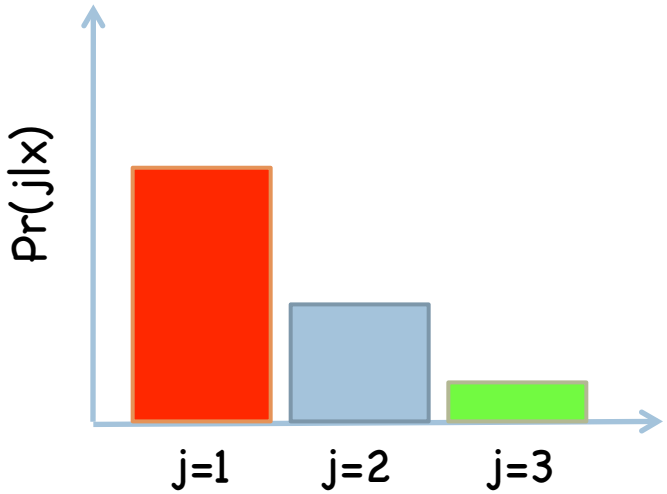
Hidden Variable Interpretation

THING TO NOTICE #2:

If we knew the parameters θ it would very easy to estimate the posterior distribution over each hidden variable j_i using Bayes' rule:

$$p(j \mid x, \theta) = \frac{p(x \mid j, \theta) P_j}{\sum_{j=1}^{J} p(x \mid j, \theta) P_j}$$





Expectation Maximization

Chicken and egg problem:

- could find $j_{1...N}$ if we knew θ
- could find θ if we knew $j_{1...N}$

Solution: Expectation Maximization (EM) algorithm

(Dempster, Laird and Rubin 1977)

EM for Gaussian mixtures can be viewed as alternation between 2 steps:

1. Expectation Step (E-Step)
   - For fixed θ find posterior distribution over responsibilities $j_{1...N}$
2. Maximization Step (M-Step)
   - Now use these posterior probabilities to re-estimate θ

8.2 Math

Mixture Model

Let

$\mathbf{x}_k, k = 1,\ldots N$ denote the training input observations, assumed to be independent

$j_k \in [1,\ldots J]$ indicate the component of the mixture from which the observation was drawn

(Note that $\mathbf{x}_k$ is observable but j_k is hidden.)

Let

$\Theta^t = \left(\theta^t, \mathbf{P}^t\right)$ represent the unknown parameters we are trying to estimate, where

θ represents the vector of coefficients for the component distributions and

$\mathbf{P}$ represents the mixing coefficients.

Our mixture model is $p\left(\mathbf{x}_k \mid \Theta\right) = \sum_{j=1}^{J} P_{j_k} p\left(\mathbf{x}_k \mid j_k; \Theta\right)$

Q Function

We will iteratively estimate Θ, starting with an initial guess $\Theta(0)$ and monotonically improving our estimate $\Theta(t)$ at successive time steps t.

For this purpose, we define a **Q function**

$$Q\left(\Theta;\Theta(t)\right) = E\left[\sum_{k=1}^{N} P_{j_k} \log p\left(\mathbf{x}_k \mid j_k;\theta\right)\right]$$

The Q function represents the **expected log likelihood** of the training data, given our most recent estimate of the parameters $\Theta(t)$, where the expectation is taken over the possible values of the hidden labels j_k.

Expectation Step

- In the E-Step, we calculate the (expected) log probability over the possible parameter values:

$$
\begin{aligned}
Q\left(\Theta;\Theta(t)\right) &= E\left[\sum_{k=1}^{N} P_{j_k} \log p\left(\mathbf{x}_k \mid j_k;\theta\right)\right] \\
&= \sum_{k=1}^{N} E\left[P_{j_k} \log p\left(\mathbf{x}_k \mid j_k;\theta\right)\right] \\
&= \sum_{k=1}^{N}\sum_{j_k=1}^{J} P\left(j_k \mid \mathbf{x}_k;\Theta\right) P_{j_k} \log p\left(\mathbf{x}_k \mid j_k;\theta\right)
\end{aligned}
$$

Maximization Step

- In the M-Step, we select for our new parameter estimate the value that maximizes this expected log probability:

$$\Theta(t+1) = \arg\max_{\Theta} Q\left(\Theta; \Theta(t)\right)$$

Example: Mixture of Isotropic Gaussians

$$p\left(\mathbf{x}_k \mid j;\theta\right) = \frac{1}{\left(2\pi\sigma_j^2\right)^{\frac{\ell}{2}}} \exp\left(-\frac{\left\|\mathbf{x}_k - \mu_j\right\|^2}{2\sigma_j^2}\right)$$

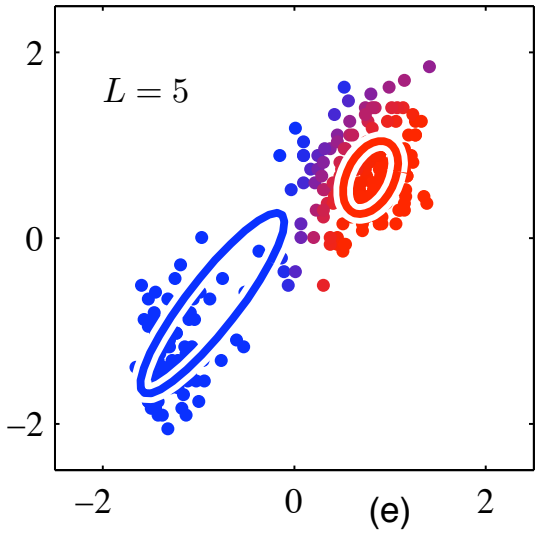
- E-Step:

$$Q\left(\Theta;\Theta(t)\right) = \sum_{k=1}^{N}\sum_{j=1}^{J} p\left(j \mid \mathbf{x}_k;\Theta\right)\left(-\frac{l}{2}\log\sigma_j^2 - \frac{1}{2\sigma_j^2}\left\|\mathbf{x}_k - \mu_j\right\|^2 + \log P_j\right)$$

Example: Mixture of Isotropic Gaussians

- Responsibilities Update Equation:

$$P\left(j \mid \mathbf{x}_k; \Theta(t)\right) = \frac{p\left(\mathbf{x}_k \mid j; \theta(t)\right) P_j(t)}{\sum_{j=1}^{J} p\left(\mathbf{x}_k \mid j; \theta(t)\right) P_j(t)}$$



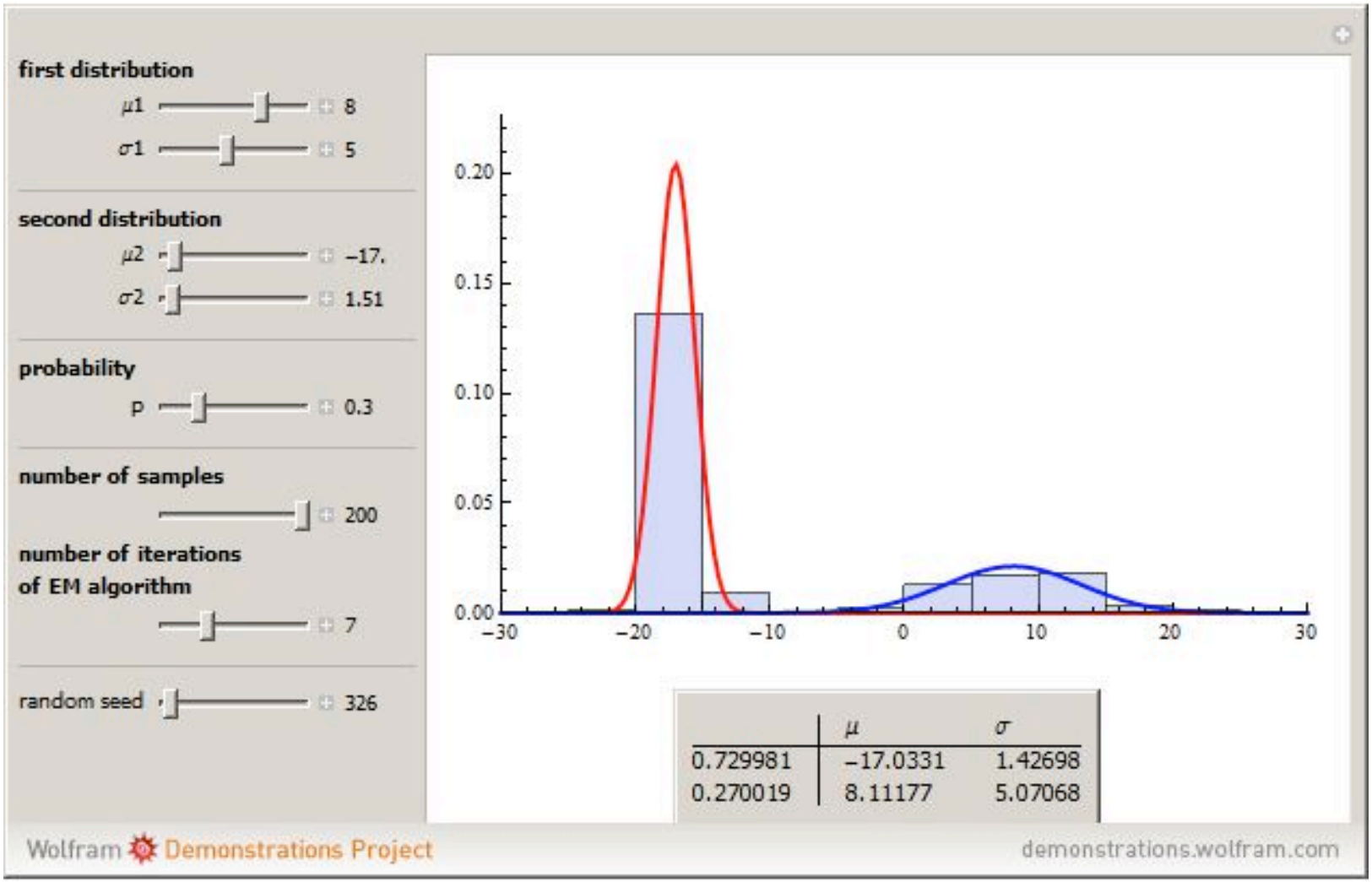
- Parameter Update Equations:

$$\mu_j(t+1) = \frac{\sum_{k=1}^{N} P\left(j \mid \mathbf{x}_k; \Theta(t)\right) \mathbf{x}_k}{\sum_{k=1}^{N} P\left(j \mid \mathbf{x}_k; \Theta(t)\right)}$$

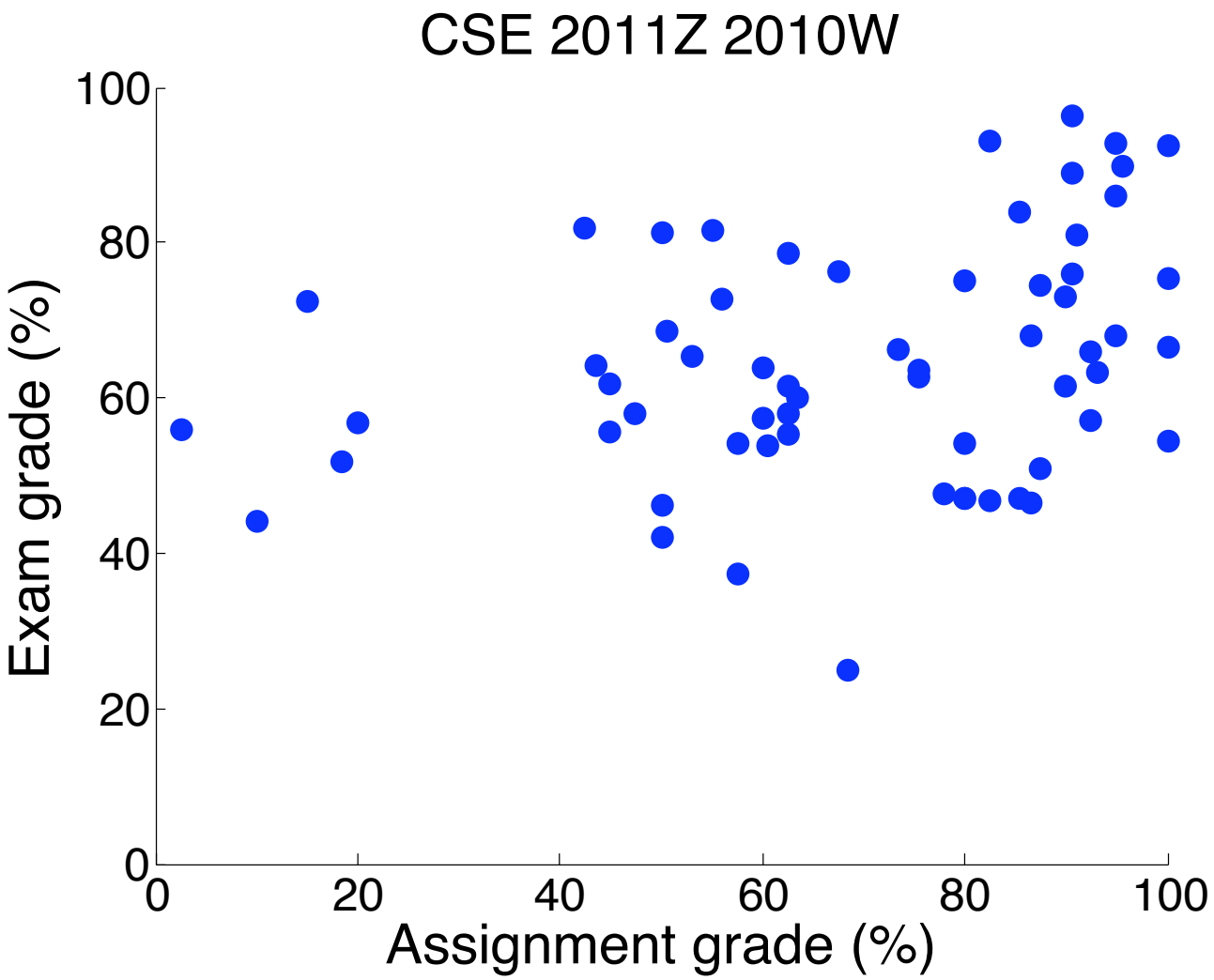
$$\sigma_j^2(t+1) = \frac{\sum_{k=1}^{N} P\left(j \mid \mathbf{x}_k; \Theta(t)\right) \left\| \mathbf{x}_k - \mu_j(t+1) \right\|^2}{l \sum_{k=1}^{N} P\left(j \mid \mathbf{x}_k; \Theta(t)\right)}$$

$$P_j(t+1) = \frac{1}{N} \sum_{k=1}^{N} P\left(j \mid \mathbf{x}_k; \Theta(t)\right)$$

Univariate Gaussian Mixture Example

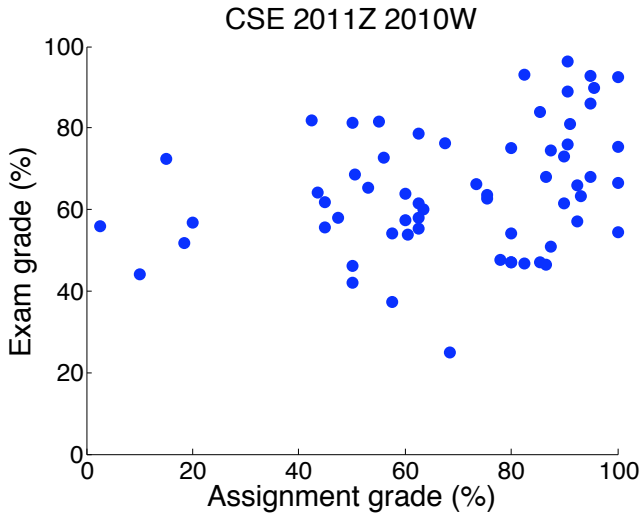


2-Component Bivariate MATLAB Example

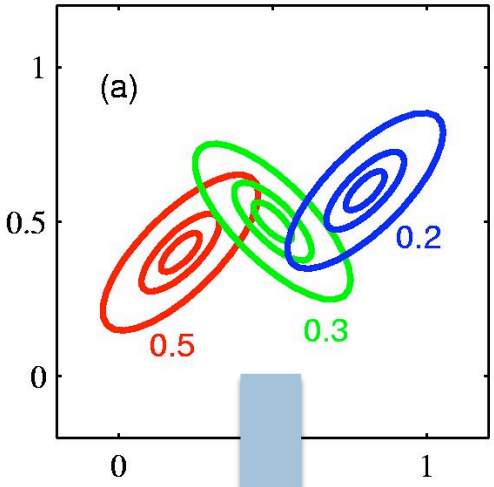


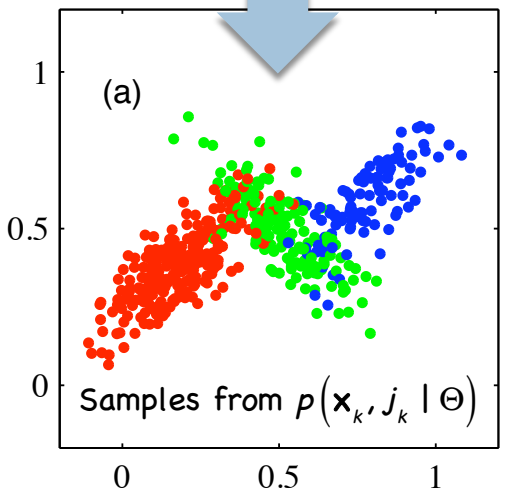
2-Component Bivariate MATLAB Example

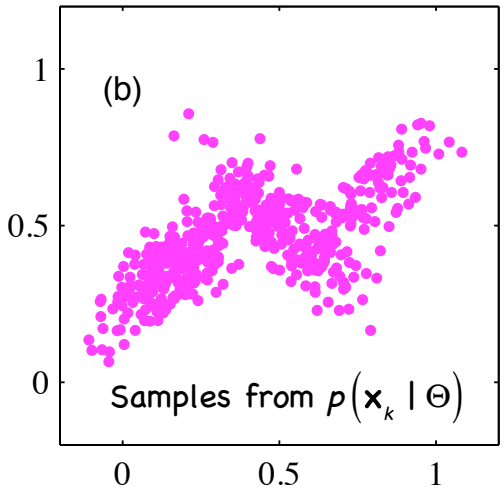

```
%update responsibilities
    for i=1:k
        p(:,i)=alphas(i).*mvnpdf(x,mu(i,:),squeeze(S(i,:,:)));
    end
    p=p./repmat((sum(p,2)),1,k);


%update parameters
    for i=1:k
        Nk=sum(p(:,i));
        mu(i,:)=p(:,i)'*x/Nk;
        dev=x-repmat(mu(i,:),N,1);
        S(i,:,:)=(repmat(p(:,i),1,D).*dev)'*dev/Nk;
        alphas(i)=Nk/N;
    end
```

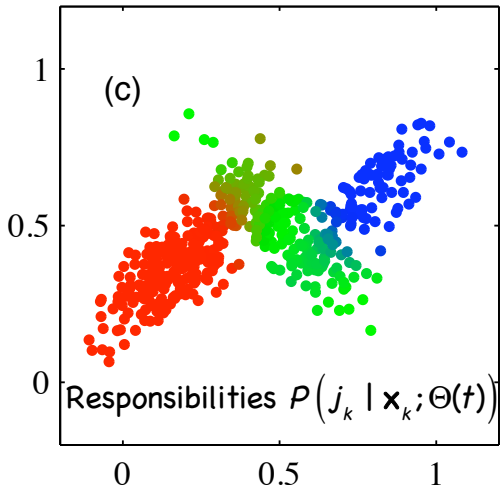


Bivariate Gaussian Mixture Example







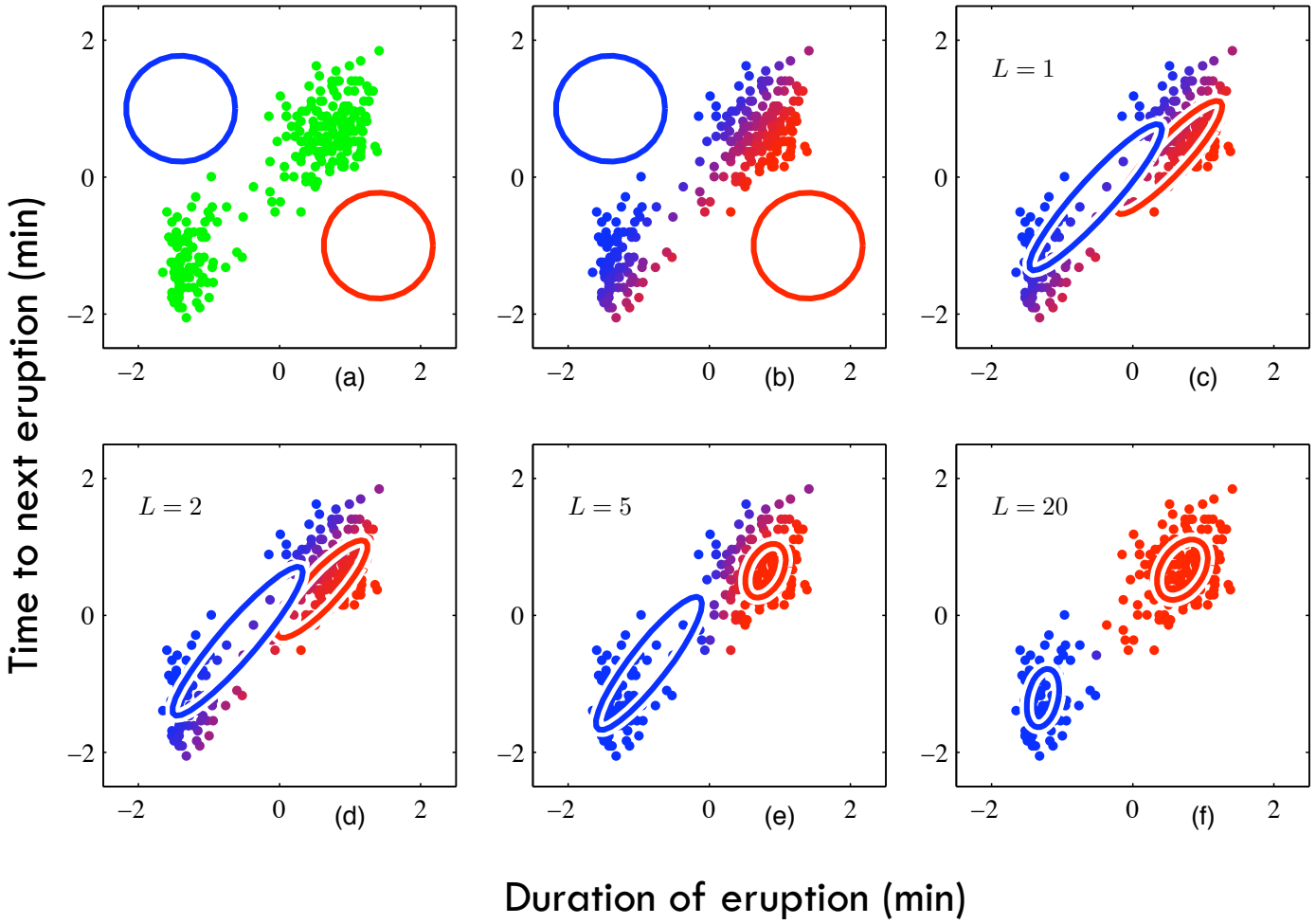


Expectation Maximization

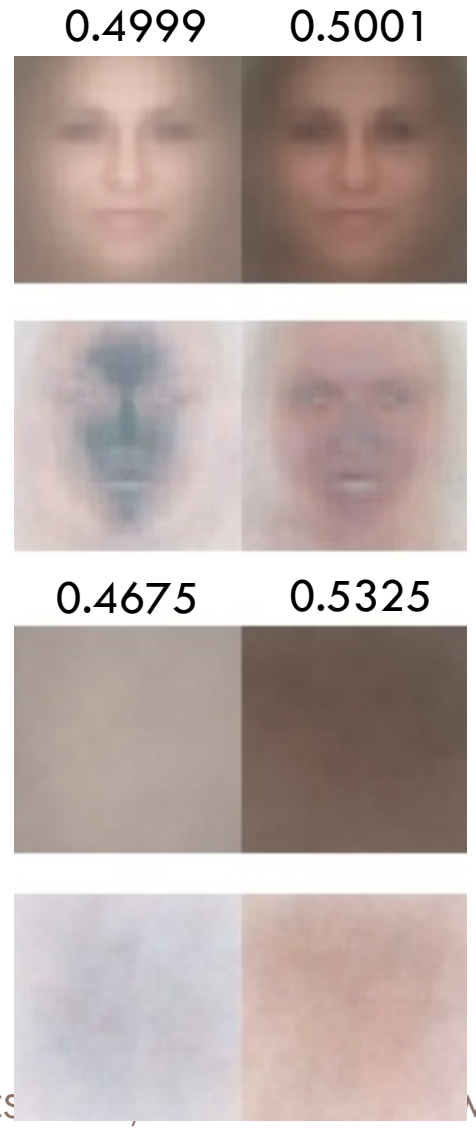
- EM is guaranteed to monotonically increase the likelihood.
- However, since in general the likelihood is non-convex, we are not guaranteed to find the globally optimal parameters.

8.3 Applications

Old Faithful Example



Face Detection Example: 2 Components



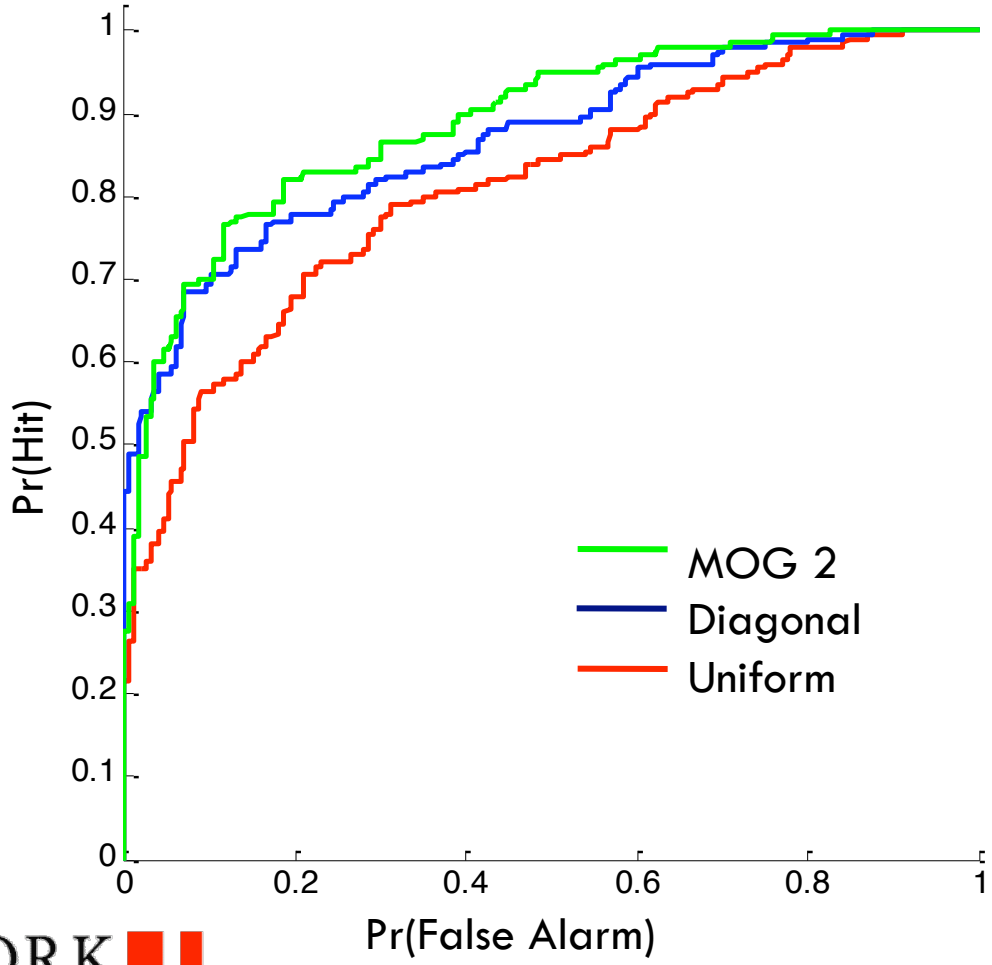
Each component is still assumed to have diagonal covariance.

The face model and non-face model have divided the data into two clusters. In each case, these clusters have roughly equal weights.

The primary thing that these seem to have captured is the photometric (luminance) variation.

Note that the standard deviations have become smaller than for the single Gaussian model as any given data point is likely to be close to one mean or the other.

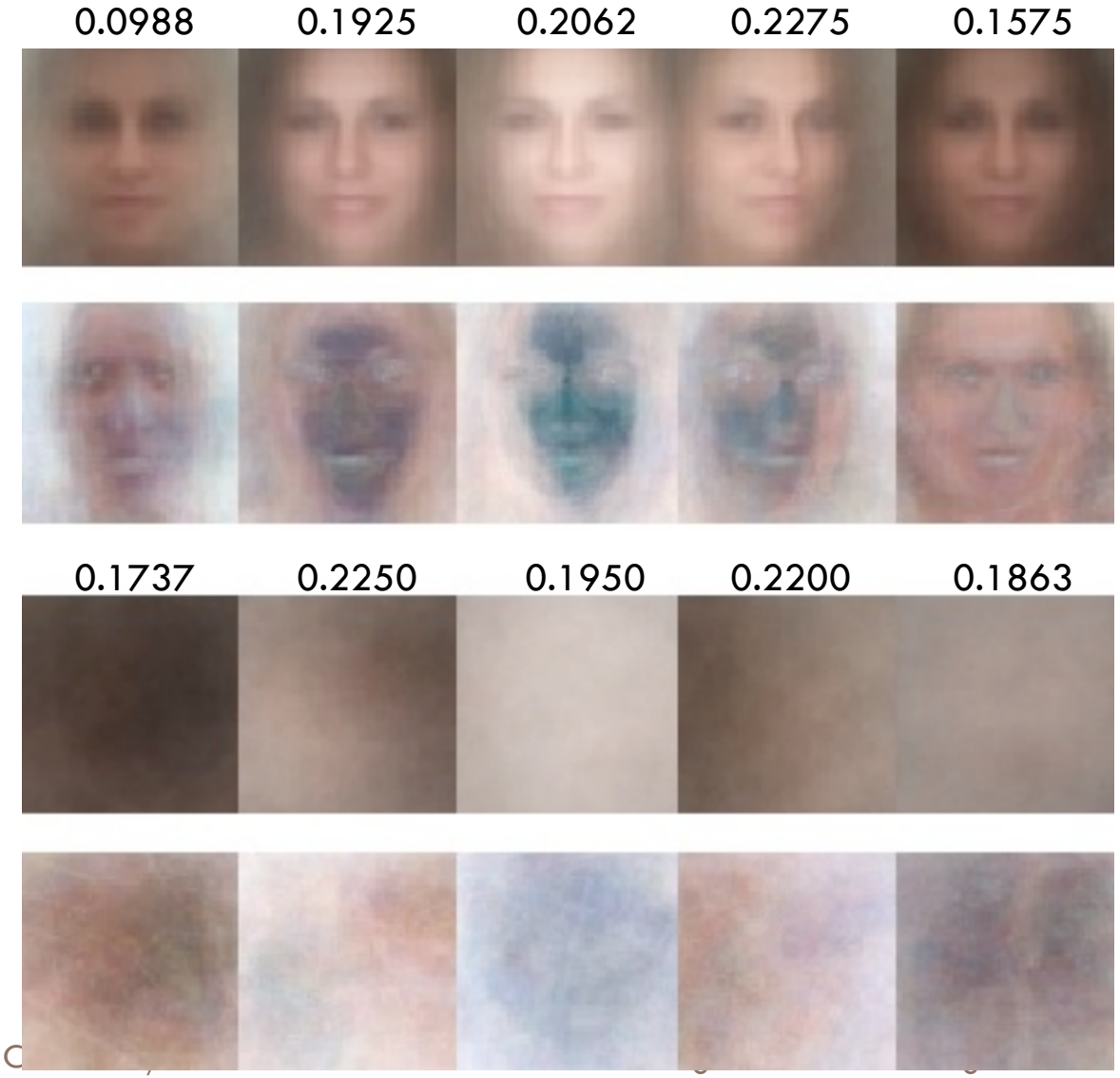
Results for MOG 2 Model



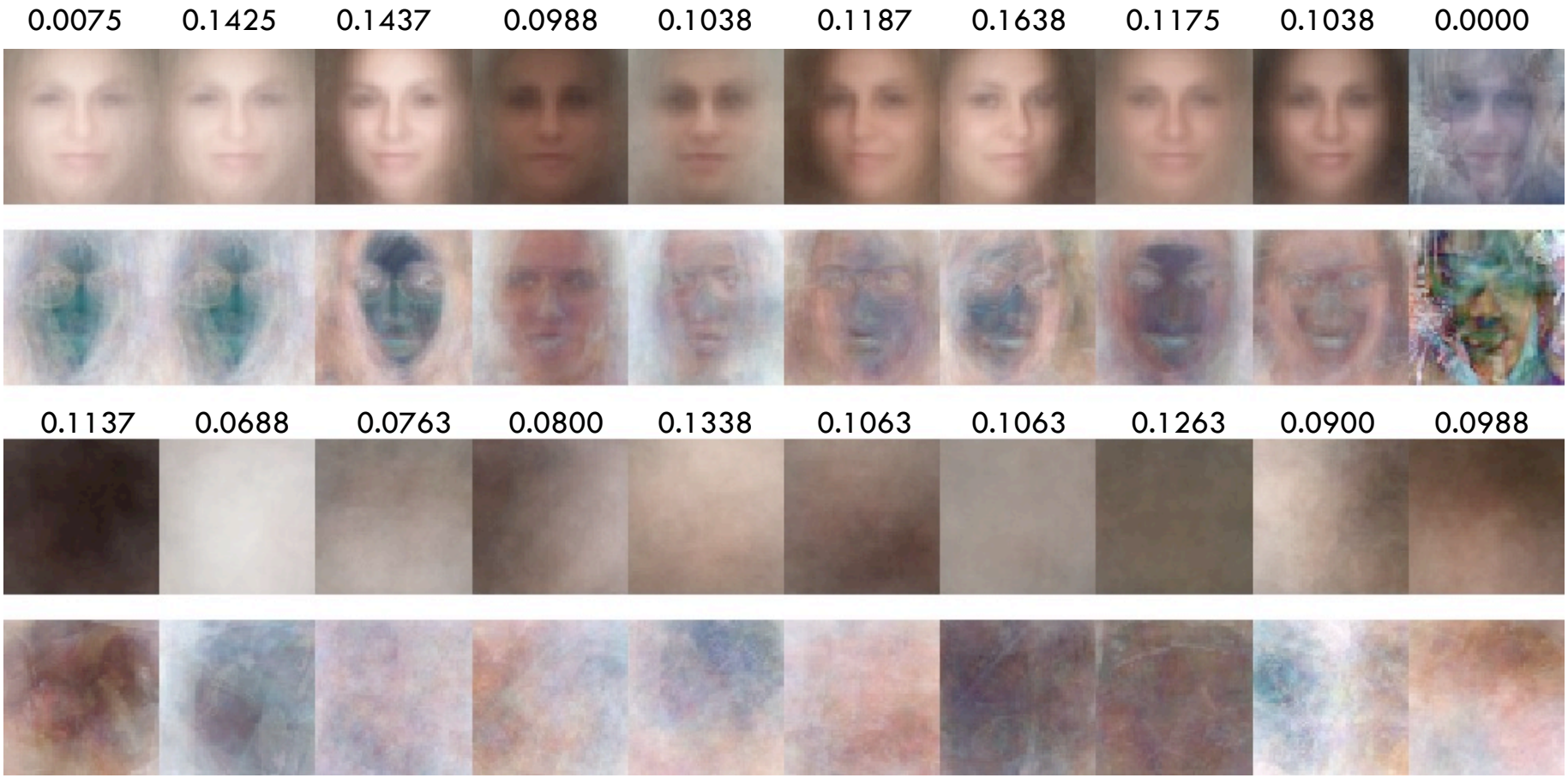
Performance improves relative to a single Gaussian model, although it is not dramatic.

We have a better description of the data likelihood.

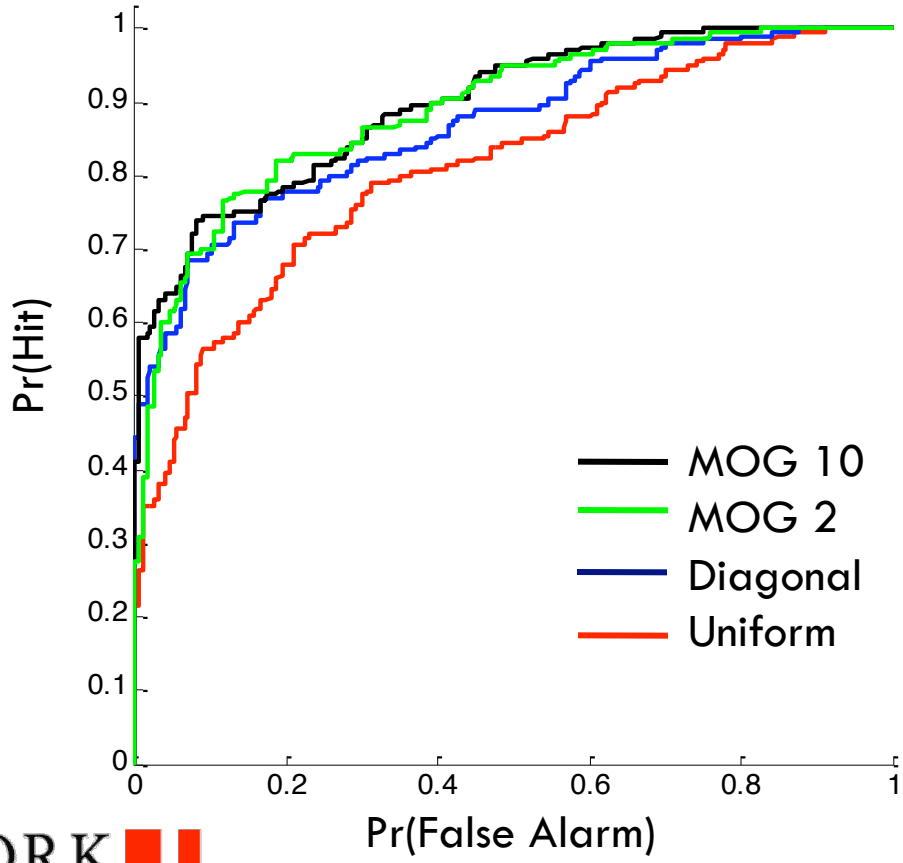
MOG 5 Components



MOG 10 Components



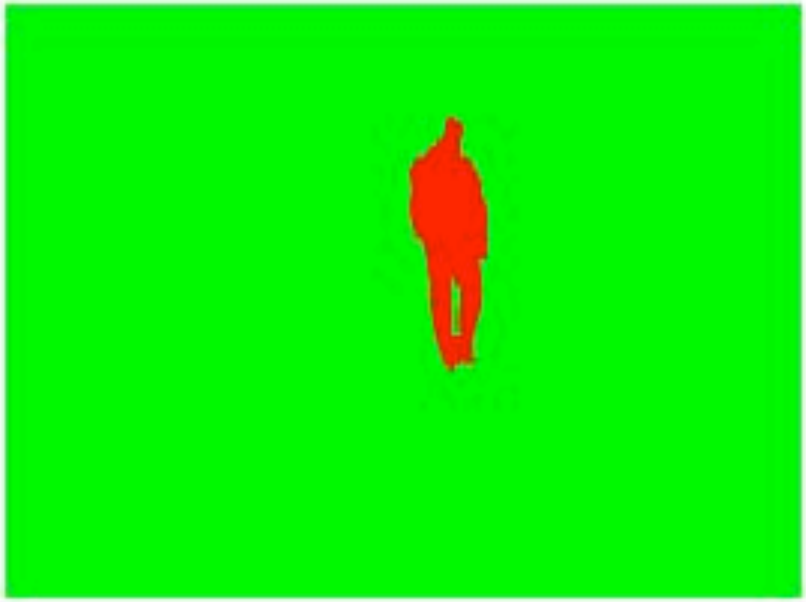
Results for Mog 10 Model



Performance improves slightly more, particularly at low false alarm rates.

Background Subtraction

Test Image



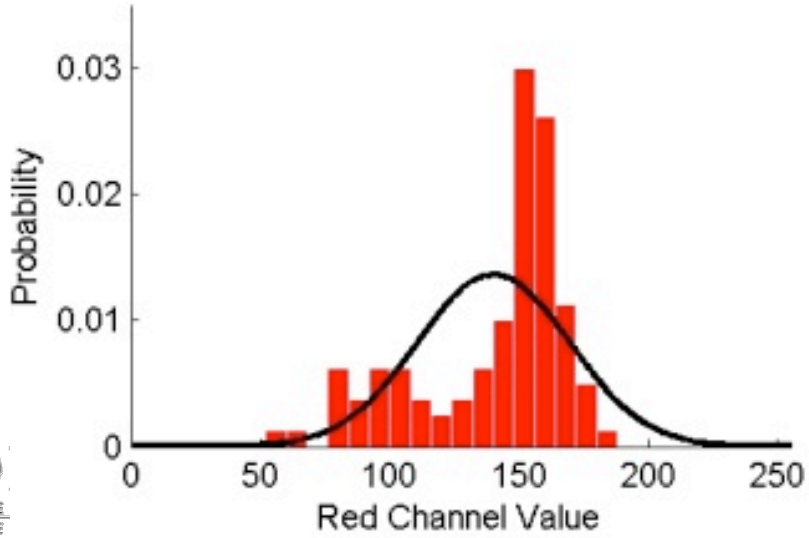
Desired Segmentation

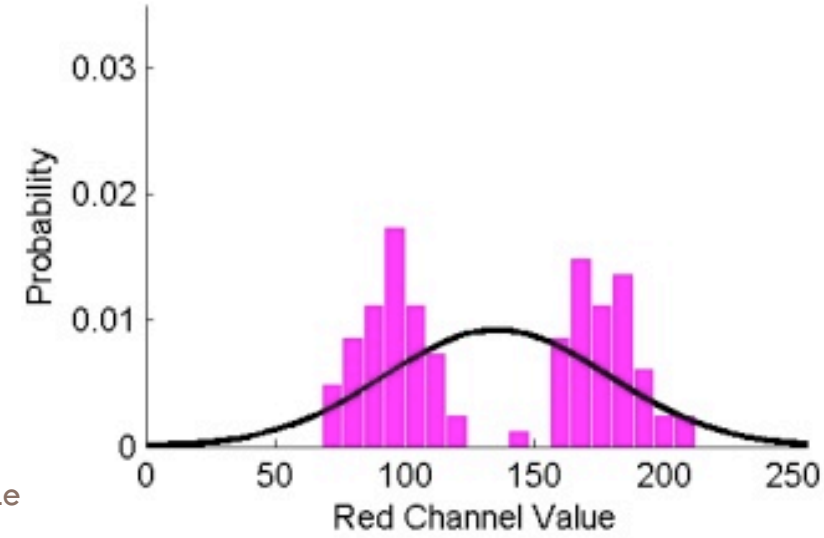
GOAL : (i) Learn background model (ii) use this to segment regions where the background has been occluded

What if the scene isn't static?

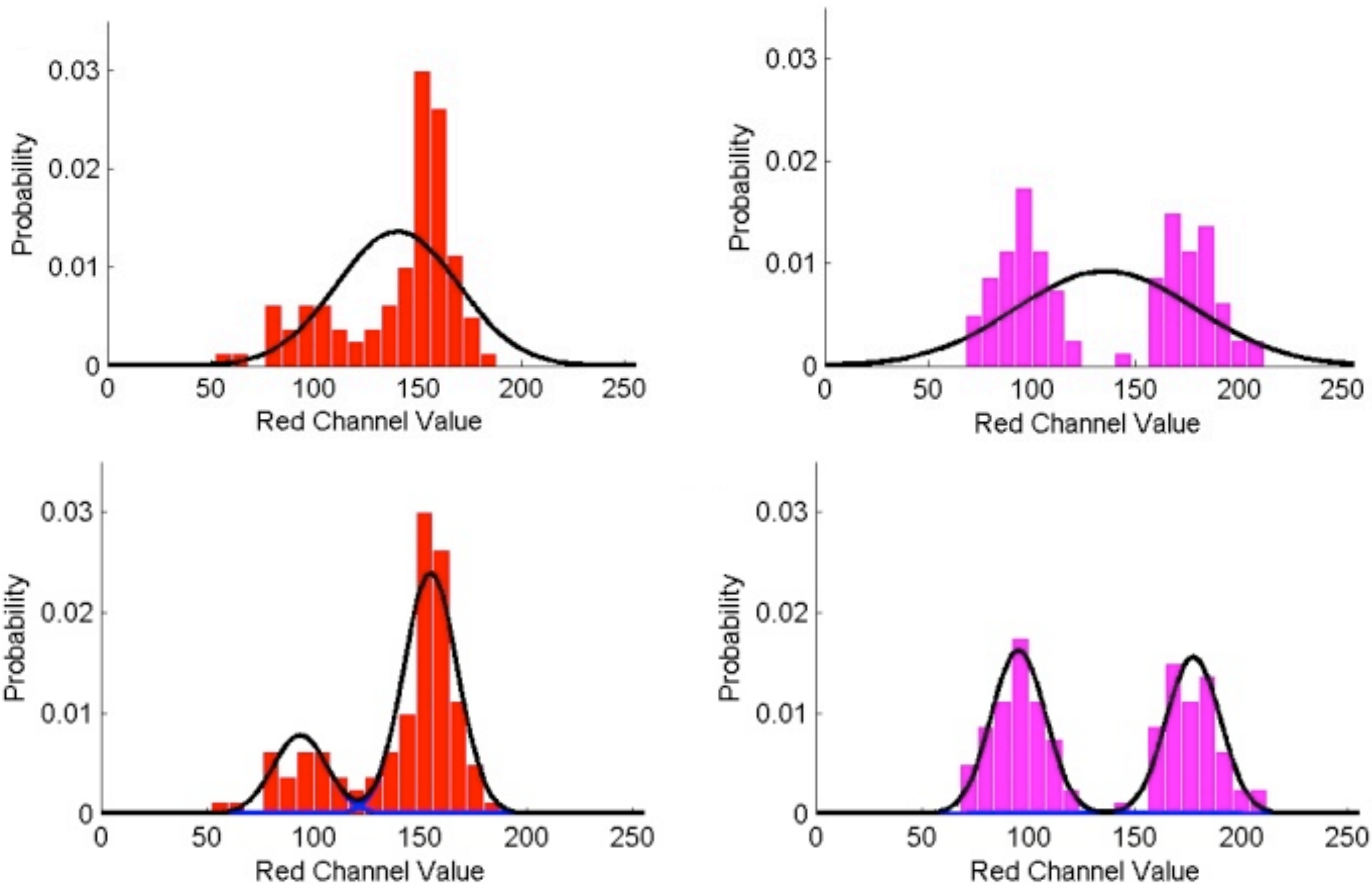
Gaussian is no longer a good fit to the data.

Not obvious exactly what probability model would fit better.

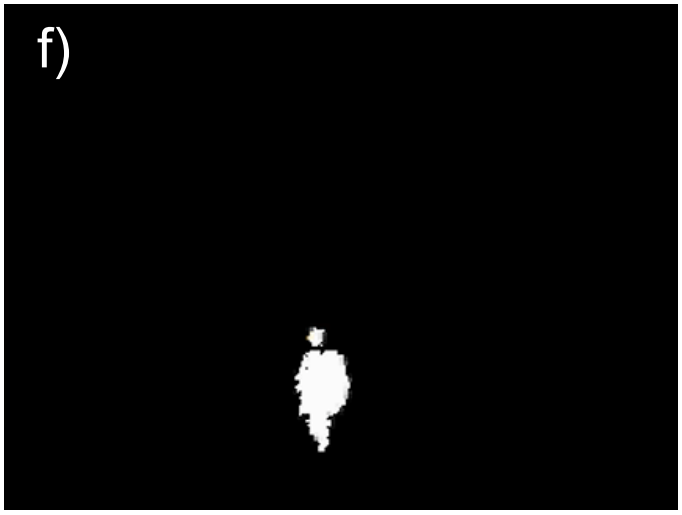




Background Mixture Model



Background Subtraction Example



End of Lecture 7

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. **Nonparametric Density Estimation**
10. Training and Evaluation Methods
11. What are Bayes Nets?

9. Nonparametric Methods

Nonparametric Methods

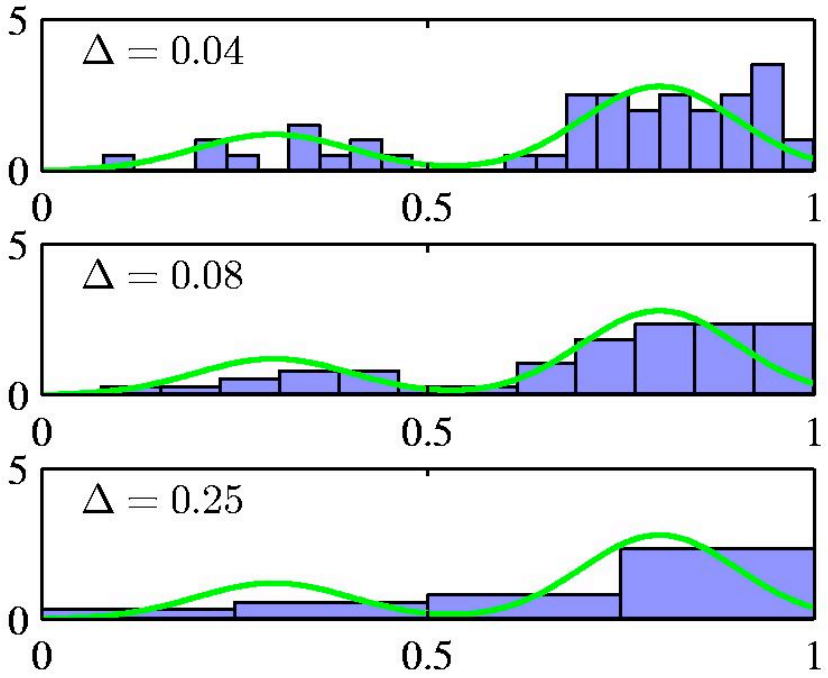
- Parametric distribution models are restricted to specific forms, which may not always be suitable; for example, consider modelling a multimodal distribution with a single, unimodal model.
- You can use a mixture model, but then you have to decide on the number of components, and hope that your parameter estimation algorithm (e.g., EM) converges to a global optimum!
- Nonparametric approaches make few assumptions about the overall shape of the distribution being modelled, and in some cases may be simpler than using a mixture model.

Histogramming

- **Histogram methods** partition the data space into distinct bins with widths Δ_i and count the number of observations, n_i, in each bin.

$$p_i = \frac{n_i}{N\Delta_i}$$

- Often, the same width is used for all bins, $\Delta_i = \Delta$.
- Δ acts as a smoothing parameter.



- In a D-dimensional space, using M bins in each dimension will require M^D bins!

The curse of dimensionality

Kernel Density Estimation

- Assume observations drawn from a density $p(x)$ and consider a small region R containing x such that

$$P = \int_{\mathcal{R}} p(\mathbf{x})\, \mathrm{d}\mathbf{x}.$$

- The expected number K out of N observations that will lie inside R is given by

$$K \simeq NP.$$

- If the volume V of R is sufficiently small, $p(x)$ is approximately constant over R and

$$P \simeq p(\mathbf{x})V$$

- Thus

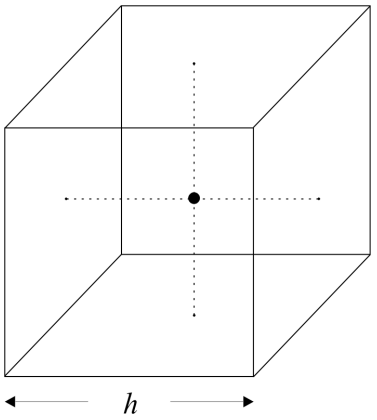
$$p(\mathbf{x}) = \frac{K}{NV}.$$

Kernel Density Estimation

Kernel Density Estimation: fix *V*, estimate *K* from the data. Let *R* be a hypercube centred on *x* and define the kernel function (Parzen window)

$$p(\mathbf{x}) = \frac{K}{NV}.$$

$$k((\mathbf{x}-\mathbf{x}_n)/h) = \begin{cases} 1, & |(x_i - x_{ni})/h| \leqslant 1/2, \quad i = 1,\ldots,D, \\ 0, & \text{otherwise.} \end{cases}$$



It follows that

$$K = \sum_{n=1}^{N} k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right)$$

and hence

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{h^D}k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right).$$

Kernel Density Estimation

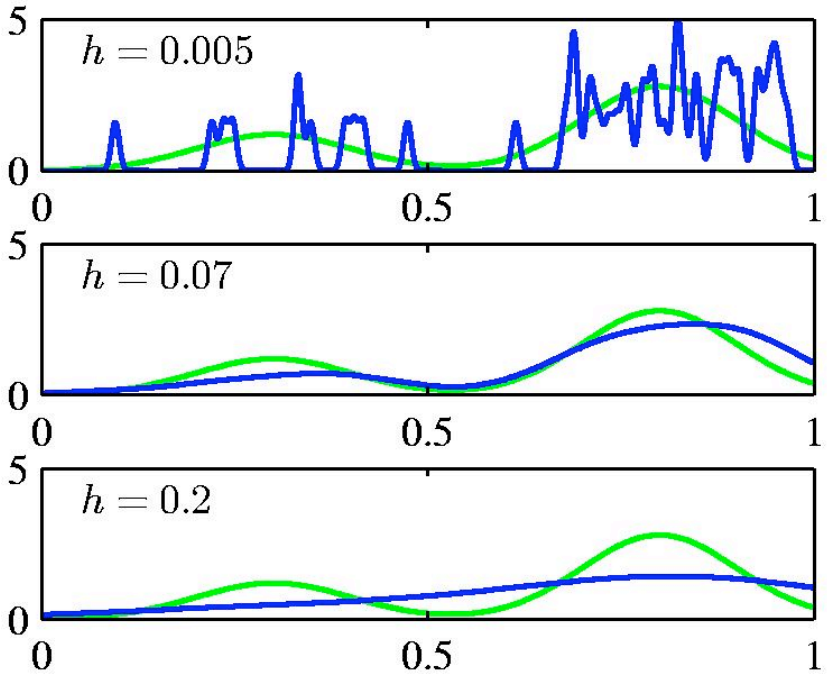
To avoid discontinuities in p(x), use a smooth kernel, e.g. a Gaussian

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{D/2}} \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2h^2}\right\}$$

(Any kernel *k(u)* such that

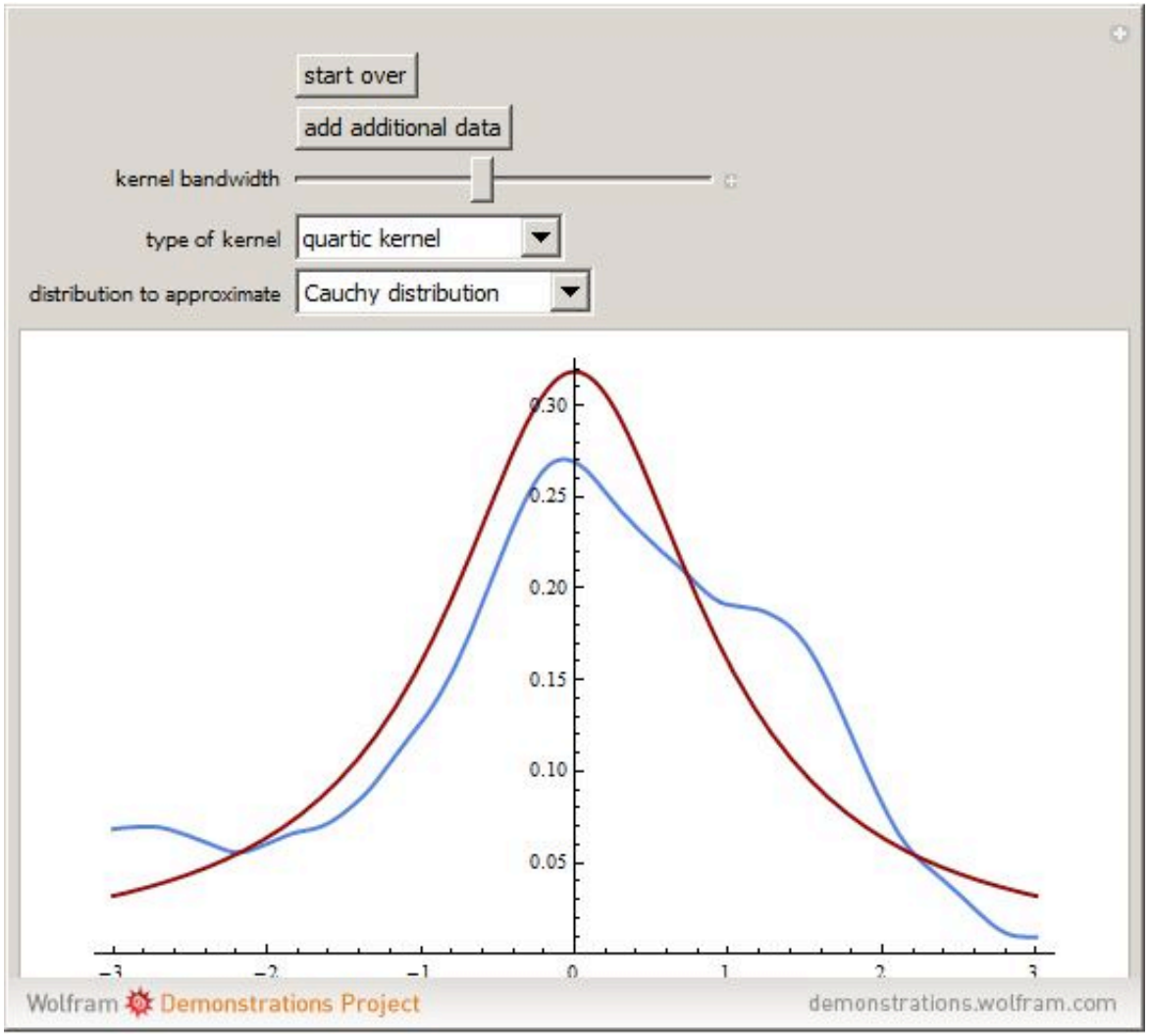
$$k(\mathbf{u}) \geqslant 0,$$
$$\int k(\mathbf{u})\,\mathrm{d}\mathbf{u} = 1$$

will work.)



h acts as a smoother.

KDE Example


start over
add additional data
kernel bandwidth
type of kernel
quartic kernel
distribution to approximate
Cauchy distribution
0.30
0.25
0.20
0.15
0.10
0.05
−3
−2
−1
0
1
2
3
Wolfram Demonstrations Project
demonstrations.wolfram.com

Kernel Density Estimation

- Problem: if V is fixed, there may be too few points in some regions to get an accurate estimate.

Nearest Neighbour Density Estimation

Nearest Neighbour Density Estimation: fix *K*, estimate *V* from the data. Consider a hypersphere centred on *x* and let it grow to a volume *V** that includes *K* of the given *N* data points. Then

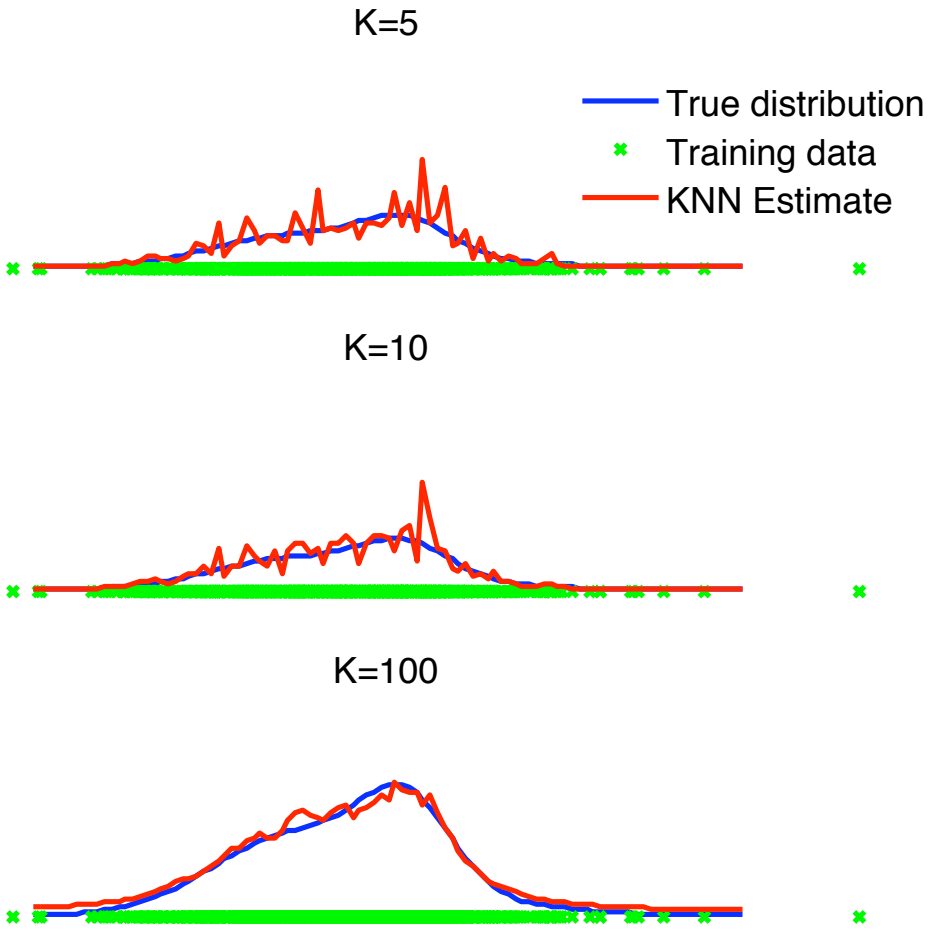
$$p(\mathbf{x}) \simeq \frac{K}{NV^{\star}}.$$

```
for j=1:np
    d=sort(abs(x(j)-xi));
    V=2*d(K(i));
    phat(j)=K(i)/(N*V);
end
```

Nearest Neighbour Density Estimation

Nearest Neighbour Density Estimation: fix *K*, estimate *V* from the data. Consider a hypersphere centred on *x* and let it grow to a volume *V** that includes *K* of the given *N* data points. Then

$$p(\mathbf{x}) \simeq \frac{K}{NV^{\star}}.$$



Nearest Neighbour Density Estimation

- Problem: does not generate a proper density (for example, integral is unbounded on $\mathbb{R}^D$)
- In practice, on finite domains, can normalize.
- But makes strong assumption on tails $\left(\propto \frac{1}{x}\right)$

Nonparametric Methods

- Nonparametric models (not histograms) require storing and computing with the entire data set.
- Parametric models, once fitted, are much more efficient in terms of storage and computation.

K-Nearest-Neighbours for Classification

- Given a data set with N_k data points from class C_k and $\sum_k N_k = N$, we have

$$p(\mathbf{x}) = \frac{K}{NV}$$

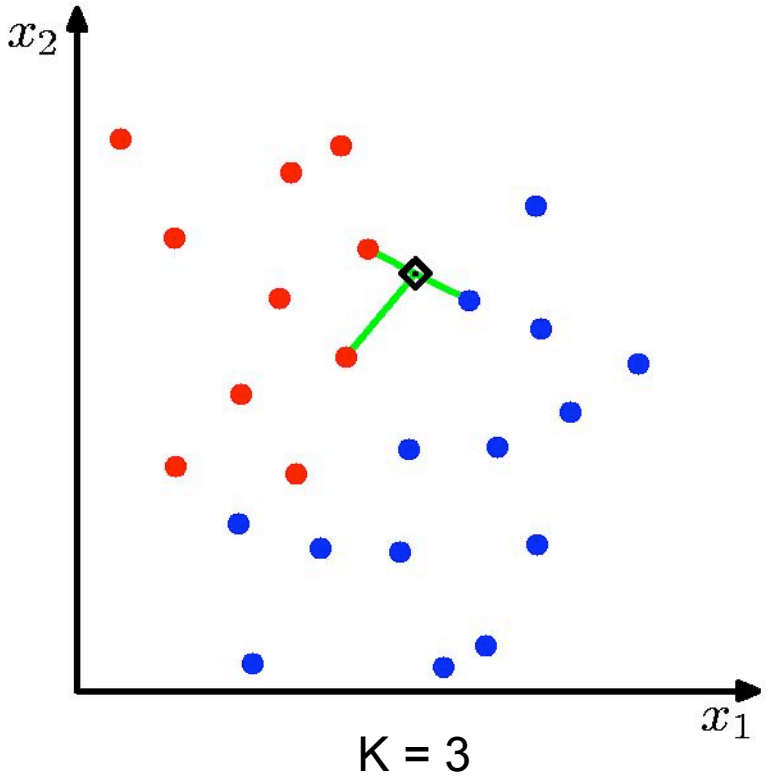
- and correspondingly

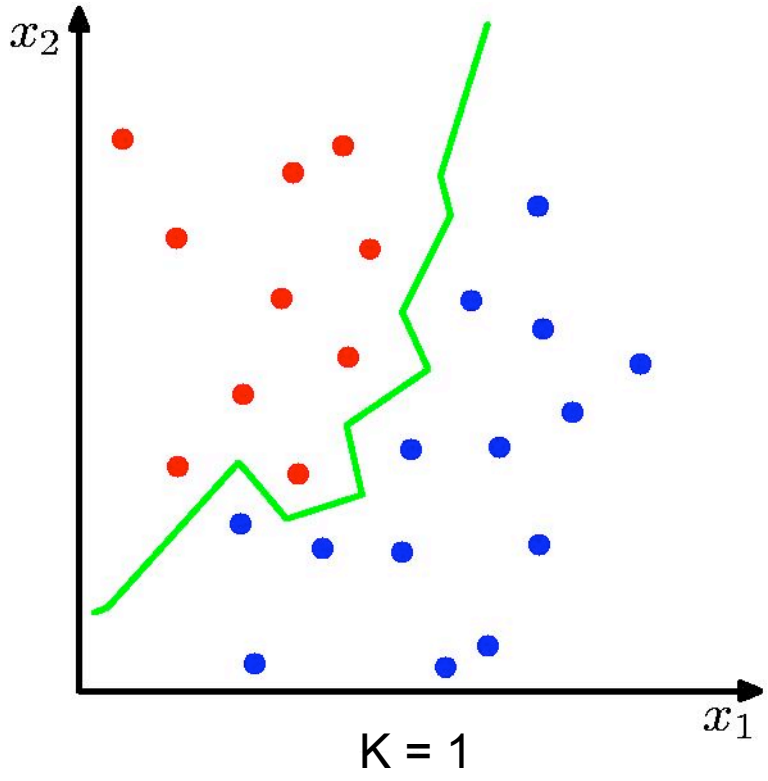
$$p(\mathbf{x}|\mathcal{C}_k) = \frac{K_k}{N_k V}.$$

- Since $p(\mathcal{C}_k) = N_k/N$, Bayes' theorem gives

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} = \frac{K_k}{K}.$$

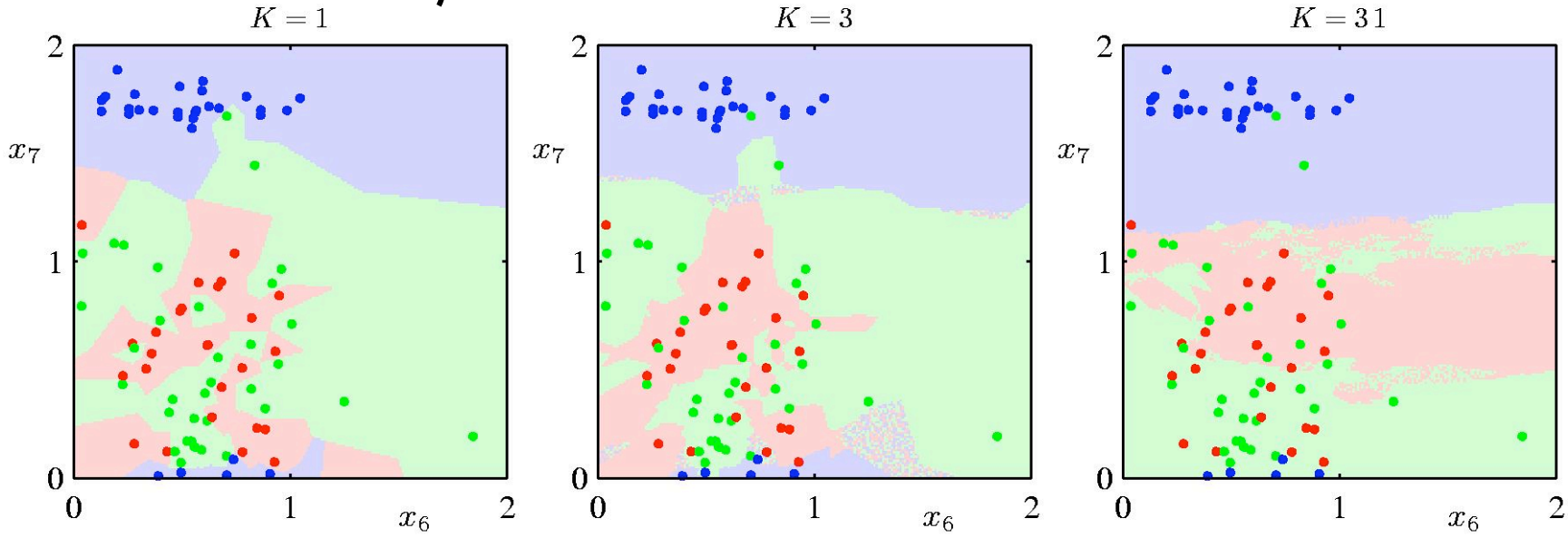
K-Nearest-Neighbours for Classification



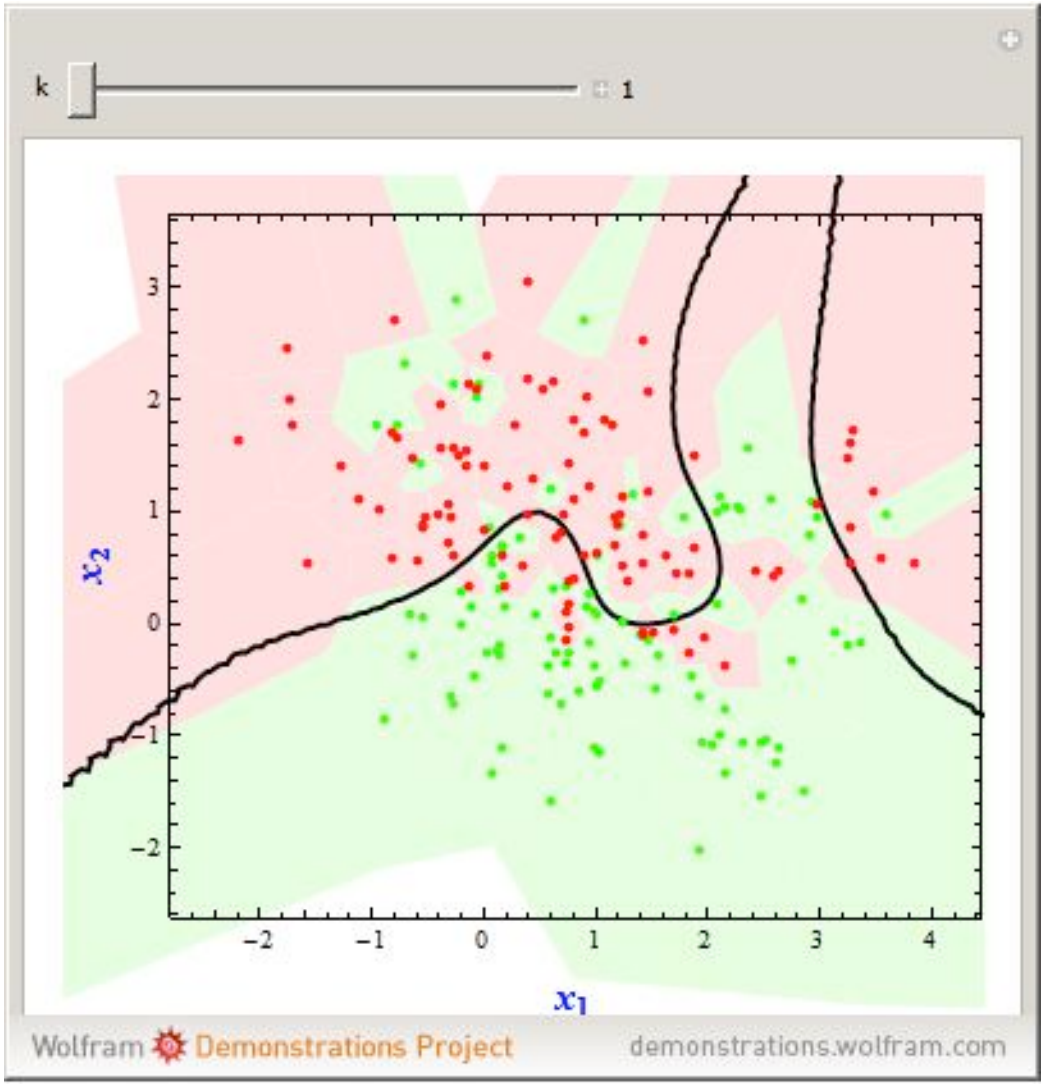


K-Nearest-Neighbours for Classification

- K acts as a smother
- As $N \rightarrow \infty$, the error rate of the 1-nearest-neighbour classifier is never more than twice the optimal error (obtained from the true conditional class distributions).



KNN Example


k
1
x_2
x_1
3
2
1
0
−1
−2
−2
−1
0
1
2
3
4
Wolfram Demonstrations Project
demonstrations.wolfram.com

Naïve Bayes Classifiers

- All of these nonparametric methods require lots of data to work. If $\mathcal{O}(N)$ training points are required for accurate estimation in 1 dimension, then $\mathcal{O}(N^D)$ points are required for *D*-dimensional input vectors.
- It may sometimes be possible to assume that the individual dimensions of the feature vector are conditionally independent. Then we have

$$p(\underline{x} \mid \omega_i) = \prod_{j=1}^{D} p(x_j \mid \omega_i)$$

- This reduces the data requirements to $\mathcal{O}(DN)$.

End of Lecture 8

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. **Training and Evaluation Methods**
11. What are Bayes Nets?

10. Training and Evaluation Methods

Machine Learning System Design

- The process of solving a particular classification or regression problem typically involves the following sequence of steps:
  1. **Design and code** promising candidate systems
  2. **Train** each of the candidate systems (i.e., learn the parameters)
  3. **Evaluate** each of the candidate systems
  4. **Select and deploy** the best of these candidate systems

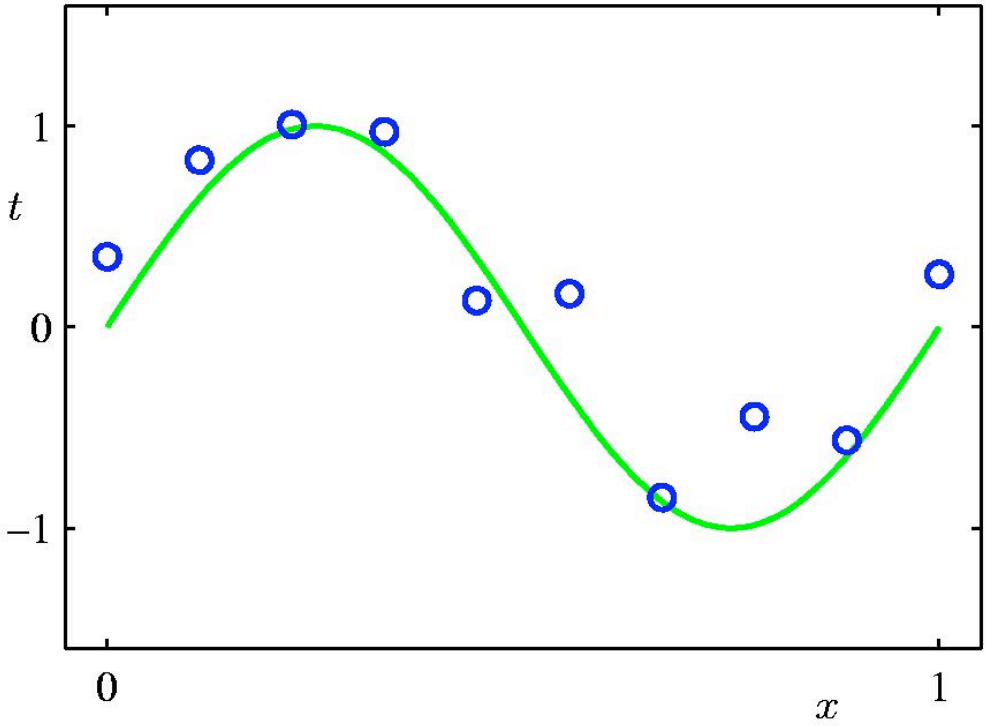
Using Your Training Data

- You will always have a finite amount of data on which to train and evaluate your systems.
- The performance of a classification system is often **data-limited**: if we only had more data, we could make the system better.
- Thus it is important to use your finite data set wisely.

Overfitting

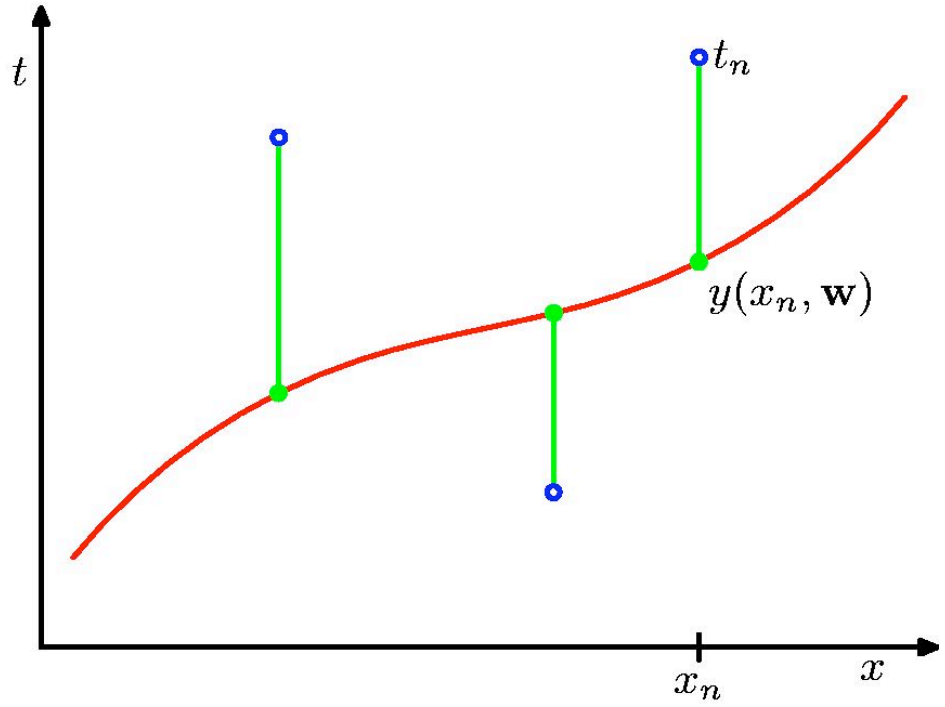
- Given that learning is often data-limited, it is tempting to use all of your data to estimate the parameters of your models, and then select the model with the lowest error on your training data.
- Unfortunately, this leads to a notorious problem called **over-fitting**.

Example: Polynomial Curve Fitting



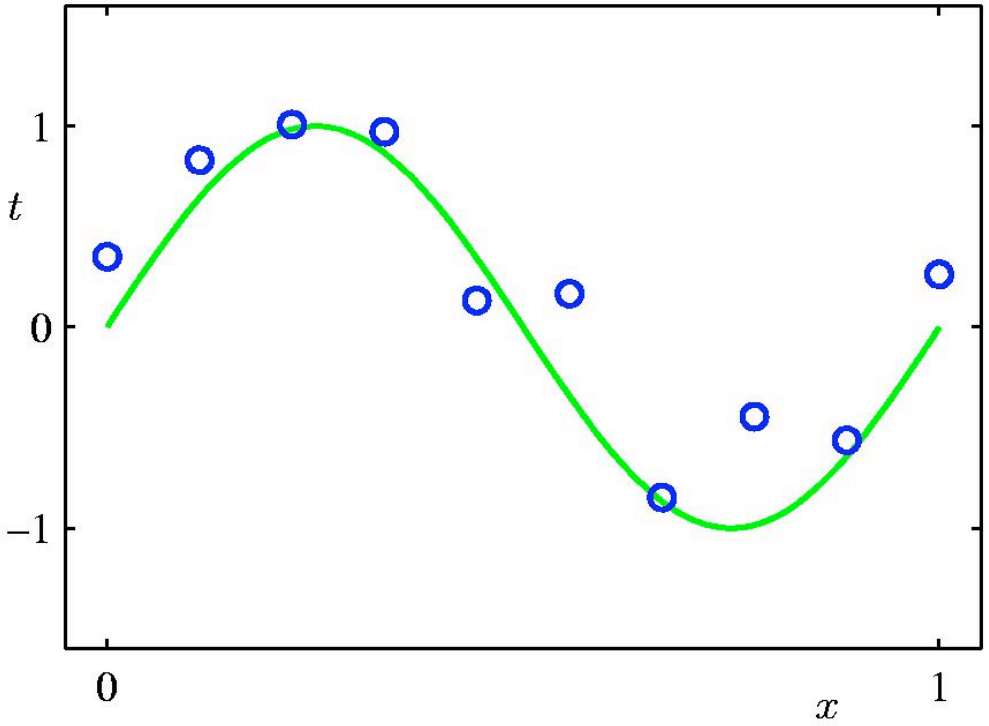
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sum-of-Squares Error Function



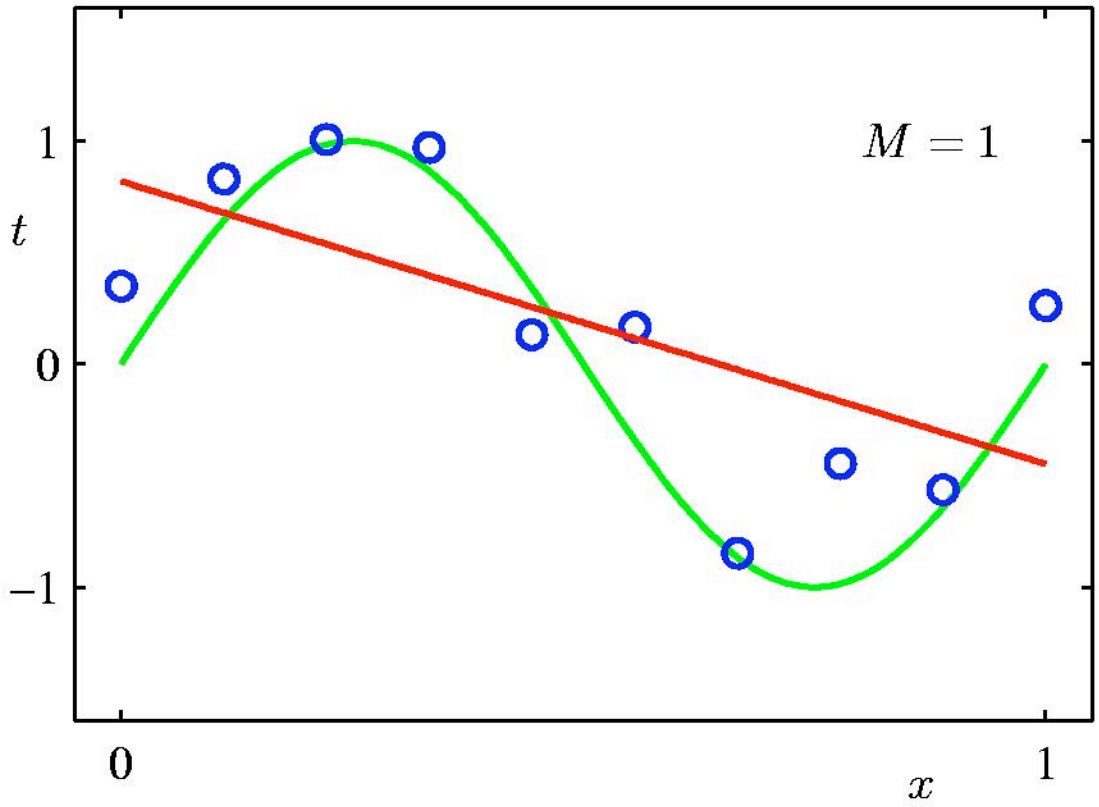
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

How do we choose M, the order of the model?

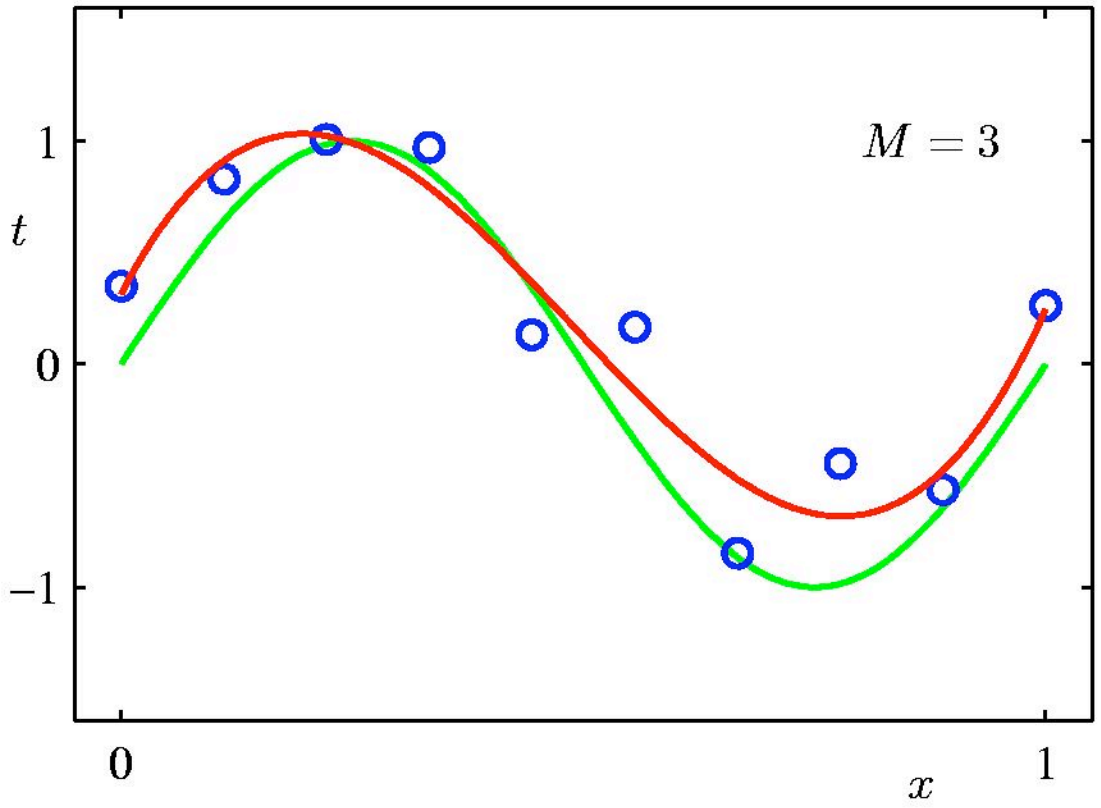


$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

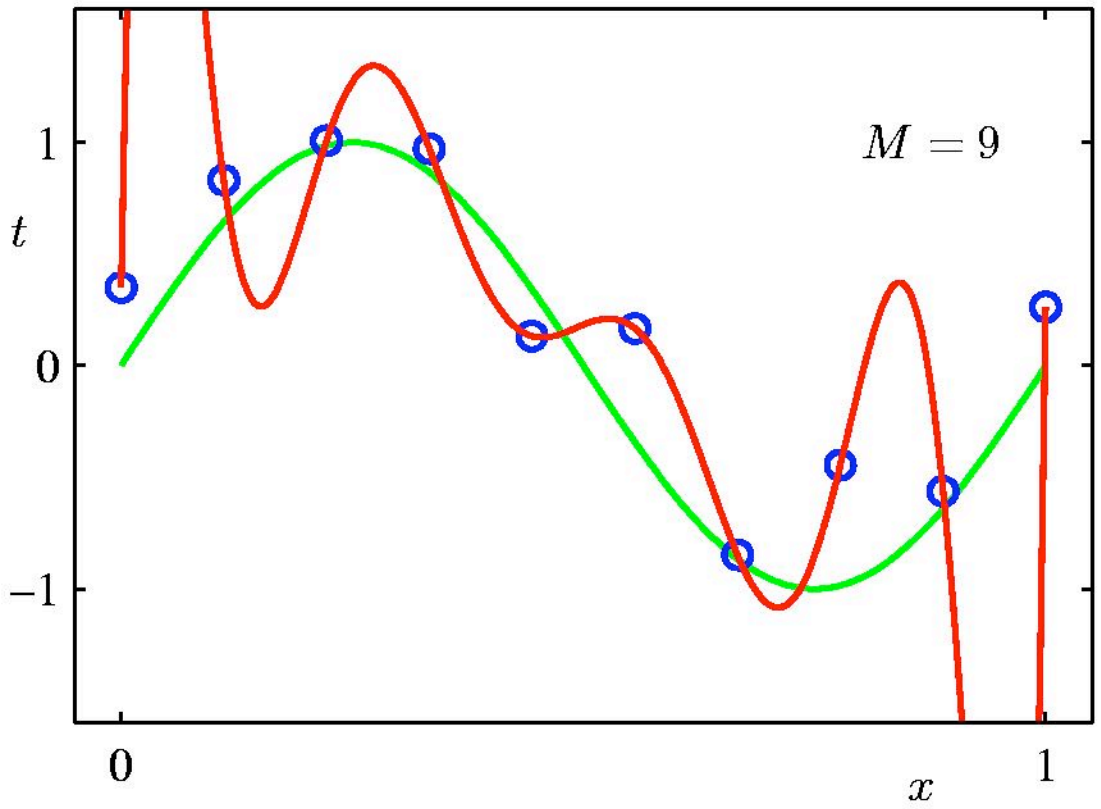
1st Order Polynomial



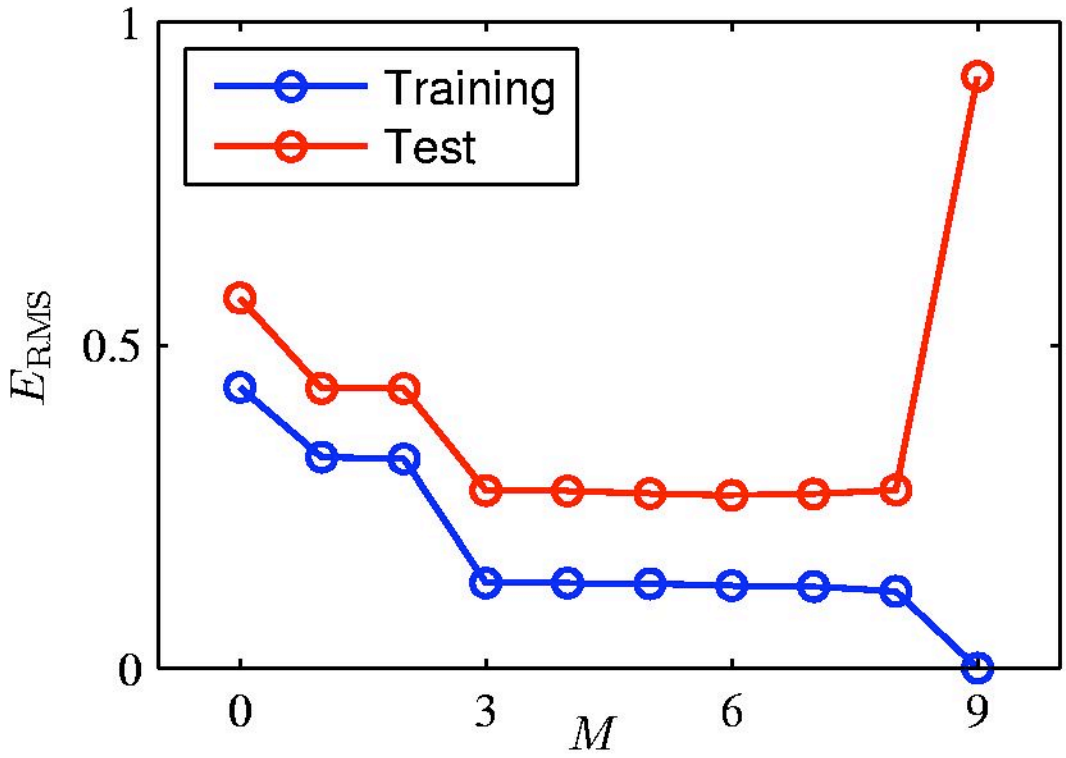
3rd Order Polynomial



9th Order Polynomial



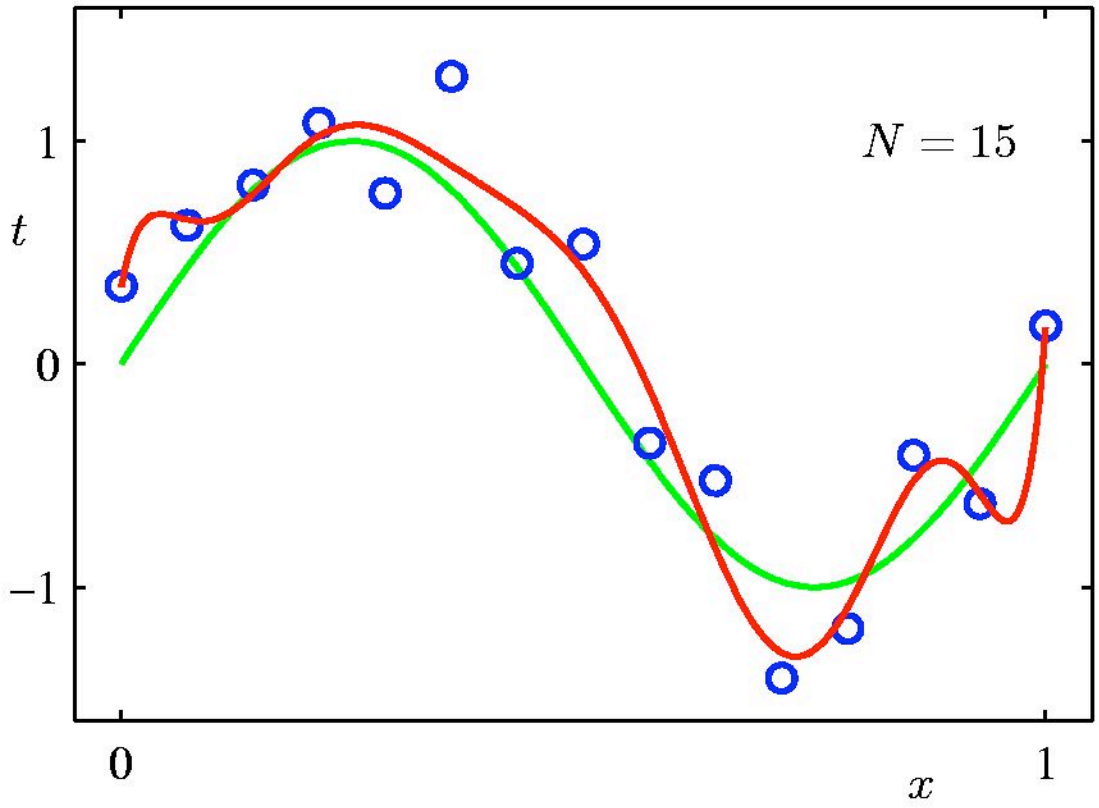
Over-fitting



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

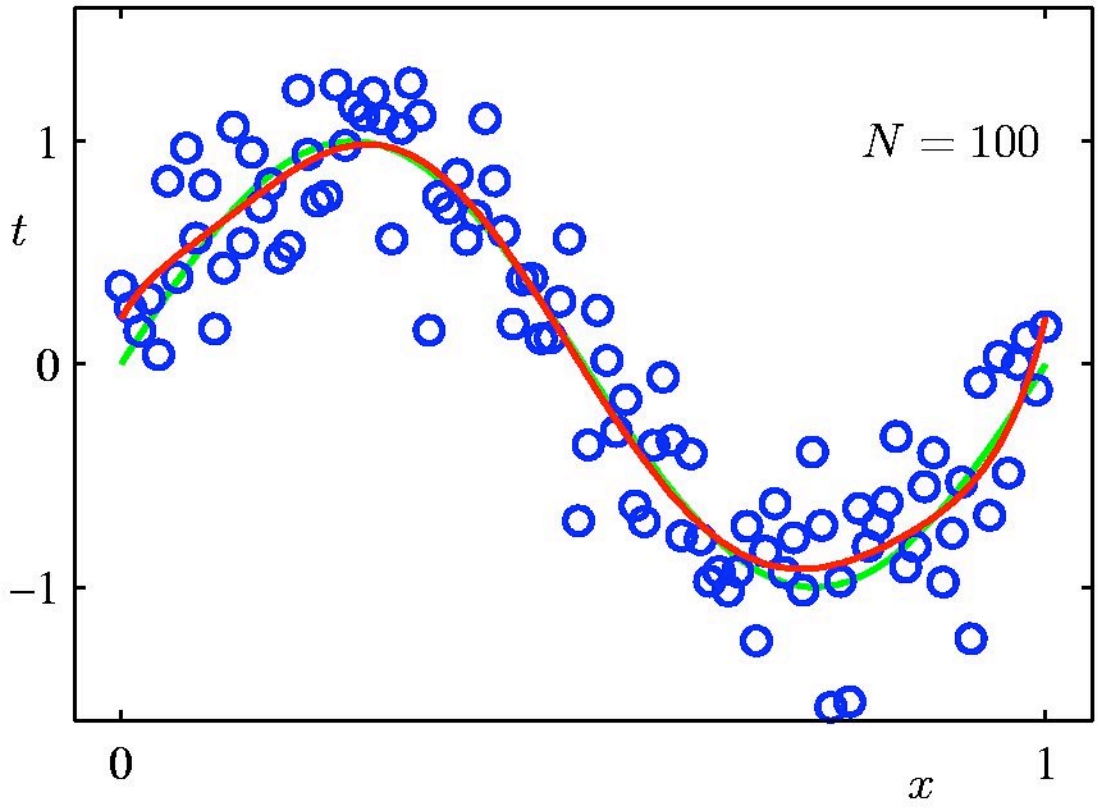
Overfitting and Sample Size

9th Order Polynomial



Over-fitting and Sample Size

9th Order Polynomial

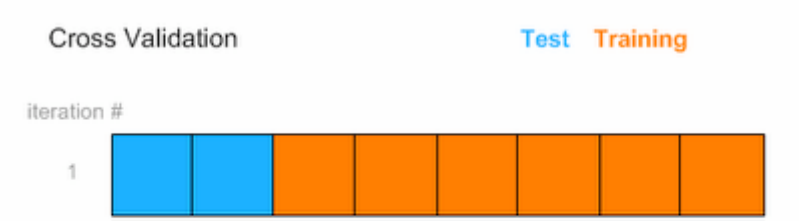


Methods for Preventing Over-Fitting

- Bayesian parameter estimation
  - Application of prior knowledge regarding the probable values of unknown parameters can often limit over-fitting of a model
- Model selection criteria
  - Methods exist for comparing models of differing complexity (i.e., with different types and numbers of parameters)
    - Bayesian Information Criterion (BIC)
    - Akaike Information Criterion (AIC)
- Cross-validation
  - This is a very simple method that is universally applicable.

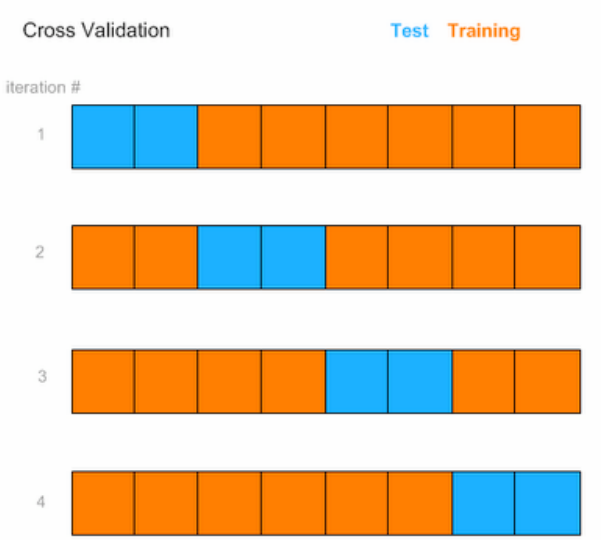
Cross-Validation

- The available data are partitioned into disjoint training and test subsets.
- Parameters are learned on the training sets.
- Performance of the model is then evaluated on the test set.
- Since the test set is independent of the training set, the evaluation is fair: models that overlearn the noise in the training set will perform poorly on the test set.



Cross-Validation: Choosing the Partition

- What is the best way to partition the data?
  - A larger training set will lead to more accurate parameter estimation.
  - However a small test set will lead to a noisy performance score.
  - If you can afford the computation time, repeat the training/test cycle on complementary partitions and then average the results. This gives you the best of all worlds: accurate parameter estimation and accurate evaluation.
  - In the limit: the **leave-one-out method**



Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. Training and Evaluation Methods
11. **What are Bayes Nets?**

10. What are Bayes Nets?

Directed Graphical Models and the Role of Causality

- Bayes nets are directed acyclic graphs in which each node represents a random variable.
- Arcs signify the existence of direct causal influences between linked variables.
- Strengths of influences are quantified by conditional probabilities

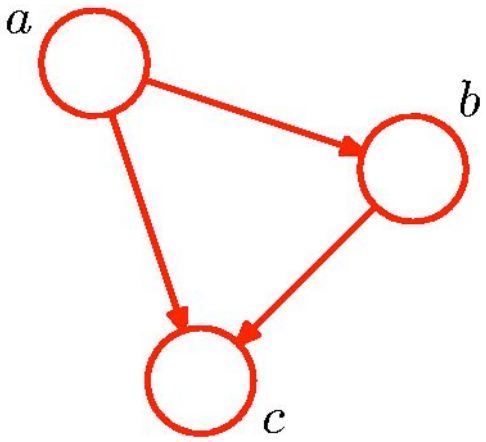
$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

where pa_k is the set of 'parent' nodes of node k.

- NB: For this to hold it is critical that the graph be acyclic.

Bayesian Networks

- Directed Acyclic Graph (DAG)



From the definition of conditional probabilities (product rule):

$$p(a,b,c) = p(c|a,b)p(a,b) = p(c|a,b)p(b|a)p(a)$$

In general:

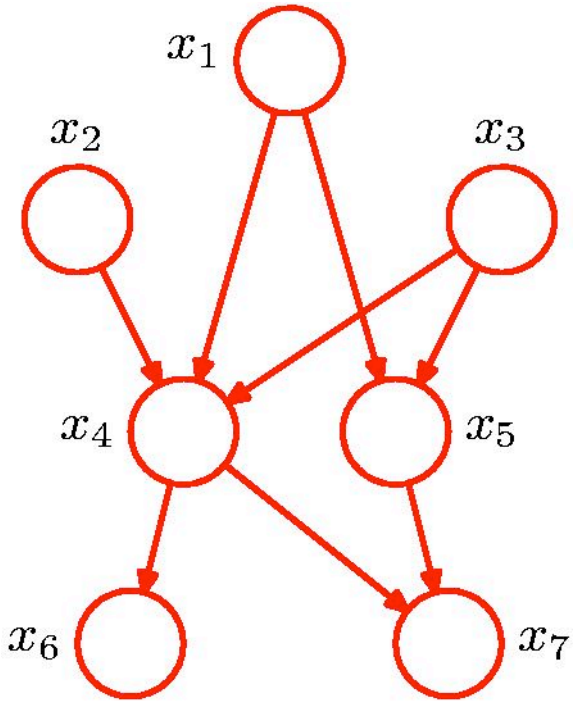
$$p(x_1,\ldots,x_K) = p(x_K|x_1,\ldots,x_{K-1})\ldots p(x_2|x_1)p(x_1)$$

This corresponds to a complete graph.

Bayesian Networks

- However, many systems have sparser causal relationships between their variables.

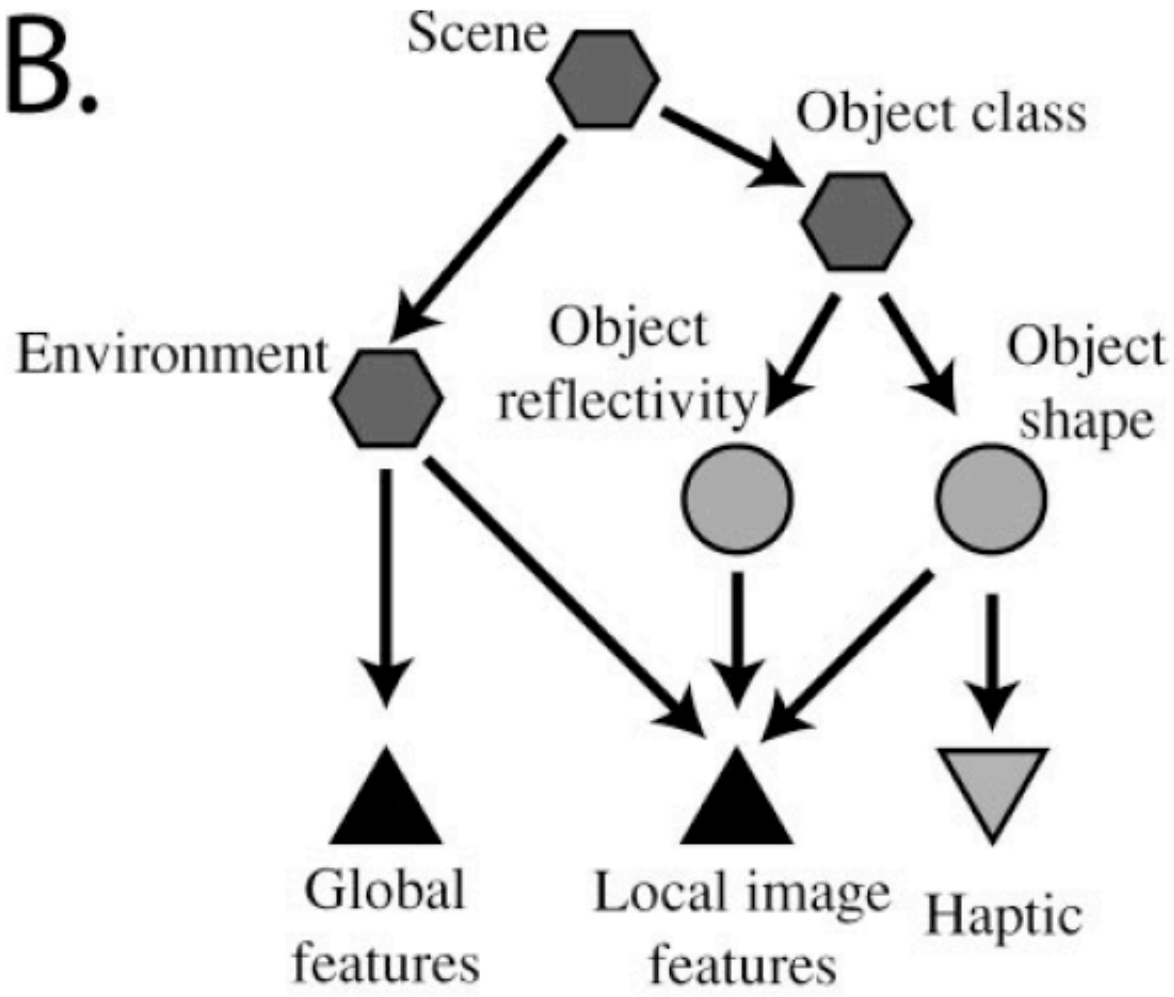
$$p(x_1,\ldots,x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)$$
$$p(x_5|x_1,x_3)p(x_6|x_4)p(x_7|x_4,x_5)$$



General Factorization

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

Generative Models of Perception



Discrete Variables

- General joint distribution: $K^2 - 1$ parameters

$$p(\mathbf{x}_1, \mathbf{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K} \prod_{l=1}^{K} \mu_{kl}^{x_{1k} x_{2l}}$$

- Independent joint distribution: $2(K - 1)$ parameters

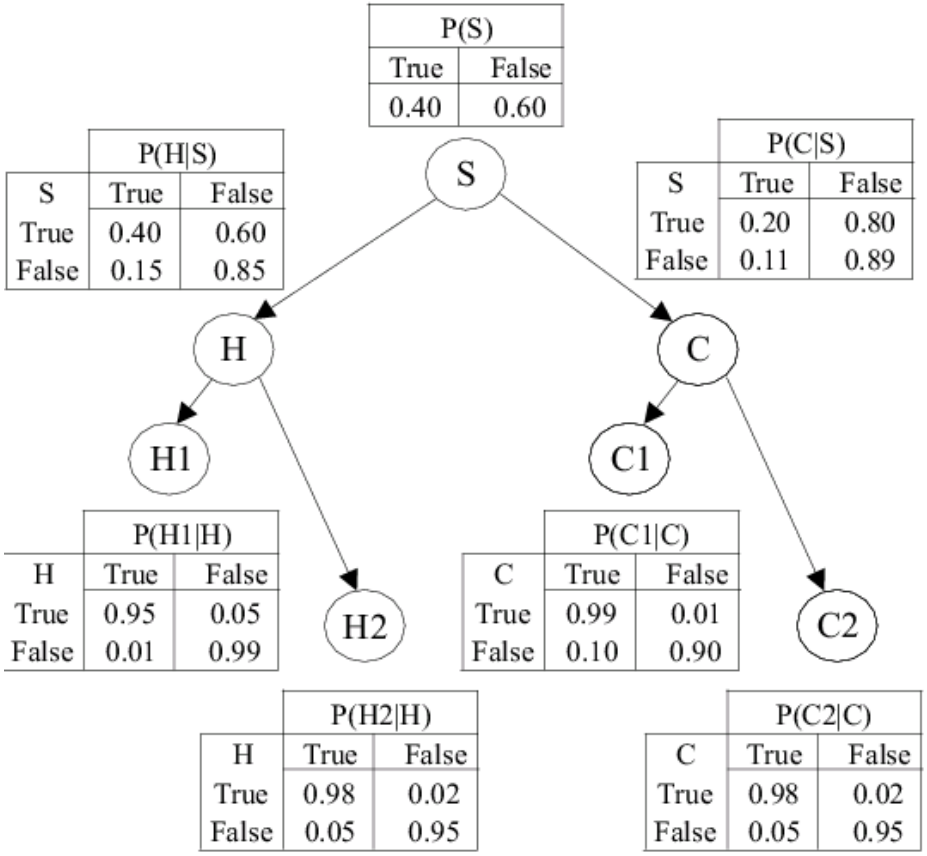
$$\hat{p}(\mathbf{x}_1, \mathbf{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_{1k}^{x_{1k}} \prod_{l=1}^{K} \mu_{2l}^{x_{2l}}$$

Discrete Variables

- General distributions require many parameters.
- General joint distribution over M variables: $K^M - 1$ parameters
- It is thus extremely important to identify structure in the system that corresponds to a sparser graphical model and hence fewer parameters.

Binary Variable Example

- S: Smoker?
- C: Cancer?
- H: Heart Disease?
- (H1, H2): Results of medical tests for heart disease
- (C1, C2): Results of medical tests for cancer



Discrete Variables

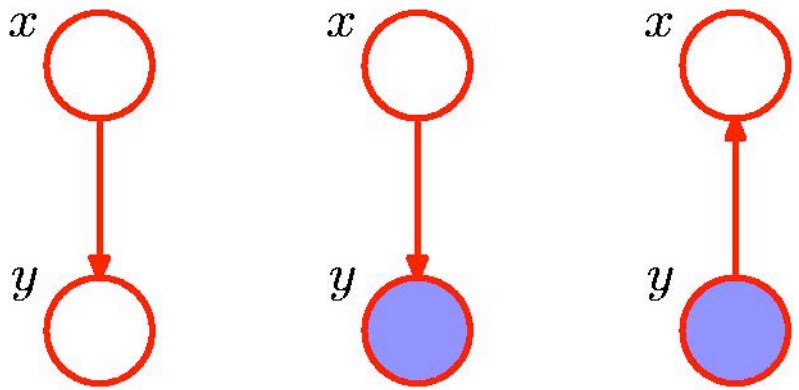
- Example: M -node Markov chain
  - K -1 + (M -1) K(K -1) parameters

Using Bayes Nets

- Once a DAG has been constructed, the joint probability can be obtained by multiplying the marginal (root nodes) and the conditional (non-root nodes) probabilities.
- Training: Once a topology is given, probabilities are estimated via the training data set. There are also methods that learn the topology.
- Probability Inference: This is the most common task that Bayesian networks help us to solve efficiently. Given the values of some of the variables in the graph, known as evidence, the goal is to compute the conditional probabilities for some of the other variables, given the evidence.

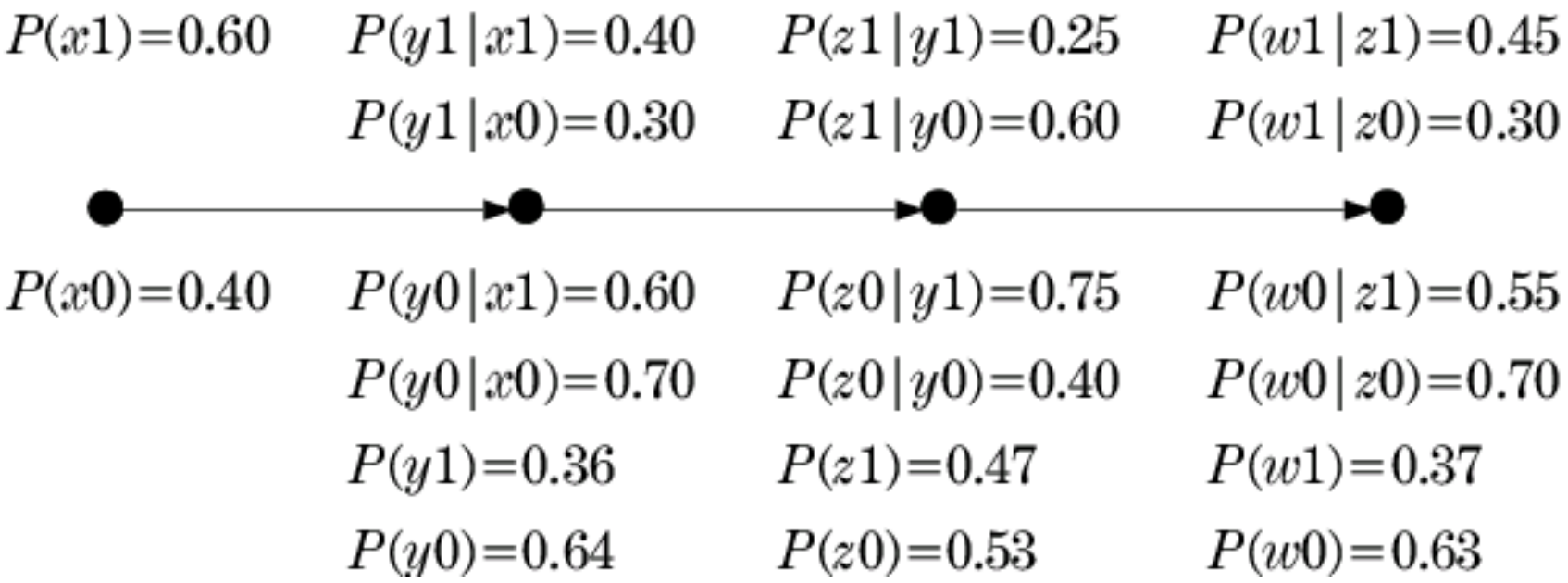
Inference in Bayes Nets

- In inference, we clamp some of the variables to observed values, and then compute the posterior over other, unobserved variables.
- Simple example:



$$p(y) = \sum_{x'} p(y|x')p(x') \qquad p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Example



a) Suppose *x* has been measured and its value is 1. What is the probability that *w* is 0?

b) Suppose *w* is measured and its value is 1. What is the probability that *x* is 0?

Message Passing

- For a), computation propagates from node x to node w, resulting in $P(w0|x1) = 0.63$.
- For b), computation propagates in the opposite direction, resulting in $P(x0|w1) = 0.4$.
- In general, the required inference information is computed via a combined process of "message passing" among the nodes of the DAG.

Complexity:

- For singly connected graphs, message passing algorithms amount to a complexity linear in the number of nodes.

Bayesian Decision Theory: Topics

1. Probability
2. The Univariate Normal Distribution
3. Bayesian Classifiers
4. Minimizing Risk
5. The Multivariate Normal Distribution
6. Decision Boundaries in Higher Dimensions
7. Parameter Estimation
8. Mixture Models and EM
9. Nonparametric Density Estimation
10. What are Bayes Nets?