Outline

- Review
- Maximum A-Posteriori (MAP) Estimation
- Bayesian Parameter Estimation
- Example:The Gaussian Case
- Recursive Bayesian Incremental Learning
- Problems of Dimensionality
- Linear Algebra review
- Principal Component Analysis
- Fisher Discriminant

Bayesian Decision Theory

- Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification.
  - ➢ Decision making when all the probabilistic information is known.
  - ➢ For given probabilities the decision is optimal.
  - ➢ When new information is added, it is assimilated in optimal fashion for improvement of decisions.

Bayes' formula

$$\boldsymbol{P(\omega_j \,|\, x) = P(x \,|\omega_j)\; P(\omega_j)\,/\,P(x),}$$

where

$$P(x) = \sum_{j=1}^{2} p(x \,|\, \omega_j) P(\omega_j)$$

$$\text{Posterior} = \frac{\text{Likelihood} \,*\, \text{Prior}}{\text{Evidence}}$$

Bayes' formula cont.

- $p(x|\boldsymbol{\omega}_j)$ is called the ***likelihood*** of $\boldsymbol{\omega}_j$ with respect to ***x.***

(the $\boldsymbol{\omega}_j$ category for which $p(x|\boldsymbol{\omega}_j)$ is large

is more "likely" to be the true category)

- p(x) is the ***evidence***

how frequently we will measure a pattern with feature value x.

Scale factor that guarantees that the posterior probabilities sum to 1.

Bayes' Decision Rule
(Minimizes the probability of error)

$$\omega_1 : \textit{if } P(\omega_1|x) > P(\omega_2|x)$$
$$\omega_2 : \textit{otherwise}$$

or

$$\omega_1 : \textit{if } P(x|\omega_1)\, P(\omega_1) > P(x|\omega_2)\, P(\omega_2)$$
$$\omega_2 : \textit{otherwise}$$

and

$$P(Error|x) = \min\,[P(\omega_1|x)\,,\, P(\omega_2|x)]$$

Normal Density - Univariate Case

- Gaussian density with mean $\mu \in$ ° and standard deviation $\sigma \in$ ° $_+$, (σ^2 named variance)

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left[-\frac{1}{2}\left(\frac{\mathbf{x}-\boldsymbol{\mu}}{\sigma}\right)^2\right]$$

$$p(x) \sim N(\mu, \sigma^2)$$

- It can be shown that:

$$\mu = \mathbf{E}[x] = \int_{-\infty}^{\infty} x p(x) dx, \qquad \sigma^2 = \mathbf{E}[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 p(x) dx.$$

Normal Density - Multivariate Case

- The general ***multivariate normal density*** (MND) in a d dimensions is written as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\,|\,\Sigma\,|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t\,\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

- It can be shown that:

$$\boldsymbol{\mu} = \mathbf{E}[\mathbf{x}] = \int_{\circ\ d} \mathbf{x}\; p(\mathbf{x})d\mathbf{x}, \qquad \boldsymbol{\Sigma} = \mathbf{E}[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^t] \; .$$

which means for components

$$\sigma_{i\,j} = \mathbf{E}[(x_i - \mu_i)(x_j - \mu_j)] \; .$$

- The covariance matrix $\boldsymbol{\Sigma}$ is always symmetric and positive semidefinite.

Normal Density - Multivariate Case

- The general ***multivariate normal density*** (MND) in a *d* dimensions is written as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

- It can be shown that:

$$\boldsymbol{\mu} = \mathbf{E}[\mathbf{x}] = \int_{\mathrm{R}^d} \mathbf{x}\, p(\mathbf{x}) d\mathbf{x}, \qquad \boldsymbol{\Sigma} = \mathbf{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^t] \;\; .$$

which means for components

$$\sigma_{i\,j} = \mathbf{E}[(x_i - \mu_i)(x_j - \mu_j)] \;\; .$$

Maximum Likelihood and Bayesian Parameter Estimation

- To design an optimal classifier we need $P(\omega_i)$ and $p(x|\omega_i)$, but usually we do not know them.

- Solution – to use training data to estimate the unknown probabilities. Estimation of class-conditional densities is a difficult task.

Maximum Likelihood and Bayesian Parameter Estimation

- Supervised learning: we get to see samples from each of the classes "separately" (called tagged or labeled samples).

- Tagged samples are "expensive". We need to learn the distributions as efficiently as possible.

- Two methods: parametric (easier) and non-parametric (harder)

Maximum Likelihood and Bayesian Parameter Estimation

- Program for parametric methods:
  - Assume specific parametric distributions with parameters $\theta \in \Theta \subset R^p$
  - Estimate parameters $\hat{\theta}(\mathrm{D})$ from training data D.
  - Replace true value of class-conditional density with approximation and apply the Bayesian framework for decision making.

Maximum Likelihood and Bayesian Parameter Estimation

- Suppose we can assume that the relevant (class-conditional) densities are of some parametric form. That is,

  $p(x|\omega)=p(x|\theta)$, where $\theta \in \Theta \subset \mathrm{R}^p$

- Examples of parameterized densities:
  - Binomial: $x^{(n)}$ has m 1's and n-m 0's

$$p(x^{(n)} \mid \theta) = \binom{n}{m} \theta^m (1-\theta)^{n-m}, \qquad \Theta = [0,1]$$

  - Exponential: Each data point x is distributed according to

$$p(x \mid \theta) = \theta e^{-\theta x}, \qquad \Theta = (0, \infty)$$

Maximum Likelihood and Bayesian Parameter Estimation cont.

- Two procedures for parameter estimation will be considered:
  - Maximum likelihood estimation: choose parameter value $\hat{\theta}$ that makes the data most probable (i.e., maximizes the probability of obtaining the sample that has actually been observed),

$$p(\mathbf{x} \mid \mathrm{D}) = p(\mathbf{x} \mid \hat{\theta}(\mathrm{D})), \qquad \hat{\theta}(\mathrm{D}) = \arg\max_{\theta} p(\mathrm{D} \mid \theta)$$

  - Bayesian learning: define a prior probability on the model space $p(\theta)$ and compute the posterior $p(\theta \mid \mathrm{D})$.
    Additional samples sharp the posterior density which peaks near the true values of the parameters .

Sampling Model

- It is assumed that a sample set $S = \{(\mathbf{x}_l, \omega_l) : l = 1, \ldots, N\}$ with independently generated samples is available.
- The sample set is partitioned into separate sample sets for each class, $\mathrm{D}_\mathrm{j} = \{\mathbf{x}_l : (\mathbf{x}_l, \omega_l) \in \mathrm{D}\}$
- A generic sample set will simply be denoted by D .
- Each class-conditional $p(\mathbf{x} | \omega_j)$ is assumed to have a known parametric form and is uniquely specified by a parameter (vector) $\boldsymbol{\theta}_j$.
- Samples in each set D_j are assumed to be independent and identically distributed (i.i.d.) according to some true probability law $p(\mathbf{x} | \omega_j)$.

Log-Likelihood function and Score Function

- The sample sets are assumed to be functionally independent, i.e., the training set s_j contains no information about $\boldsymbol{\theta}_i$ for $i \neq j$.
- The i.i.d. assumption implies that

$$p(\mathrm{D_j}|\boldsymbol{\theta}_j) = \prod_{x \in \mathrm{D}_j} p(\mathbf{x} \,|\, \boldsymbol{\theta}_j)$$

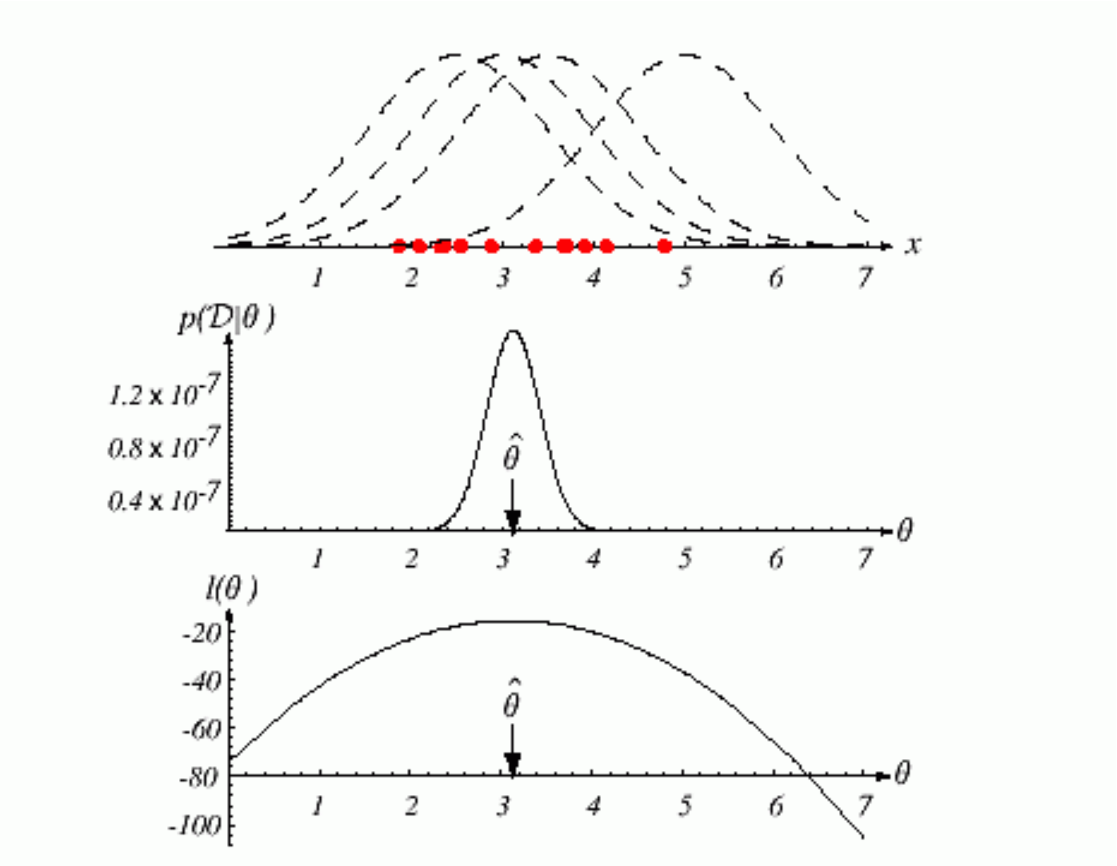
- Let D be a generic sample of size $n \equiv |\mathrm{D}|$.
- Log-likelihood function:

$$l(\boldsymbol{\theta}; \mathrm{D}) \equiv \ln p(\mathrm{D}|\theta) = \sum_{k=1}^{n} \ln p(\mathbf{x}_k \,|\, \boldsymbol{\theta})$$

- The log-likelihood function is identical to the logarithm of the probability density function, but is interpreted as a function over the sample space for given parameter $\boldsymbol{\theta}$.

Log-Likelihood Illustration

- Assume that all the points in $\mathcal{D}$ are drawn from some (one-dimensional) normal distribution with some (known) variance and unknown mean.



Log-Likelihood function and Score Function cont.

- Maximum likelihood estimator (MLE):

$$\hat{\boldsymbol{\theta}}(\mathrm{D}) = \underbrace{\arg\max}_{\theta\in\Theta} l(\boldsymbol{\theta};\mathrm{D})$$

(tacitly assuming that such a maximum exists!)

- Score function:

$$U_k(\boldsymbol{\theta};\mathrm{D}) \equiv \frac{\partial l(\boldsymbol{\theta};\mathrm{D})}{\partial \theta_k} \qquad 1 \le k \le p$$

and hence

$$\mathbf{U}(\boldsymbol{\theta};\mathrm{D}) \equiv \nabla_\theta l(\boldsymbol{\theta};\mathrm{D})$$

- Necessary condition for MLE (if not on border of domain Θ) :

$$\mathbf{U}(\boldsymbol{\theta};\mathrm{D}) = 0$$

Maximum *A Posteriory*

- Maximum a posteriory (MAP):

  Find the value of θ that maximizes $l(\theta)+ln(p(\theta))$, where $p(\theta)$, is a prior probability of different parameter values.A MAP estimator finds the peak or *mode* of a posterior.

  Drawback of MAP: after arbitrary nonlinear transformation of the parameter space, the density will change, and the MAP solution will no longer be correct.

Maximum A-Posteriori (MAP) Estimation

- The "most likely value" is given by θ

$$\hat{\theta} = \arg\max_{\theta} p(\theta \mid X^{(n)}) = \arg\max_{\theta} \frac{p_0(\theta)\, p(X^{(n)} \mid \theta)}{p(X^{(n)})}$$

$$= \arg\max_{\theta} \frac{p_0(\theta) \prod_{i=1}^{n} p(x_i \mid \theta)}{\int p(X^{(n)} \mid \theta')\, p_0(\theta')\, d\theta'}$$

Maximum A-Posteriori (MAP) Estimation

$$p(X^{(n)} \mid \theta) = \prod_{i=1}^{n} p(x_i \mid \theta)$$

since the data is i.i.d.

- We can disregard the normalizing factor $p(X^{(n)})$ when looking for the maximum

MAP - continued

So, the $\hat{\theta}$ we are looking for is

$$\hat{\theta} = \arg\max_{\theta} \left[p_0(\theta) \prod_{i=1}^{n} p(x_i \mid \theta)] \right]$$

(log is monotonically increasing)

$$= \arg\max_{\theta} \left(\log \left[p_0(\theta) \prod_{i=1}^{n} p(x_i \mid \theta)] \right] \right)$$

$$= \arg\max_{\theta} \left(\log p_0(\theta) + \log \prod_{i=1}^{n} p(x_i \mid \theta) \right)$$

$$= \arg\max_{\theta} \left(\log p_0(\theta) + \sum_{i=1}^{n} \log p(x_i \mid \theta) \right)$$

The Gaussian Case: Unknown Mean

- Suppose that the samples are drawn from a multivariate normal population with mean μ , and covariance matrix Σ .
- Consider fist the case where only the mean is unknown $\theta = \mu$.
- For a sample point x_k , we have

$$\ln P(\mathbf{x}_k \mid \boldsymbol{\mu}) = -\frac{1}{2}\ln\left[(2\pi)^d \mid \Sigma \mid\right] - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^t \Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

and

$$\nabla_\mu \ln P(\mathbf{x}_k \mid \boldsymbol{\mu}) = \Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

- The maximum likelihood estimate for μ must satisfy

The Gaussian Case: Unknown Mean

$$\sum_{k=1}^{n} \Sigma^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0$$

- Multiplying by Σ , and rearranging, we obtain

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$$

- The MLE estimate for the unknown population mean is just the arithmetic average of the training samples (sample mean).
- Geometrically, if we think of the n samples as a cloud of points, the sample mean is the centroid of the cloud

The Gaussian Case: Unknown Mean and Covariance

- In the general multivariate normal case, neither the mean nor the covariance matrix is known $\theta = [\boldsymbol{\mu}, \Sigma]$.
- Consider fist the univariate case with $\theta_1 = \mu$ and $\theta_2 = \sigma^2$. The log-likelihood of a single point is

$$\ln p(\mathbf{x}_k \mid \boldsymbol{\theta}) = -\frac{1}{2}\ln 2\pi\theta_2 - \frac{1}{2\theta_2}(\mathbf{x}_k - \theta_1)^2$$

and its derivative is

$$\nabla_{\boldsymbol{\theta}} l = \nabla_{\boldsymbol{\theta}} \ln p(x_k \mid \boldsymbol{\theta}) = \begin{bmatrix} \dfrac{1}{\theta_2}(x_k - \theta_1) \\ -\dfrac{1}{2\theta_2} + \dfrac{(x_k - \theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

The Gaussian Case: Unknown Mean and Covariance

- Setting the gradient to zero, and using all the sample points, we get the following necessary conditions:

$$\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}(x_k - \hat{\theta}_1) = 0 \quad and \quad -\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0$$

- where $\hat{\theta}_1 = \hat{\mu}$ and $\hat{\theta}_2 = \hat{\sigma}^2$, are the MLE estimates for $\hat{\theta}_1$, and $\hat{\theta}_2$ respectively.
- Solving for $\hat{\mu}$ and $\hat{\sigma}^2$, we obtain

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k \text{ and } \hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})^2$$

The Gaussian multivariate case

- For the multivariate case, it is easy to show that the MLE estimates for are given by

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k \quad \text{and} \quad \hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n} (\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t$$

- The MLE for the mean vector is the sample mean, and the MLE estimate for the covariance matrix is the arithmetic average of the n matrices $(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t$
- The MLE for σ^2 is biased (i.e., the expected value over all data sets of size n of the sample variance is not equal to the true variance:

$$E\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2\right] = \frac{n-1}{n}\sigma^2 \neq \sigma^2$$

The Gaussian multivariate case

- Unbiased estimator for μ and Σ are given by

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k$$

and

$$C = \frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t$$

C is called the sample covariance matrix . C is *absolutely unbiased*. $\hat{\sigma}^2$ is *asymptotically unbiased*.

Bayesian Estimation: Class-Conditional Densities

- The aim is to find posteriors P($\omega_i|\mathbf{x}$) knowing p($\mathbf{x}|\omega_i$) and P(ω_i), but they are unknown. How to find them?
- Given the sample D, we say that the aim is to find P($\omega_i|\mathbf{x}, D$)
- Bayes formula gives:

$$P(\omega_i \mid \mathbf{x}, D) = \frac{p(\mathbf{x} \mid \omega_i, D)P(\omega_i \mid D)}{\sum_{j=1}^{c} p(\mathbf{x} \mid \omega_i, D)P(\omega_i \mid D)}.$$

- We use the information provided by training samples to determine the class conditional densities and the prior probabilities.
- Generally used assumptions:
  - Priors generally are known or obtainable from a trivial calculations. Thus P(ω_i)= P($\omega_i|D$).
  - The training set can be separated into c subsets: $D_1, \ldots, D_c$

Bayesian Estimation: Class-Conditional Densities

  - The samples D_j have no influence on p($\mathbf{x}|\omega_{i,}D_i$) if $i \neq j$

- Thus we can write:

$$P(\omega_i \mid \mathbf{x}, D) = \frac{p(\mathbf{x} \mid \omega_i, D_i) P(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x} \mid \omega_j, D_j) P(\omega_j)}.$$

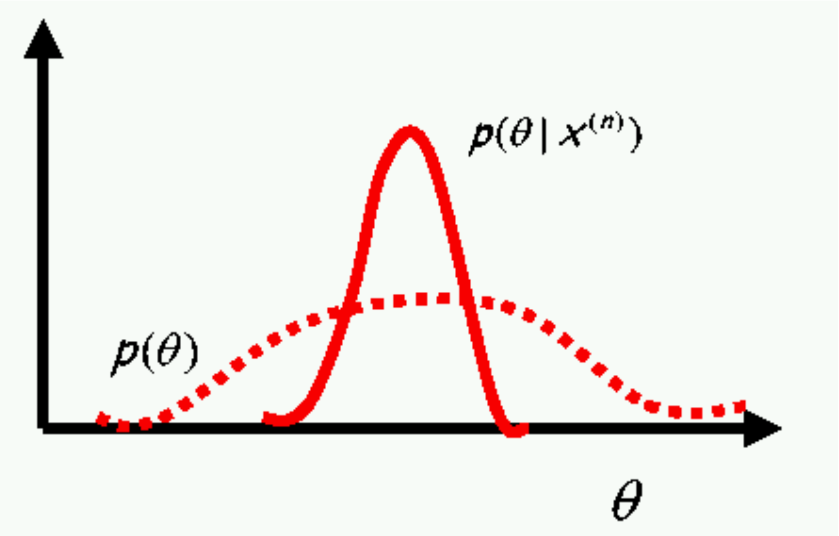
- We have *c* separate problems of the form:

Use a set *D* of samples drawn independently according to a fixed but unknown probability distribution *p(**x**)* to determine *p(**x**|D)*.

Bayesian Estimation: General Theory

- *Bayesian leaning* considers θ (the parameter vector to be estimated) to be a ***random variable***.

Before we observe the data, the parameters are described by a *prior* $p(\theta)$ which is typically very broad. Once we observed the data, we can make use of Bayes' formula to find *posterior* $p(\theta|D)$. Since some values of the parameters are more consistent with the data than others, the *posterior* is narrower than *prior*. This is *Bayesian learning* (see fig.)



General Theory cont.

- Density function for ***x***, given the training data set D ,

$$p(\mathbf{x} \mid \mathrm{D}) = \int p(\mathbf{x}, \theta \mid \mathrm{D}) d\theta$$

- From the definition of conditional probability densities

$$p(\mathbf{x}, \theta \mid \mathrm{D}) = p(\mathbf{x} \mid \theta, \mathrm{D}) p(\theta \mid \mathrm{D}).$$

- The first factor is independent of D since it just our assumed form $p(\mathbf{x} \mid \theta, D\ ) \Rightarrow p(\mathbf{x} \mid \theta)$ for parameterized density.
- Therefore

$$p(\mathbf{x} \mid \mathrm{D}) = \int p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathrm{D}) d\boldsymbol{\theta}$$

- Instead of choosing a specific value for θ , the Bayesian approach performs a weighted average over all values of θ.

  The weighting factor $p(\theta \mid \mathrm{D})$, which is a posterior of θ is determined by starting from some assumed prior $p(\theta)$

General Theory cont.

- Then update it using Bayes' formula to take account of data set D. Since $\mathrm{D}=\{\mathbf{x}^1,\ldots,\mathbf{x}^N\}$ are drawn independently

$$p(\mathrm{D}\,|\,\theta)=\prod_{n=1}^{N} p(x^n\,|\,\theta)\ , \qquad (*)$$

which is likelihood function.

- Posterior for θ is

$$p(\theta\,|\,\mathrm{D})=\frac{p(\mathrm{D}\,|\theta)p(\theta)}{p(\mathrm{D})}=\frac{p(\theta)}{p(\mathrm{D})}\prod_{n=1}^{N} p(x^n\,|\theta)\,, \qquad (**)$$

where normalization factor

$$p(\mathrm{D})=\int p(\theta')\prod_{n=1}^{N} p(x^n\,|\theta')d\theta',$$

Bayesian Learning – Univariate Normal Distribution

- Let us use the Bayesian estimation technique to calculate *a posteriori* density $p(\boldsymbol{\theta} \mid \mathrm{D})$ and the desired probability density $p(\mathbf{x} \mid \mathrm{D})$ for the case $p(\mathbf{x} \mid \boldsymbol{\mu}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
  - Univariate Case: $p(\mu \mid \mathrm{D})$

    Let ***μ*** ***be the only unknown*** parameter

$$p(x \mid \mu) \sim N(\mu, \sigma^2)$$

Bayesian Learning – Univariate Normal Distribution

- Prior probability: normal distribution over μ ,

$$p(\mu) \sim N(\mu_0, \sigma_0^2)$$

μ_0 encodes some prior knowledge about the true mean μ , while σ_0^2 measures our prior uncertainty.

- If μ is drawn from *p(μ)* then density for x is completely determined. Letting $\mathrm{D} = \{x_1, \ldots, x_n\}$ we use

$$p(\mu \mid \mathrm{D}) = \frac{p(\mathrm{D} \mid \mu) p(\mu)}{\int p(\mathrm{D} \mid \mu) p(\mu) d\mu}$$

$$= \alpha \prod_{k=1}^{n} p(x_k \mid \mu) p(\mu)$$

Bayesian Learning – Univariate Normal Distribution

- Computing the posterior distribution

$$p(\mu \mid \mathrm{D}) \propto p(\mathrm{D} \mid \mu)\, p(\mu)$$

$$= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{x_k-\mu}{\sigma}\right)^2 + \left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right)\right]$$

$$= \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n}x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right]$$

Bayesian Learning – Univariate Normal Distribution

- Where factors that do not depend on μ have been absorbed into the constants α' and α''
- $p(\mu \mid \mathrm{D})$ is an exponential function of a quadratic function of μ i.e. it is a normal density.
- $p(\mu \mid \mathrm{D})$ remains normal for any number of training samples.
- If we write

$$p(\mu \mid \mathrm{D}) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_n}{\sigma_n}\right)^2\right]$$

then identifying the coefficients, we get

Bayesian Learning – Univariate Normal Distribution

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \qquad \frac{\mu_n}{\sigma_n^2} = \frac{n}{\sigma^2}\hat{\mu}_n + \frac{\mu_0}{\sigma_0^2}$$

where $\hat{\mu}_n = \frac{1}{n}\sum_{k=1}^{n} x_k$ is the sample mean.

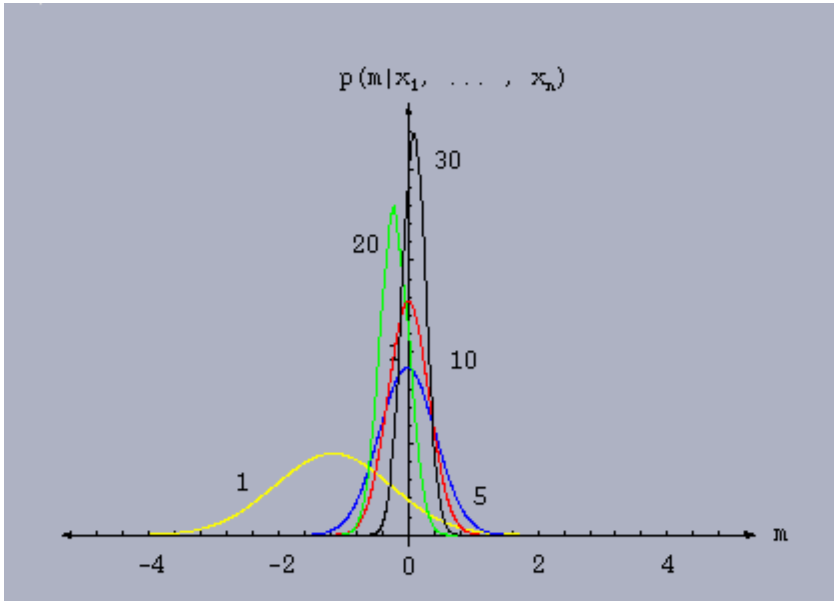
- Solving explicitly for μ_n and σ_n^2 we obtain

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2}\right)\hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2}\mu_0 \quad \text{and} \quad \sigma_n^2 = \frac{\sigma_0^2\sigma^2}{n\sigma_0^2 + \sigma^2}$$

- μ_n represents our best guess for μ after observing *n* samples.
- σ_n^2 measures our uncertainty about this guess.
- σ_n^2decreases monotonically with n (approaching σ^2 / n as *n* approaches infinity)

Bayesian Learning – Univariate Normal Distribution

- Each additional observation decreases our uncertainty about the true value of μ.
- As n increases, $p(\mu \mid \mathrm{D})$ becomes more and more sharply peaked, approaching a Dirac delta function as n approaches infinity. This behavior is known as *Bayesian Learning.*



Bayesian Learning – Univariate Normal Distribution

- In general, μ_n is a linear combination of $\hat{\mu}_n$ and μ_0 , with coefficients that are non-negative and sum to 1.
- Thus μ_n lies somewhere between $\hat{\mu}_n$ and μ_0 .
- If $\sigma_0 \neq 0$, $\mu_n \rightarrow \hat{\mu}_n$ as $n \rightarrow \infty$
- If $\sigma_0 = 0$, our a priori certainty that $\mu = \mu_0$ is so strong that no number of observations can change our opinion.
- If σ_0 ? σ, a priori guess is very uncertain, and we take $\mu_n = \hat{\mu}_n$
- The ratio σ^2 / σ_0^2 is called *dogmatism*.

Bayesian Learning – Univariate Normal Distribution

- The Univariate Case: $p(x \mid \mathrm{D})$

$$p(x \mid \mathrm{D}) = \int p(x \mid \mu) P(\mu \mid \mathrm{D}) d\mu$$

$$= \int \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right] \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2} \left(\frac{\mu - \mu_n}{\sigma_n} \right)^2 \right] d\mu$$

$$= \frac{1}{2\pi\sigma\sigma_n} \exp\left[-\frac{1}{2} \frac{(x-\mu_n)^2}{\sigma^2 + \sigma_n^2} \right] f(\sigma, \sigma_n)$$

where

$$f(\sigma, \sigma_n) = \int \exp\left[-\frac{1}{2} \frac{\sigma^2 + \sigma_n^2}{\sigma^2 \sigma_n^2} \left(\mu - \frac{\sigma_n^2 x + \sigma^2 \mu_n}{\sigma^2 + \sigma_n^2} \right)^2 \right] d\mu$$

Bayesian Learning – Univariate Normal Distribution

- Since $p(x \mid \mathrm{D}) \propto \exp\left[-\frac{1}{2}\frac{(x-\mu_n)^2}{\sigma^2+\sigma_n^2}\right]$ we can write

$$p(x \mid \mathrm{D}) : N(\mu_n, \sigma^2 + \sigma_n^2)$$

- To obtain the class conditional probability $p(x \mid \mathrm{D})$, whose parametric form is known to be $p(x \mid \mu) : N(\mu, \sigma)$ we replace μ by μ_n and σ^2 by $\sigma^2 + \sigma_n^2$
- The conditional mean μ_n is treated as if it were the true mean, and the known variance is increased to account for the additional uncertainty in *x* resulting from our lack of exact knowledge of the mean μ .

Example (demo-MAP)

- We have N points which are generated by one dimensional Gaussian, $p(x \mid \mu) = G_x[\mu, 1]$. Since we think that the mean should not be very big we use as a prior $p(\mu) = G_\mu[0, \alpha^2]$, where α is a hyperparameter. The total objective function is:

$$E \propto -\sum_{n=1}^{N} (x_n - \mu)^2 - \frac{\mu^2}{\alpha^2}$$

which is maximized to give,

$$\mu = \frac{1}{N + {}^{1}\!/\!{}_{\alpha^2}} \sum_{n=1}^{N} x_n$$

For $N ? \frac{1}{\alpha^2}$ influence of prior is negligible and result is ML estimate. But for very strong belief in the prior $\frac{1}{\alpha^2} ? \; N$ the estimate tends to zero. Thus,

if few data are available, the prior will bias the estimate towards the prior expected value

Recursive Bayesian Incremental Learning

- We have seen that $p(\mathrm{D}\,|\,\boldsymbol{\theta}) = \prod_{k=1}^{n} p(x_k\,|\,\boldsymbol{\theta})$, Let us define $\mathrm{D}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
Then $\quad p(\mathrm{D}^n\,|\,\boldsymbol{\theta}) = p(x_n\,|\,\boldsymbol{\theta})\,p(\mathrm{D}^{n-1}\,|\,\boldsymbol{\theta}).$
- Substituting into $p(\theta\,|\,\mathrm{D})$, and using Bayes we have:

$$p(\boldsymbol{\theta}\,|\,\mathrm{D}^n) = \frac{p(\mathrm{D}^{\mathrm{n}}\,|\,\boldsymbol{\theta})\,p(\boldsymbol{\theta})}{\int p(\mathrm{D}^n\,|\,\boldsymbol{\theta})\,p(\boldsymbol{\theta})\,d\boldsymbol{\theta}} = \frac{p(\mathbf{x}_n\,|\,\boldsymbol{\theta})\,p(\mathrm{D}^{n-1}\,|\,\boldsymbol{\theta})\,p(\boldsymbol{\theta})}{\int p(\mathbf{x}_n\,|\,\boldsymbol{\theta})\,p(\mathrm{D}^{n-1}\,|\,\boldsymbol{\theta})\,p(\boldsymbol{\theta})\,d\boldsymbol{\theta}}$$

$$= \frac{\mathrm{p}(\mathrm{x_n}|\theta)\mathrm{p}(\theta|\mathrm{D}^{\mathrm{n-1}})\dfrac{\mathrm{p}(\mathrm{D}^{\mathrm{n-1}})}{\mathrm{p}(\theta)}\mathrm{p}(\theta)}{\int \mathrm{p}(\mathrm{x_n}|\theta)\mathrm{p}(\theta|\mathrm{D}^{\mathrm{n-1}})\dfrac{\mathrm{p}(\mathrm{D}^{\mathrm{n-1}})}{\mathrm{p}(\theta)}\mathrm{p}(\theta)\mathrm{d}\theta}$$

Finally

$$\mathrm{p}(\theta|\mathrm{D}^{\mathrm{n}}) = \frac{\mathrm{p}(\mathrm{x_n}|\theta)\mathrm{p}(\theta|\mathrm{D}^{\mathrm{n-1}})}{\int \mathrm{p}(\mathrm{x_n}|\theta)\mathrm{p}(\theta|\mathrm{D}^{\mathrm{n-1}})\mathrm{d}\theta}$$

Recursive Bayesian Incremental Learning

- While $p(\theta|D^0)=p(\theta)$, repeated use of this eq. produces a sequence
- $$p(\boldsymbol{\theta}), p(\boldsymbol{\theta} \mid \mathbf{x}_1), p(\boldsymbol{\theta} \mid \mathbf{x}_1, \mathbf{x}_1), \ldots$$
- This is called the *recursive Bayes* approach to the parameter estimation. (Also *incremental* or *on-line* learning).
- When this sequence of densities converges to a Dirac delta function centered about the true parameter value, we have *Bayesian learning*.

Maximal Likelihood vs. Bayesian

- ML and Bayesian estimations are asymptotically equivalent and “consistent”. They yield the same class-conditional densities when the size of the training data grows to infinity.
- ML is typically computationally easier: in ML we need to do (multidimensional) differentiation and in Bayesian (multidimensional) integration.
- ML is often easier to interpret: it returns the single best model (parameter) whereas Bayesian gives a weighted average of models.
- But for a finite training data (and given a reliable prior) Bayesian is more accurate (uses more of the information).
- Bayesian with “flat” prior is essentially ML; with asymmetric and broad priors the methods lead to different solutions.

Problems of Dimensionality:Accuracy, Dimension, and Training Sample Size

- Consider two-class multivariate normal distributions $p(\mathbf{x}|\omega_i) : N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$ with the same covariance. If priors are equal then Bayesian error rate is given by

$$P(e) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{-u^2/2} du,$$

where r^2 is the squared Mahalanobis distance:

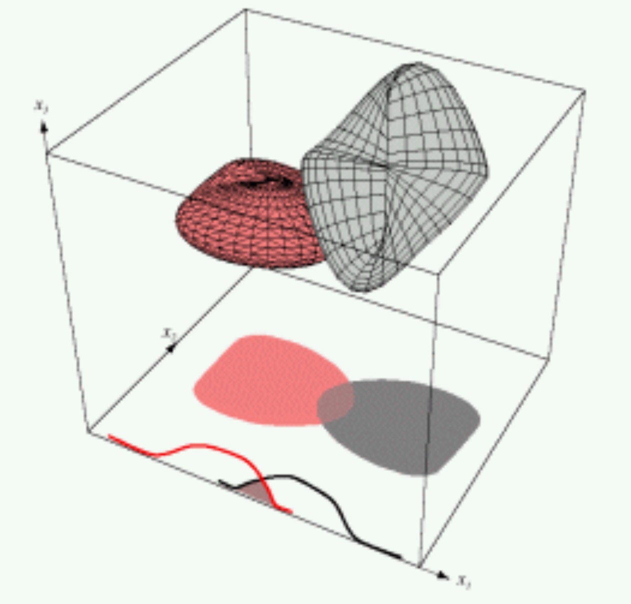
$$r^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \Sigma^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2).$$

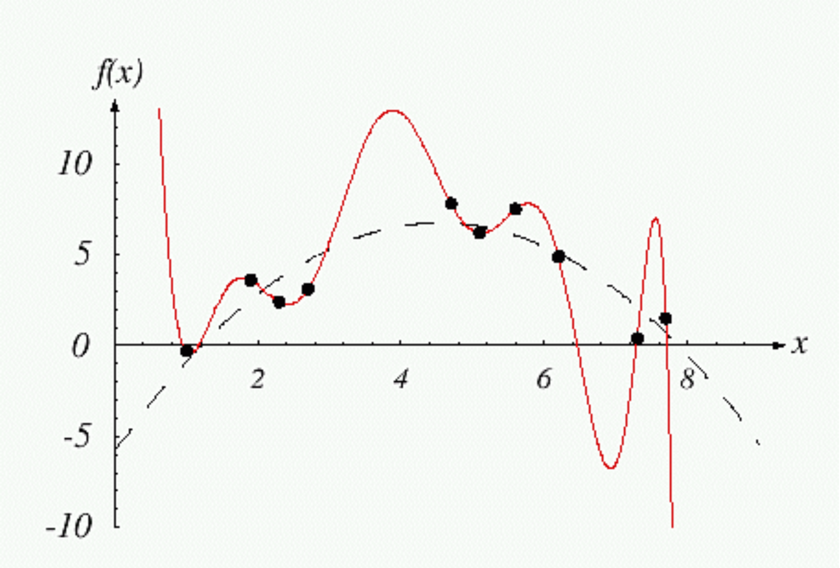
- Thus the probability of error decreases as *r* increases. In the conditionally independent case $\boldsymbol{\Sigma} = diag(\sigma_1^2, ..., \sigma_d^2)$ and

$$r^2 = \sum_{i=1}^{d} \left(\frac{\mu_{i1} - \mu_{i2}}{\sigma_i} \right)^2$$

Problems of Dimensionality

- While classification accuracy can become better with growing of dimensionality (and an amount of training data),
  - **beyond a certain point, the inclusion of additional features leads to worse rather then better performance**
  - **computational complexity grows**
  - **the problem of overfitting arises**





Occam's Razor

- "*Pluralitas non est ponenda sine neccesitate*" or "plurality should not be posited without necessity." The words are those of the medieval English philosopher and Franciscan monk William of Occam (ca. 1285-1349).

Decisions based on overly complex models often lead to lower accuracy of the classifier.

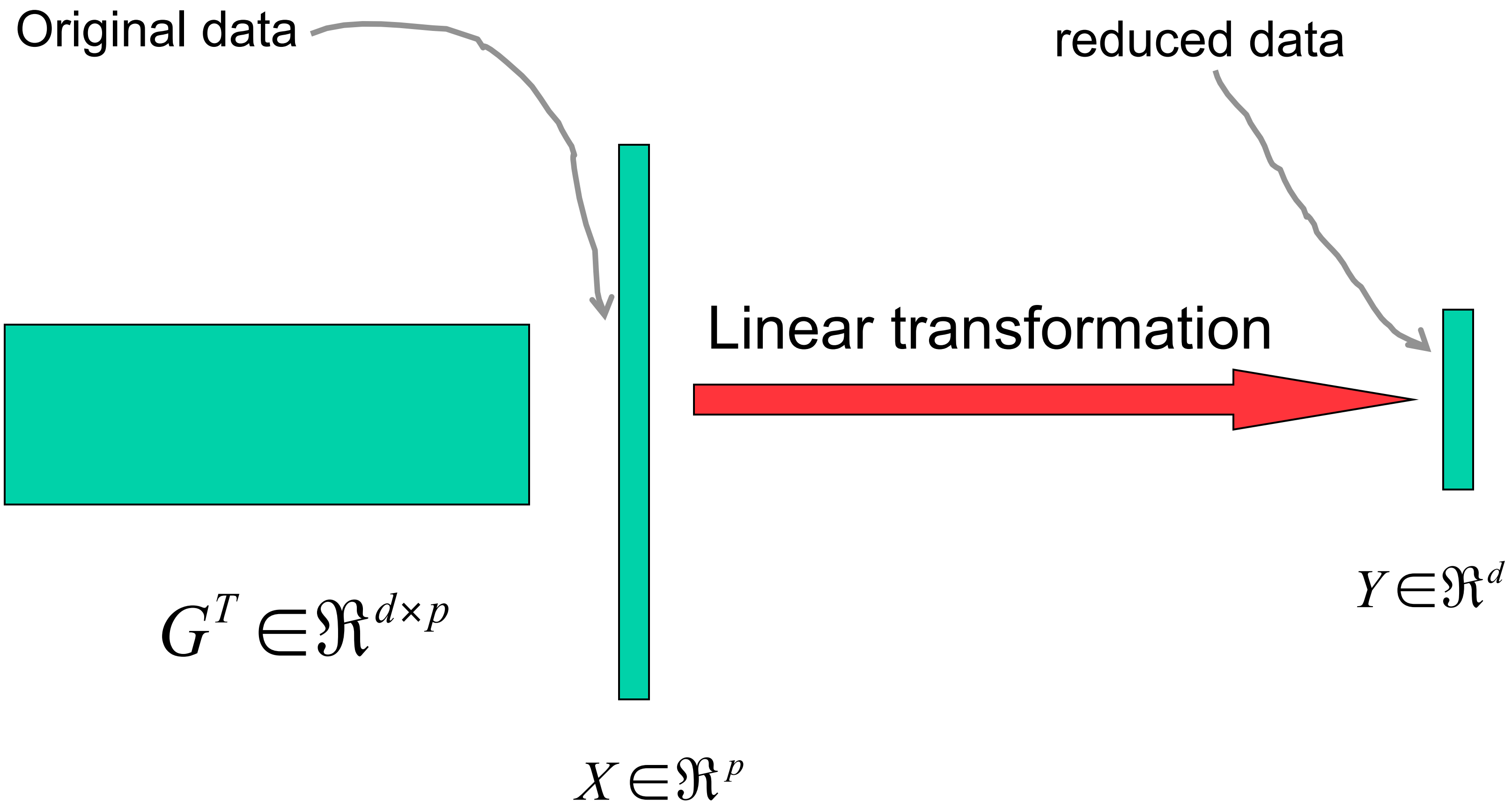
What is feature reduction?

- Feature reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space.
  - Criterion for feature reduction can be different based on different problem settings.
    - Unsupervised setting: minimize the information loss
    - Supervised setting: maximize the class discrimination
- Given a set of data points of p variables $\{x_1, x_2, \cdots, x_n\}$

  Compute the linear transformation (projection)

$$G \in \Re^{p \times d} : x \in \Re^{p} \rightarrow y = G^{T} x \in \Re^{d} \ (d << p)$$

What is feature reduction?



$G^T \in \Re^{d \times p}$

$X \in \Re^{p}$

$Y \in \Re^{d}$

$$G \in \Re^{p \times d} : X \rightarrow Y = G^T X \in \Re^{d}$$