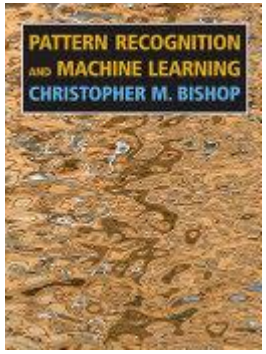
Pattern Analysis and Recognition

Lecture 6: Parameter Estimation,
Bayesian Classification

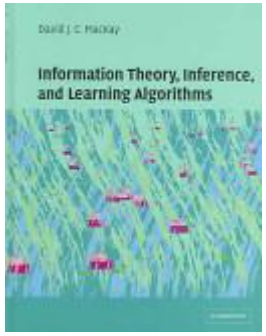
Resources

Some of the material in this slides was borrowed from:

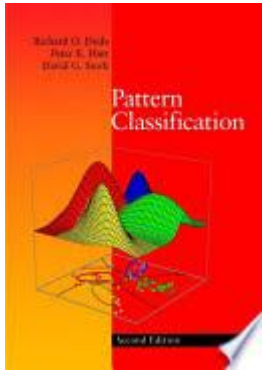
C. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006
Some related material available:
http://research.microsoft.com/en-us/um/people/cmbishop/prml/index.htm



D. MacKay, "Information Theory, Inference and Learning Algorithms", Cambridge University Press, 2003.
Book available online:
http://www.inference.phy.cam.ac.uk/mackay/



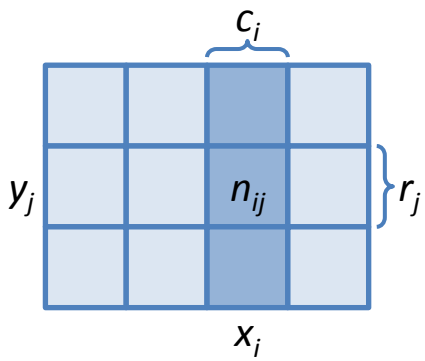
R.O. Duda, P.E. Hart, D.G. Stork, "Pattern Classification", Wiley & Sons, 2000
Have a look inside at selected chapters:
http://books.google.es/books/about/Pattern_Classification.html?id=Br33IRC3PkQC&redir_esc=y



Last time on Pattern Analysis and Recognition

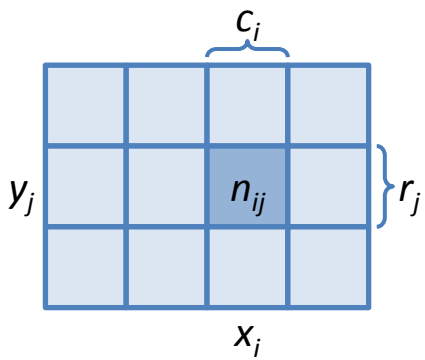
RECAP

Probability Theory



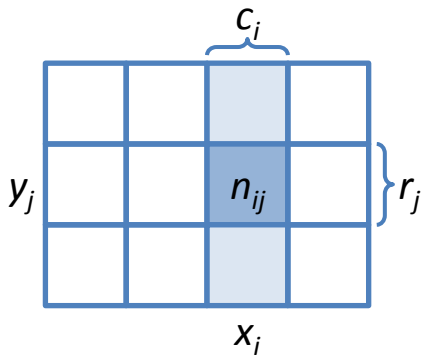
Marginal Probability

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



Joint Probability

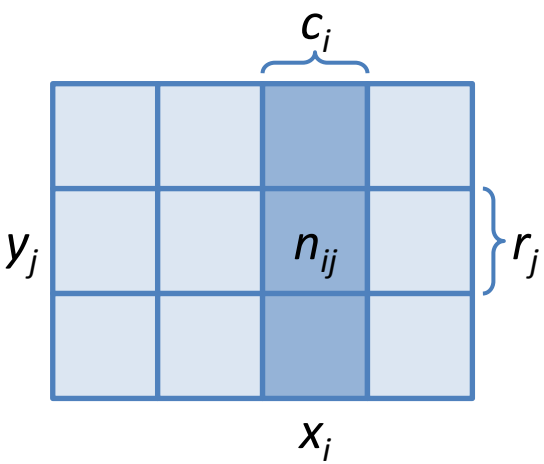
$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$



Conditional Probability

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

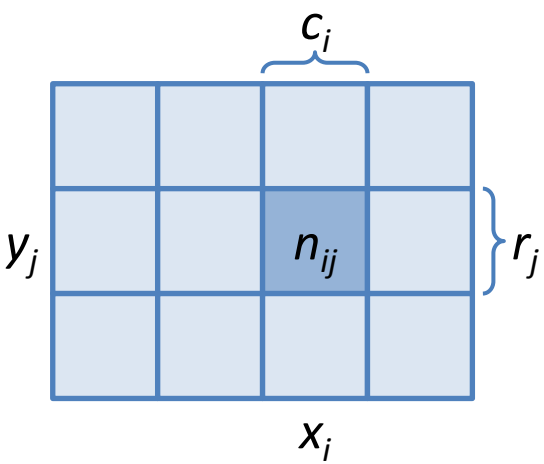
Probability Theory



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

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

$$= \frac{n_{ij}}{c_i}\frac{c_i}{N}$$

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

Bayes' Theorem

likelihood

prior

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

posterior

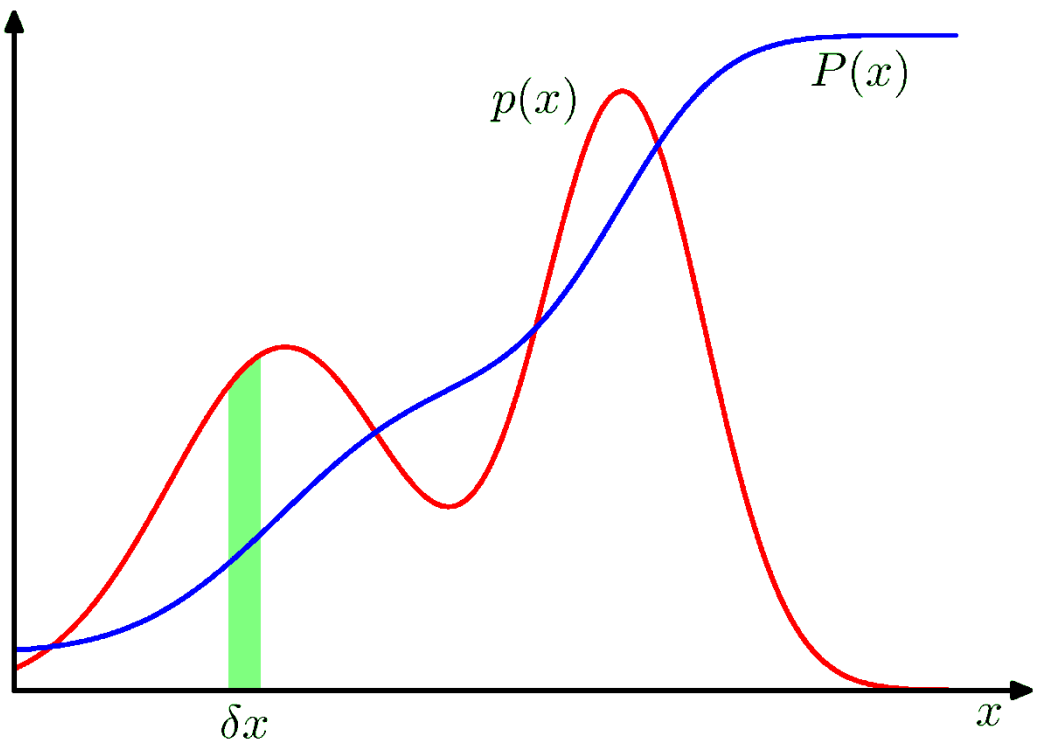
evidence

$$p(X) = \sum_{Y} p(X|Y)\,p(Y)$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

Probability Densities

The concept of probability for discrete variables can be extended to that of a probability density over a continuous variable *x*



$$p(x \in (a,b)) = \int_a^b p(x)dx$$

$$p(x) \geq 0$$

$$\int_{-\infty}^{\infty} p(x)dx = 1$$

Cumulative distribution function: $P(z) = \int_{-\infty}^{z} p(x)dx$

EXPECTATIONS, COVARIANCES AND THE GAUSSIAN DISTRIBUTION

Expectations

The **expectation** of some function *f(x)* is the average value of *f(x)* under a probability distribution *p(x)*

$$\mathrm{E}[f] = \sum_x p(x) f(x) \qquad\qquad \mathrm{E}[f] = \int p(x) f(x)$$

If a conditional probability is involved we talk about the conditional expectation

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

It turns out that if we are given a finite number of N points drawn from the probability distribution, expectation can be approximated as:

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

Variances and Covariances

The **variance** of some function *f(x)* provides a measure of how much variability there is in *f(x)* around its expected value E[*f(x)*]

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

For two random variables *x* and *y* **covariance** expresses the extent to which *x* and *y* vary together

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

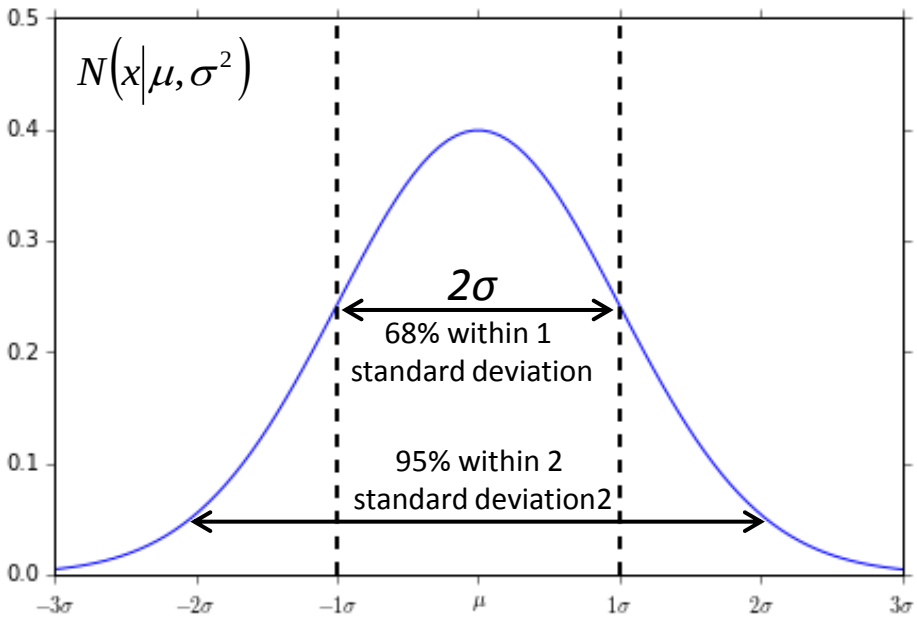
If x and y are independent the covariance vanishes

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

In the case of two vectors of random variables, the covariance is a matrix

The (univariate) Gaussian Distribution

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



$$N\left(x\middle|\mu,\sigma^2\right)>0$$

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

μ	mean
σ^2	variance
σ	standard deviation
$\beta=1/\sigma^2$	reciprocal of the variance – also called precision

Gaussian Mean and Variance

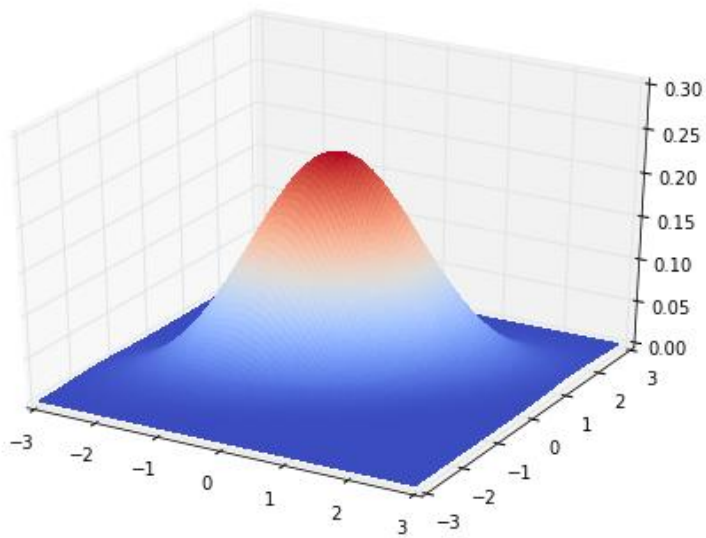
$$\mathrm{E}[x] = \int_{-\infty}^{\infty} N\left(x \middle| \mu, \sigma^2\right) x \, dx = \mu$$

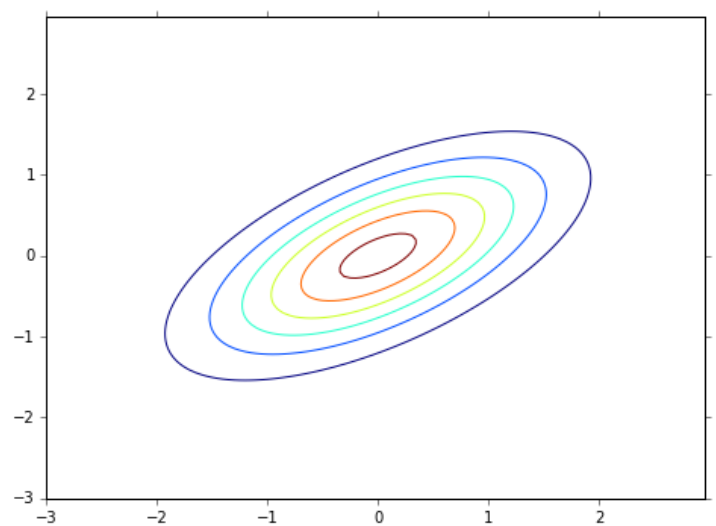
$$\mathrm{E}\left[x^2\right] = \int_{-\infty}^{\infty} N\left(x \middle| \mu, \sigma^2\right) x^2 \, dx = \mu^2 + \sigma^2$$

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

The Multivariate Gaussian

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





d	number of dimensions
μ	mean
Σ	dxd covariance matrix
|Σ|	determinant

$$\mathbf{x}=\begin{bmatrix}x_1\\x_2\\\vdots\\x_d\end{bmatrix}\qquad \boldsymbol{\mu}=\begin{bmatrix}\mu_1\\\mu_2\\\vdots\\\mu_d\end{bmatrix}\qquad \mu_i=\mathrm{E}[x_i]$$

The Multivariate Gaussian

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

this is a number, e.g. for d=2

$$(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}) = \left[(x_1-\mu_1) \quad (x_2-\mu_2)\right]\begin{bmatrix}\sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2\end{bmatrix}\begin{bmatrix}x_1-\mu_1 \\ x_2-\mu_2\end{bmatrix}$$

The Covariance Matrix

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

σ_{ii} variance of x_i: σ_1^2

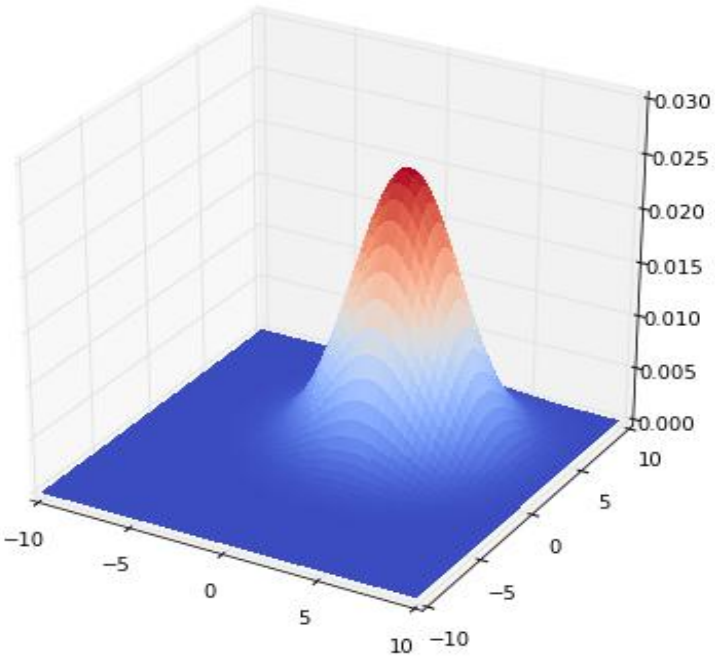
σ_{ij} covariance of x_i and x_j: $\sigma_{ij} = \mathrm{E}\left[(x_i - \mu_i)(x_j - \mu_j)\right]$

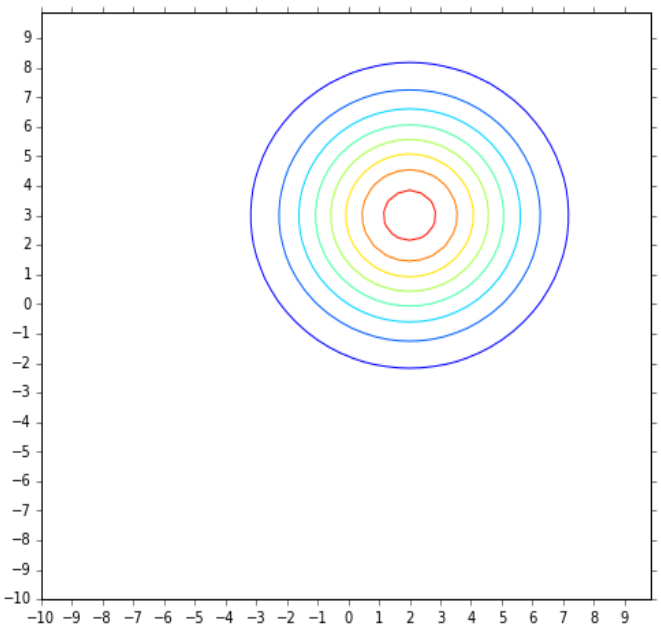
If features x_i and x_j are independent, $\sigma_{ij} = 0$

Examples of bivariate distributions

Bivariate: d=2
Independent features
Equal variance

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





$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 2.5 & 0.0 \\ 0.0 & 2.5 \end{bmatrix}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

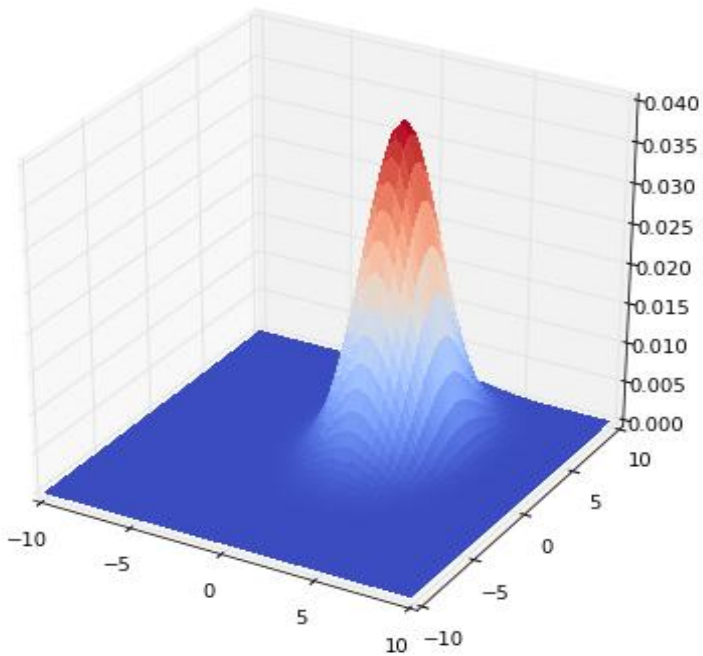
Examples of bivariate distributions

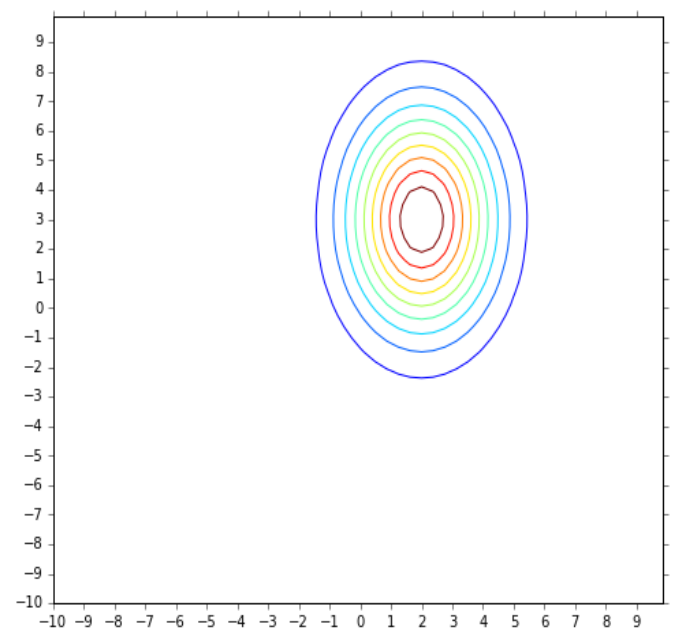
Bivariate: d=2

Independent features

Different variance $\sigma_1 < \sigma_2$

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





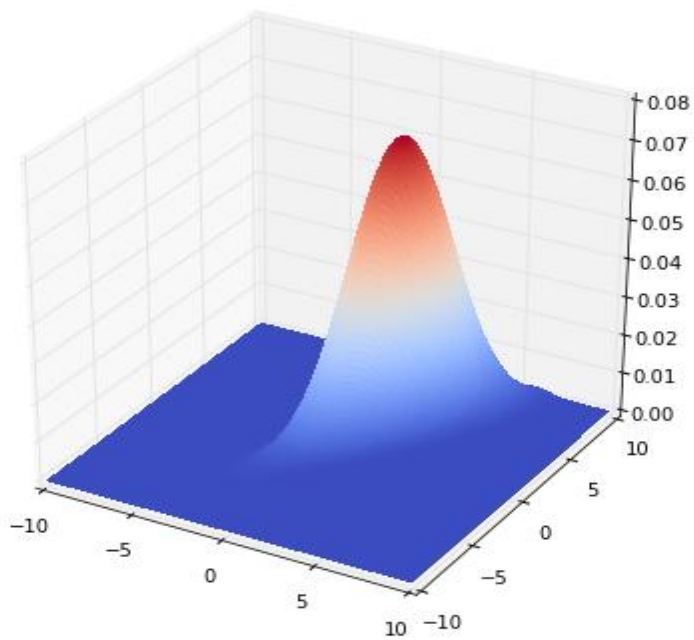
$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 1.6 & 0.0 \\ 0.0 & 2.5 \end{bmatrix}$$

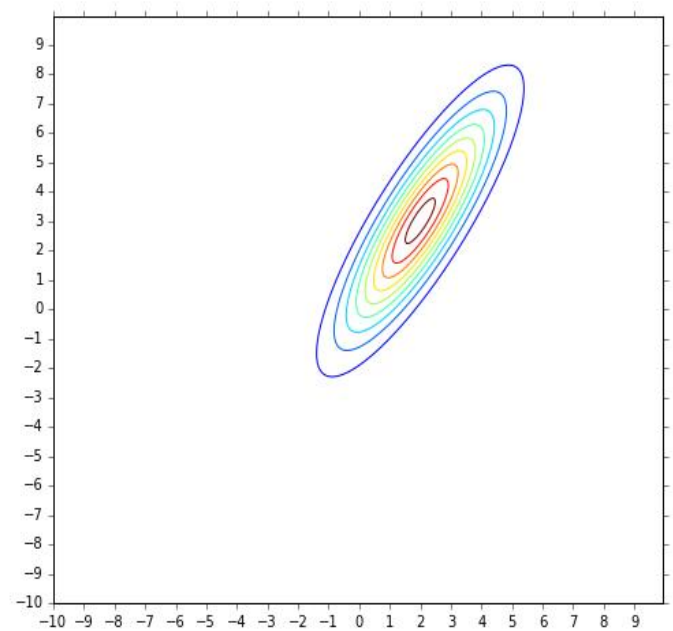
$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Examples of bivariate distributions

Bivariate: d=2
Correlated features
Different variance $\sigma_1 < \sigma_2$

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





$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 1.6 & 3.4 \\ 3.4 & 2.5 \end{bmatrix}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$\rho = \frac{\sigma_{12}}{\sigma_1 \sigma_2} = 0.85$$

PARAMETER ESTIMATION

Gaussian parameter estimation

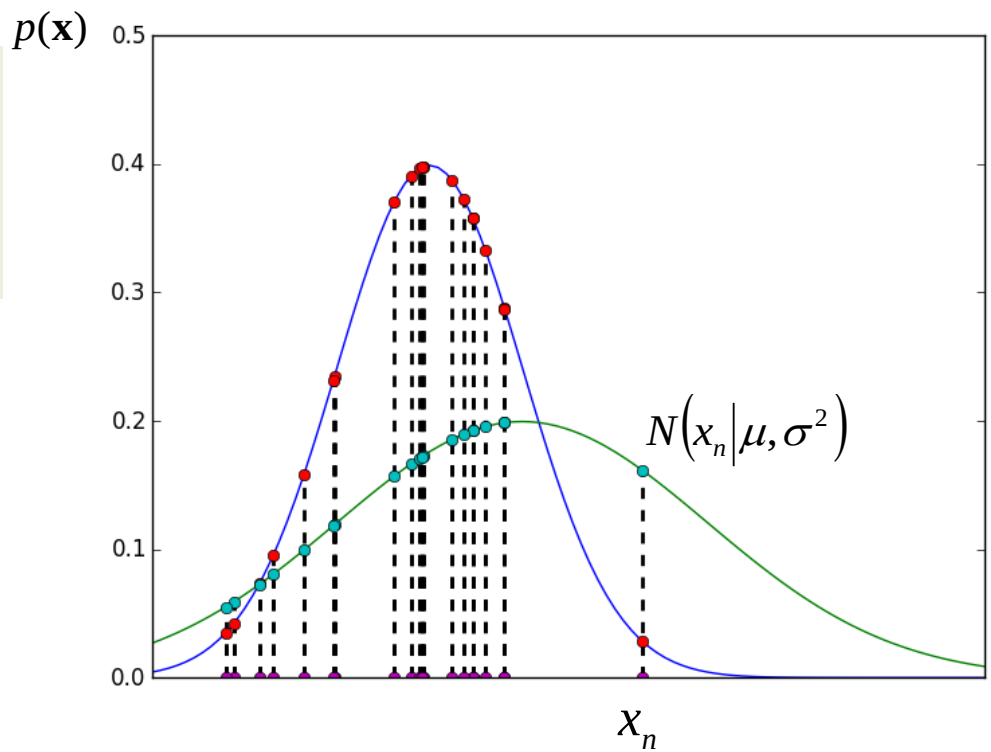
Imagine a set of samples that are drawn independently from the same distribution. The samples are **independent and identically distributed**

Suppose the underlying distribution is a Gaussian. We want to estimate the parameters of the Gaussian *(μ, σ^2)* from the samples we have

Intuition: View parameters as fixed unknown quantities. Maximise the probability of obtaining the samples

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

Likelihood function



Maximum (Log) Likelihood

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

Easier to maximise the log of this function (deal with sums instead of products)

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

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

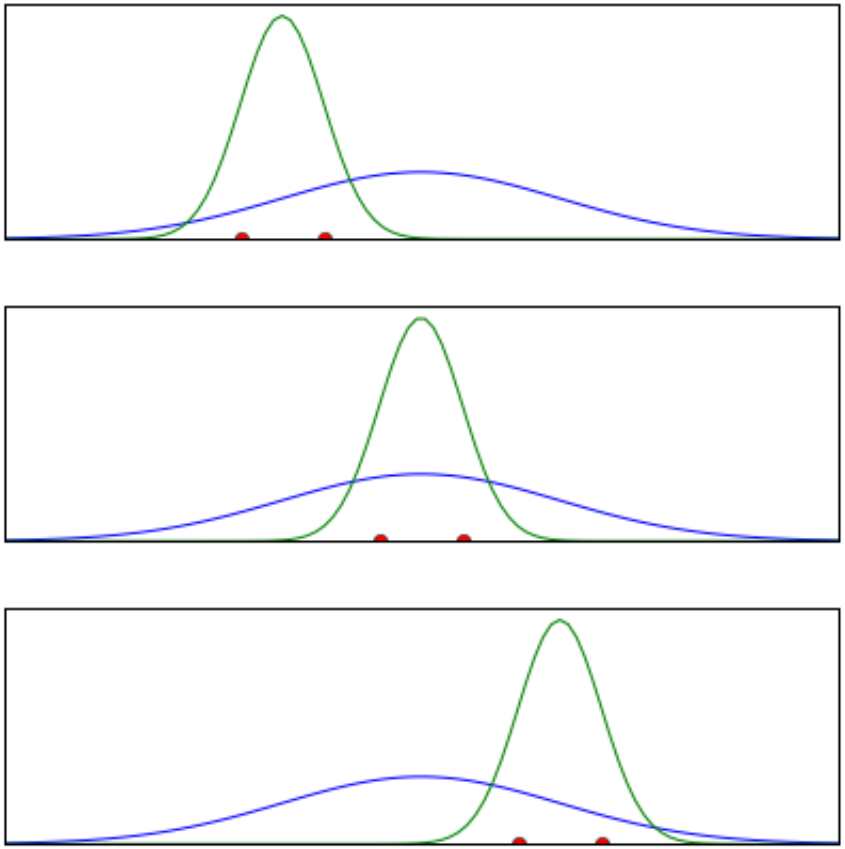
Properties of Maximum Likelihood estimations of μ_{ML} and σ^2_{ML}

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

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

$$\tilde{\sigma}^2 = \frac{N}{N-1}\sigma^2_{\mathrm{ML}}$$

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



Maximum likelihood systematically underestimates the variance of the distribution (bias phenomenon, related to overfitting).

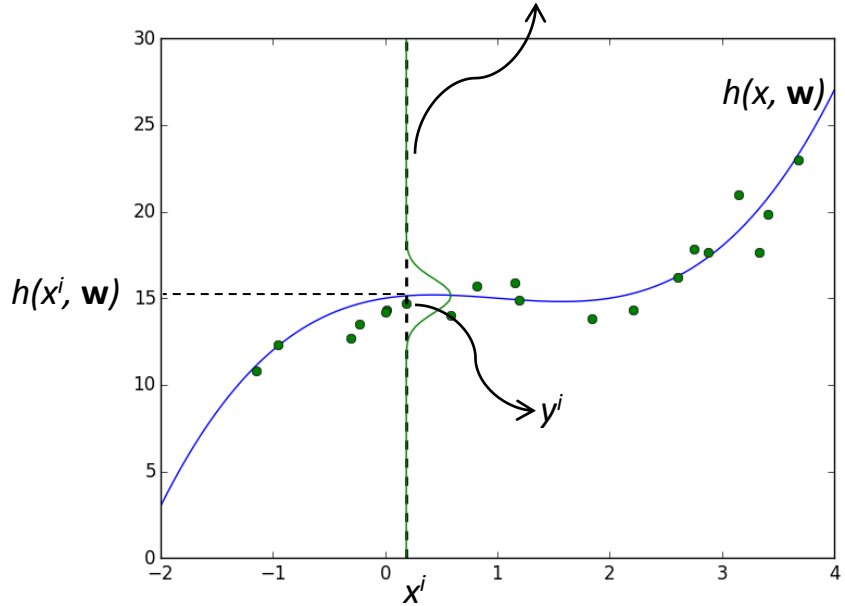
CURVE FITTING RE-VISITED

Curve Fitting – The Bayesian Way

Goal: To be able to make predictions for the target variable *y* given some new value of the input variable *x*, on the basis of a training dataset comprising N input values $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$ and their corresponding target values $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T$

Intuition: Express our uncertainty over the value of the target variable using a (Gaussian) probability distribution.

$$p(y \mid x, \mathbf{w}, \beta) = N\left(y \mid h(x, \mathbf{w}), \beta^{-1}\right)$$



The mean of this Gaussian distribution would be the target variable itself *h(x, w)* and it would have some (unknown) precision, say β

Remember $\beta = 1/\sigma^2$

Hint: remember the generative view of data (Lecture 3) , underlying "true" function and observed samples corrupted with Gaussian noise?
$y = h(x, \mathbf{w}) + N(0, \sigma)$

Maximum (Log) Likelihood

$$p(\mathbf{y} \mid \mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} N\left(y_n \mid h(x_n, \mathbf{w}), \beta^{-1}\right)$$

Easier to maximise the log of this function (deal with sums instead of products)

To determine the coefficients **w** – maximise in respect to **w**

$$\ln p(\mathbf{y} \mid \mathbf{x}, \mathbf{w}, \beta) = \underbrace{-\frac{\beta}{2} \sum_{n-1}^{N} \left\{h(x_n, \mathbf{w}) - y_n\right\}^2}_{\beta J(\mathbf{w})} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

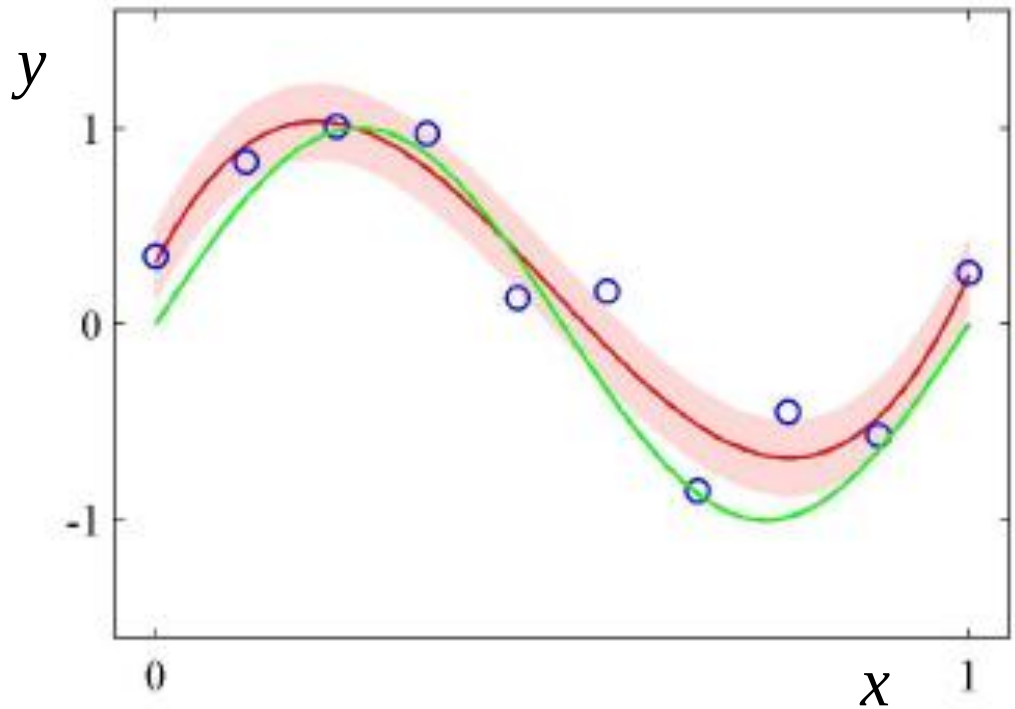
Omit the last two terms as they do not depend on **w**. Ignore β. Turns out it is equivalent to minimising the sum-of-squares error function!

To determine the precision parameter *β* – maximise with respect to *β*, given $\mathbf{w}_{\mathrm{ML}}$

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \left\{h\left(x_n, \mathbf{w}_{ML}\right) - y_n\right\}^2$$

Predictive Distribution

$$p(y \mid x, \mathbf{w}_{ML}, \beta_{ML}) = N\left(y \mid h(x, \mathbf{w}_{ML}), \beta_{ML}^{-1}\right)$$



Our predictions are now expressed in terms of the *predictive distribution* of the target value, which gives the probability distribution over *y* given an input value *x*, than simply a point estimate.

MAP: A step towards Bayes

Intuition: What if we had some idea about the right parameters in advance… the Bayesian approach gives us the mechanism to take it into account.

Total number of elements for an Mth order polynomial

$$p(\mathbf{w} \mid \alpha) = N\left(\mathbf{w} \mid \mathbf{0}, \alpha^{-1}\mathbf{I}\right) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right\}$$

α is an "hyperparameter" – controls the distribution of model parameters (**w**)

$$p(\mathbf{w} \mid \mathbf{x}, \mathbf{y}, \alpha, \beta) \propto p(\mathbf{y} \mid \mathbf{x}, \mathbf{w}, \beta)\, p(\mathbf{w} \mid a)$$

Using Bayes' theorem, we get the posterior. Determine **w** by finding the most probable value of **w** given the data **x**: maximise posterior probability

Maximising posterior is equivalent (take the negative log) to minimising regularized sum of squares error with a regularisation parameter of $\lambda = \alpha/\beta$

$$\beta \tilde{J}(\mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}\left\{h(x_n, \mathbf{w}) - y_n\right\}^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}$$

Bayesian Curve Fitting

We have found the most probable values for **w**, but this is still far from a true Bayesian treatment.

Intuition: any values are possible for **w**, with some associated probability. If we consistently apply the sum and product rules, we end up integrating over all possible values, instead of using a point estimate of **w** for our predictions.

$$p(y \mid x, \mathbf{x}, \mathbf{y}) = \int p(y \mid x, \mathbf{w}) p(\mathbf{w} \mid \mathbf{x}, \mathbf{y}) d\mathbf{w}$$

This can be calculated analytically: $p(y \mid x, \mathbf{x}, \mathbf{y}) = N\left(y \mid m(x), s^2(x)\right)$

where

$$m(x) = \beta \phi(x)^T \mathbf{S} \sum_{n=1}^{N} \phi(x_n) t_n$$

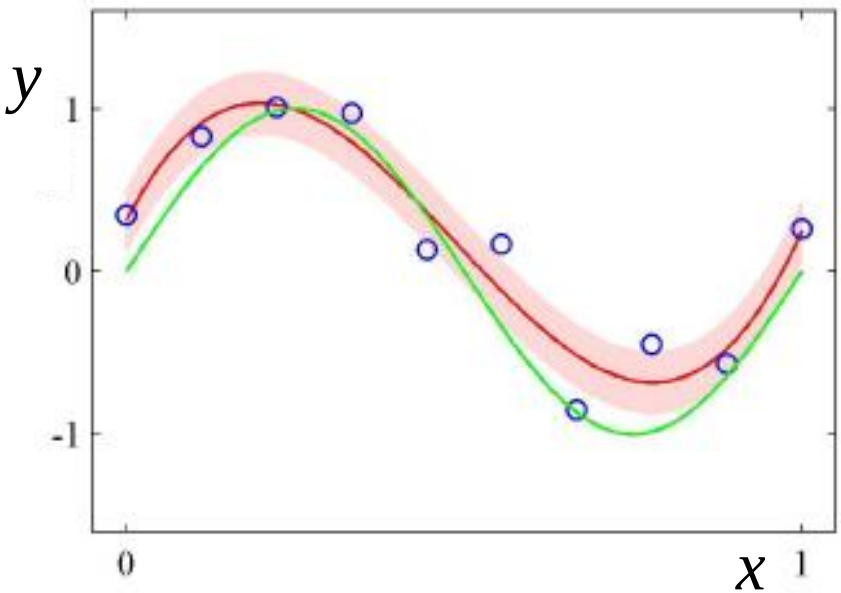
$$s^2(x) = \beta^{-1} + \underbrace{\phi(x)^T \mathbf{S} \phi(x)}$$

Uncertainty due to noise on target variables

Uncertainty in the parameters **w**

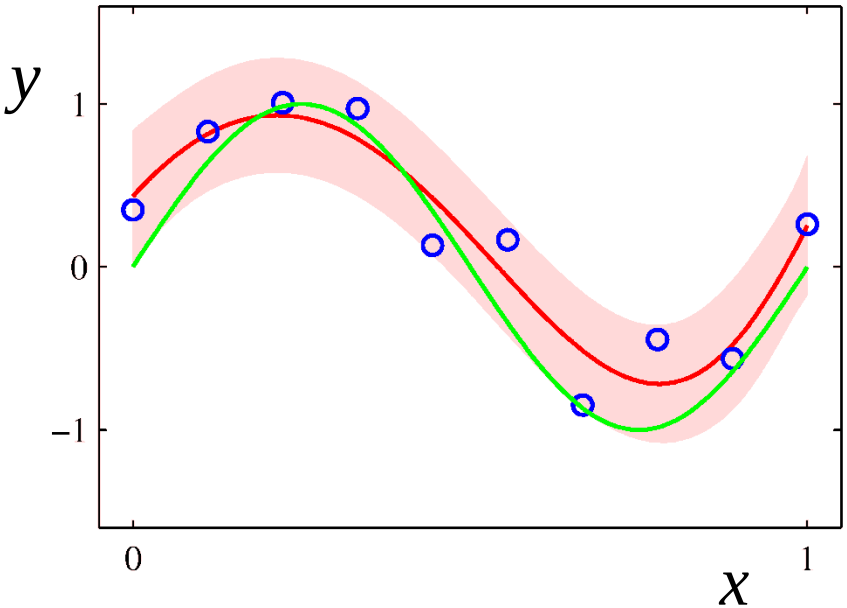
Bayesian Predictive Distribution

$$p(y \mid x, \mathbf{w}_{ML}, \beta_{ML}) = N\left(y \mid h(x, \mathbf{w}_{ML}), \beta_{ML}^{-1}\right)$$



Predictive distribution

$$p(y \mid x, \mathbf{x}, \mathbf{y}) = N\left(y \mid m(x), s^2(x)\right)$$



Bayesian Predictive distribution

PATTERN CLASSIFICATION

Two class scenario

ω_1, ω_2	two classes (e.g. good/bad, 1/0)
$P(\omega_1), P(\omega_2)$	a-priori probabilities of priors
x	continuous random variable (1 feature)
$p(x\|\omega_1), p(x\|\omega_2)$	class-conditional probability density functions

$$P(\omega_j \mid x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)} \qquad \text{posterior} \propto \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

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

Bayes' Decision Rule

Bayes' decision rule: choose the class with the highest a-posteriori probability

$$\text{Decide} \begin{cases} \omega_1 & \text{if } P(\omega_1 \mid x) \geq P(\omega_2 \mid x) \\ \omega_2 & \text{if } P(\omega_1 \mid x) < P(\omega_2 \mid x) \end{cases}$$

$$P(\omega_j \mid x) = \frac{p(x \mid \omega_j) P(\omega_j)}{p(x)}$$

Note that evidence *p(x)* is irrelevant to the decision

$$\text{Decide} \begin{cases} \omega_1 & \text{if } p(x \mid \omega_1) P(\omega_1) \geq p(x \mid \omega_2) P(\omega_2) \\ \omega_2 & \text{if } p(x \mid \omega_1) P(\omega_1) < p(x \mid \omega_2) P(\omega_2) \end{cases}$$

Applying Bayes' Decision Rule

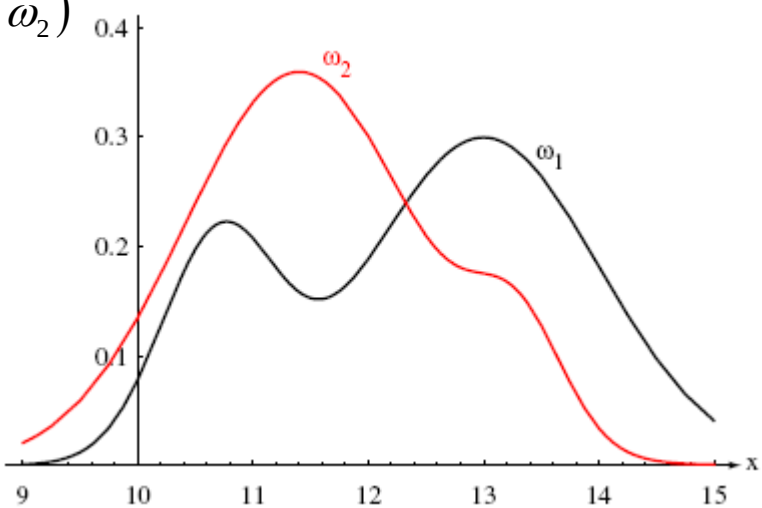
$$P(\omega_j \mid x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}$$

$$\text{if } P(\omega_1) = 2/3$$

$$P(\omega_2) = 1/3$$

$p(x \mid \omega_1)$

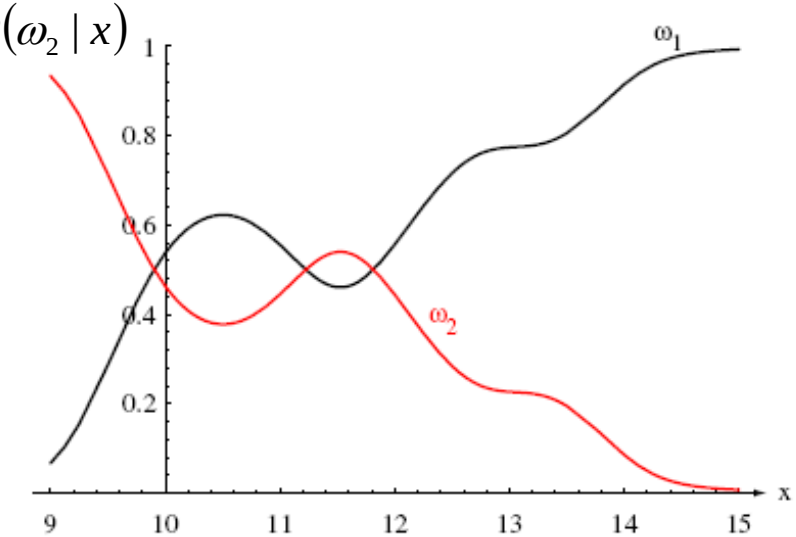
$p(x \mid \omega_2)$



Hypothetical class-conditional probability density functions. The area under each curve is *1.0*.

$P(\omega_1 \mid x)$

$P(\omega_2 \mid x)$



Posterior probabilities. At every *x* they sum to *1.0* thanks to *p(x)* that acts as a normalisation factor.

Error Definition

The Bayes' decision rule makes intuitive sense, but is it really a good decision rule?

A good rule should minimise the (average) probability of making an error

$$P(error) = \int_{-\infty}^{\infty} P(error, x)\, dx = \int_{-\infty}^{\infty} P(error \mid x)\, p(x)\, dx$$

Error Definition

Independently of the decision rule, the probability of error is equal to the probability of having selected the wrong class

$$P(error \mid x) = \begin{cases} P(\omega_1 \mid x) & \text{if we decide } \omega_2 \\ P(\omega_2 \mid x) & \text{if we decide } \omega_1 \end{cases}$$

For the Bayes' decision rule ***"decide ω_1 if $P(\omega_1|x) > P(\omega_2|x)$, otherwise decide ω_2"*** this is:

$$P(error \mid x) = \min\left[P(\omega_1 \mid x), P(\omega_2 \mid x)\right]$$

This ensures that for each x the probability of error is as small as possible – the best possible rule we can have.

Multiple Classes and Features

$\omega_1, \omega_2, \ldots, \omega_c$ finite set of *c* classes

$x = (x_1, x_2, \ldots, x_d)^T$ *d*-dimensional feature vector (d features)

Assign feature vector x to class ω_i if $\arg\max_j \; p(\mathbf{x} \mid \omega_j) P(\omega_j)$

Or equivalently if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $j \neq i$ where

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid \omega_i) P(\omega_i)$$

*g(**x**)* is a possible "*discriminant function*"

Discriminant Functions for $N(\mu,\mathbf{\Sigma})$

For normal densities, instead of:

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid \omega_i) P(\omega_i)$$

It is better to take the logarithm, as it produces simpler expressions:

$$g_i(\mathbf{x}) = \ln p(\mathbf{x} \mid \omega_i) + \ln P(\omega_i)$$

$$g_i(\mathbf{x}) = -\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}_i\right)^T \Sigma_i^{-1}\left(\mathbf{x} - \boldsymbol{\mu}_i\right) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln\left|\Sigma_i\right| + \ln P(\omega_i)$$

Loss (Cost) Function

Bayes' rule minimizes the average probability error

Considerations:
- It is not possible to **guarantee** zero errors
- Erroneous decisions have a **associated cost** in real life
- The cost is generally not uniform, but **varies** with the decision

Instead of minimising the average probability of error, we should try to minimise the overall cost that our decisions have

	Cancer	Normal
Cancer	0	1000
Normal	1	0

An example of a loss (cost) matrix

Loss (Cost) Function

$\omega_1, \omega_2, \ldots, \omega_c$	finite set of *c* classes
$\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$	*d*-dimensional feature vector (d features)
$\alpha_1, \alpha_2, \ldots, \alpha_\alpha$	finite set of α **possible actions** we can take
$\lambda(\alpha_i \mid \omega_j)$	**loss function**: the cost of taking action α_i when the true class is ω_j
$R(\alpha_i \mid \mathbf{x})$	the **conditional risk** or else expected loss of taking action α_i given by:

$$R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid \mathbf{x})$$

Note that this is the *expected* loss, not the real one, as we do not know the real class

Loss (Cost) Function

We can then define the overall risk as the *expected* loss associated with some decision rule *α(***x***)*, i.e. taking into account all **x**

$$R = \int R\big(\alpha(\mathbf{x}) \mid \mathbf{x}\big) p(\mathbf{x}) d\mathbf{x}$$

Our decision rule then, would be to minimise the overall risk R. This overall risk is minimised if for every x we choose the action with the minimum risk

$$\alpha(\mathbf{x}) = \arg\min_i R(\alpha_i \mid \mathbf{x})$$

$$= \arg\min_i \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid \mathbf{x})$$

Loss (Cost) Function

The minimum error rule that we used before is a particular case:

If

$$\lambda(\alpha_i \mid \omega_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \quad i, j = 1, \ldots, c$$

then

$$R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid \mathbf{x}) = \sum_{j \neq i} P(\omega_j \mid \mathbf{x})$$

Note that in this case $\alpha(\mathbf{x}) = \arg\min_i R(\alpha_i \mid \mathbf{x})$

is equivalent to "decide ω_i if $P(\omega_i|x) > P(\omega_j|x)$, for all $i \neq j$

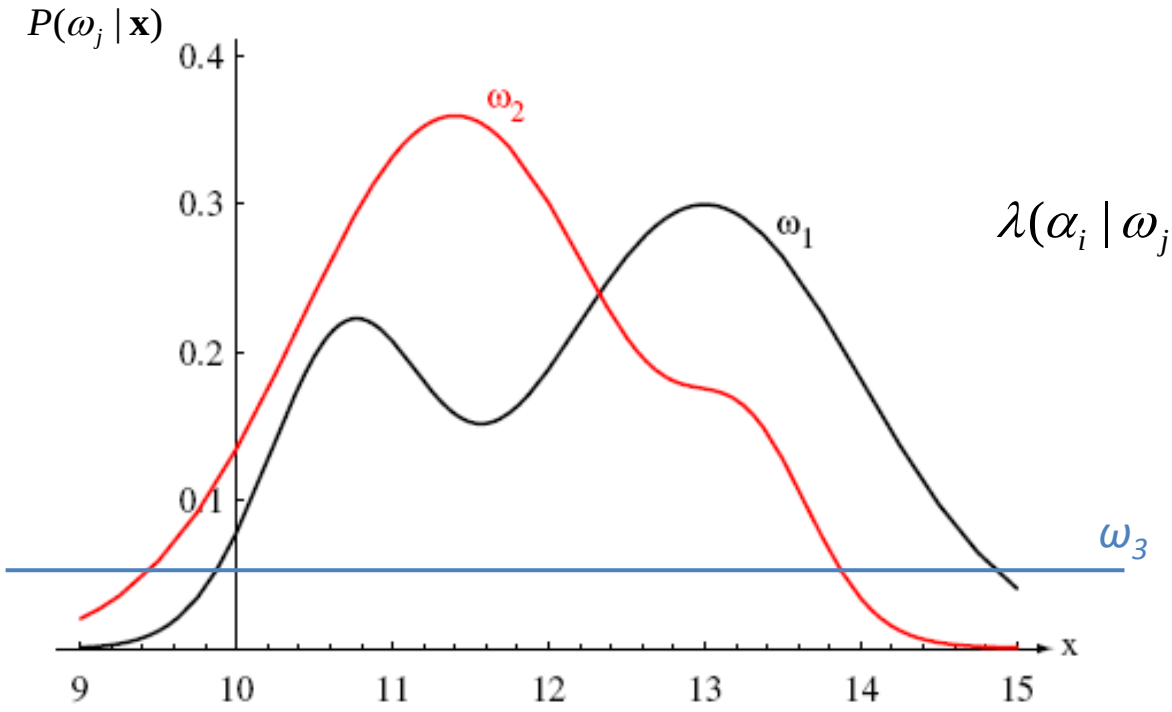
Rejection

In many cases, it might be preferable to refuse to make a decision. In different words, the cost of not making a decision might be lower than any of the other possible actions (e.g. a human doctor will look into the case if our system cannot decide)

$\omega_1, \omega_2, \ldots, \omega_c$	finite set of c classes
$\alpha_1, \alpha_2, \ldots, \alpha_c, \alpha_{c+1}$	$c+1$ **possible actions** that correspond to selecting any of the known classes (α_i is to decide ω_i, $i = 1 \ldots c$), or **not deciding** (α_{c+1}), equivalent to selecting none of the known classes
$\lambda(\alpha_{c+1} \mid \omega_j)$	costs of **not** making a decision

Rejection

Imagine that if the MAP is less than *5%* we do not want to make a decision. A way to think about this would be like having a new class *c+1* such that $P(\omega_{c+1}|\mathbf{x})=0{,}05$



$$\lambda(\alpha_i|\omega_j)=\begin{cases}0 & i=j\\ 1 & i\neq j\end{cases}$$

Classification Summary

Rule to minimise average error: $\arg\max_j P(\omega_j \mid \mathbf{x})$

Bayes' theorem: $P(\omega_j \mid \mathbf{x}) \propto p(\mathbf{x} \mid \omega_j) P(\omega_j)$

Conditional risk (expected cost) of taking an action α_i : $R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid \mathbf{x})$

Rule to minimize average risk: $\arg\min_i R(\alpha_i \mid \mathbf{x})$

Rejection action, equivalent to not making a decision, with costs: $\lambda(\alpha_{c+1} \mid \omega_j)$

What’s Next

		Mondays	Tuesdays				
		16:00 - 18:00	15:00 - 17:00				
Practical Sessions		**M**	**T**	**W**	**T**	**F**	**Lectures**
	Feb	8	9	10	11	12	Introduction and Linear Regression
P0. Introduction to Python, Linear Regression		15	16	17	18	19	Logistic Regression, Normalization
P1. Text non-text classification (Logistic Regression)		22	23	24	25	26	Regularization, Bias-variance decomposition
	Mar	29	1	2	3	4	Normalization and subspace methods (dimensionality reduction)
		7	8	9	10	11	Probabilities, Bayesian inference
Discussion of intermediate deliverables / project presentations		14	15	16	17	18	Parameter Estimation, Bayesian Classification
		21	22	23	24	25	Easter Week
	Apr	28	29	30	31	1	Clustering, Gausian Mixture Models, Expectation Maximisation
P2. Feature learning (k-means clustering, NN, bag of words)		4	5	6	7	8	Nearest Neighbour Classification
		11	12	13	14	15	
		18	19	20	21	22	Kernel methods
Discussion of intermediate deliverables / project presentations		25	26	27	28	29	Support Vector Machines, Support Vector Regression
P3. Text recognition (multi-class classification using SVMs)	**May**	2	3	4	5	6	Neural Networks
		9	10	11	12	13	Advanced Topics: Metric Learning, Preference Learning
		16	17	18	19	20	Advanced Topics: Deep Nets
Final Project Presentations		23	24	25	26	27	Advanced Topics: Structural Pattern Recognition
	Jun	30	31	1	2	3	Revision

LEGEND	
	Project Follow Up
	Project presentations
	Lectures
	Project Deliverable due date
	Vacation / No Class