Lecture 4

Bayesian Decision theory

University of Amsterdam

Features: recapitulation

A feature X_i: a certain type of observation or measurement

- A particular value of X_i (instantiation) is denoted x_i

Example

Iris example: petal length, sepal length, ...

Measurement (sample) vector $\mathbf{x} = (X_1 = x_1, \ldots, X_d = x_d)^\top$ describes measurements of d features during an experiment

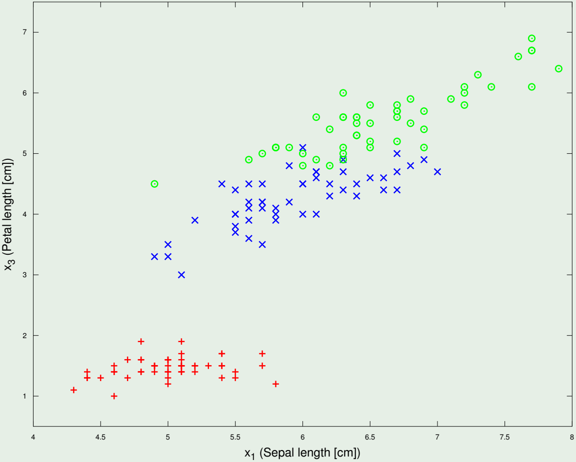
- Simplified notation: $\mathbf{x} = (x_1, \ldots, x_d)^\top$
- By measuring $x_1, \ldots, x_d$ of a vector $\mathbf{x}$, we draw a sample

Feature space: The set of all possible measurements

- In continuous domains, a d-dimensional vector is a point in a d-dimensional Euclidean space $\mathbb{R}^d$

University of Amsterdam

Example: Iris classification



Probabilistic Modelling

We are interested in how random variables are informative of each other. This can be got from the joint probability of random variables:

$$p(X = x, Y = y, \dots), \quad (1)$$

which we will generally write more compactly as

$$p(x, y, \dots) \quad (2)$$

Example

The probability that an iris should be an iris versicolor, have petals of 3cm and sepals of 5cm length, $p(\mathcal{C} = \mathcal{C}_2, X = 3, Y = 5)$.

Marginalisation

This allows us to answer questions such as "*What is the probability of seeing a particular value of x, if I don't know anything else?*" (marginal probability)

$$p(x) = \int p(x, y)dy \tag{3}$$

or, in the case of discrete (categorical) variables

$$p(x) = \sum_{y} p(x, y) \tag{4}$$

Example

What is the probability that an iris should be an "iris versicolor"?

Conditional probability

What is the probability of seeing a particular value for y, if x is known? (conditional probability)

$$p(y|x) = \frac{p(x, y)}{p(x)} \tag{5}$$

where $p(x)$ can be obtained by marginalisation.

Example

The probability that an Iris should be an Iris Versicolor and have petals of 3cm, given that its sepals are 5cm long.

Class-Conditional Probability

The conditional probability distribution $p(\mathbf{x}|\mathcal{C}_k, \boldsymbol{\theta})$ specifies with what probability we can draw a particular sample $\mathbf{x}$ given the state of the system $\mathcal{C}_k$.

- We refer to the parameters of the distribution as $\boldsymbol{\theta}$

Example

The probability that a flower will have 3cm long petals and 5cm long sepals if it's an Iris Versicolor.

Likelihood

We consider that our measurements are i.i.d., so that the total probability of all data points is

$$p(\mathbf{X}, \mathbf{t}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}^{(i)}, t^{(i)}|\boldsymbol{\theta}) \tag{6}$$

In machine learning, this quantity is often considered as a function of the parameters $\boldsymbol{\theta}$, since the data is fixed anyway. It is then called the *likelihood*.

$$p(\mathbf{X}, \mathbf{t}|\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}) \tag{7}$$

and the log-likelihood is

$$\log p(\mathbf{X}, \mathbf{t}|\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}, t^{(i)}|\boldsymbol{\theta}) \tag{8}$$

Probabilistic Modelling

Extending this to larger numbers of variables, and allowing some variables to never be observed (*latent* or *hidden* variables) makes this very powerful.

The learning process is reduced to finding a description of the joint probability distribution

- Histogram-based **non-parametric model**
- Functional representation **parametric model**

Probabilistic Modelling

Extending this to larger numbers of variables, and allowing some variables to never be observed (*latent* or *hidden* variables) makes this very powerful.
The learning process is reduced to finding a description of the joint probability distribution

- Histogram-based **non-parametric model**
- Functional representation **parametric model**

Naive Bayes

Naive Bayes: Assume all data dimensions are independent given the class

$$p(\mathbf{x}|\mathcal{C}) = p(x_1|\mathcal{C}) \cdots p(x_N|\mathcal{C})$$
$$= \prod_i p(x_i|\mathcal{C})$$

Features:

- Scales linearly in the number of features
- Overly confident is features are not independent
- Performs surprisingly well in practice
- Beware: nothing Bayesian about Naive Bayes
- Notice: conditional independence $\neq$ marginal independence

$$p(x_1, \ldots, x_n) = \sum_{\mathcal{C}} p(x_1|\mathcal{C}) \cdots p(x_N|\mathcal{C}) p(\mathcal{C}) \neq p(x_1) \cdots p(x_N)$$

Naive Bayes

Naive Bayes: Assume all data dimensions are independent given the class

$$
\begin{aligned}
p(\mathbf{x}|\mathcal{C}) &= p(x_1|\mathcal{C}) \cdots p(x_N|\mathcal{C}) \\
&= \prod_i p(x_i|\mathcal{C})
\end{aligned}
$$

Features:

- Scales linearly in the number of features
- Overly confident is features are not independent
- Performs surprisingly well in practice
- Beware: nothing Bayesian about Naive Bayes
- Notice: conditional independence $\neq$ marginal independence

$$
p(x_1, \ldots, x_n) = \sum_{\mathcal{C}} p(x_1|\mathcal{C}) \cdots p(x_N|\mathcal{C}) p(\mathcal{C}) \neq p(x_1) \cdots p(x_N)
$$

Maximising entropy

Entropy is a measure of information content:

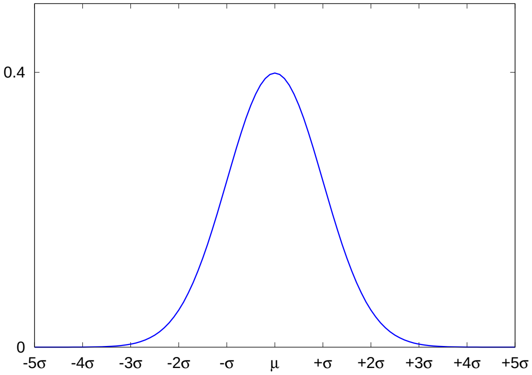
- Most informative description: maximal entropy
- Most informative PDF with parameters *mean* and *variance*:

  Gaussian distribution

- Using a Gaussian distribution basically means "I know mean and variance, and nothing more"
  - Argument for why models based on Gaussian are successful

The Gaussian or Normal distribution

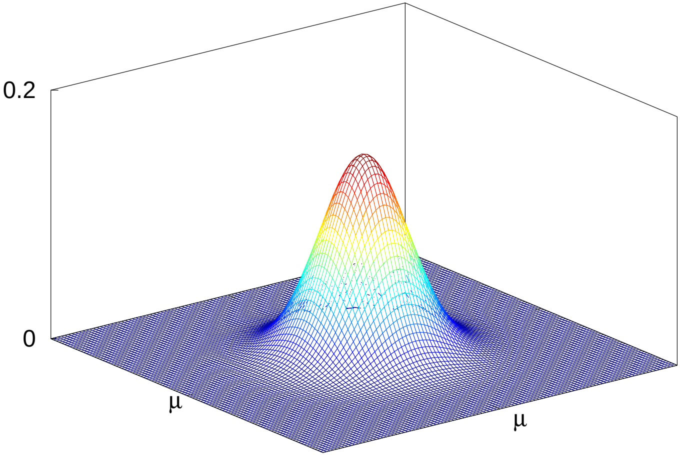
$$\mathcal{N}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp - \frac{(x-\mu)^2}{2\sigma^2} \tag{9}$$



The Gaussian or Normal distribution

UNIVERSITY OF AMSTERDAM

$$\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/1}|\boldsymbol{\Sigma}|^{1/2}} \exp -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \quad (10)$$



Central Limit Theorem

The central limit theorem can informally be stated as follows:

The Central Limit Theorem

The sum of a sufficiently large number of independent, identically distributed variables with finite variance will have an approximately Gaussian distribution.

Notice that no assumption is made about the distribution of these variables

Gaussian: Ease of manipulation

Other reason for using the Gaussian: ease of use.

- The sum of normally distributed variables is normally distributed
- The product of two normal distributions is a normal distribution
- The convolution of two normal distributions is a normal distribution

Maximum Likelihood learning

Find the parameters that maximise the likelihood function

- Results in a simple optimisation
- Prone to overfitting
- Regularisation is generally required
  - Limiting model complexity
  - *Weight decay* or *parameter shrinkage*
  - Laplace smoothing

Maximum A Posteriori (MAP) learning

Instead of learning the parameters that maximise the likelihood, why not learn the most likely parameters? Using Bayes' rule, we have:

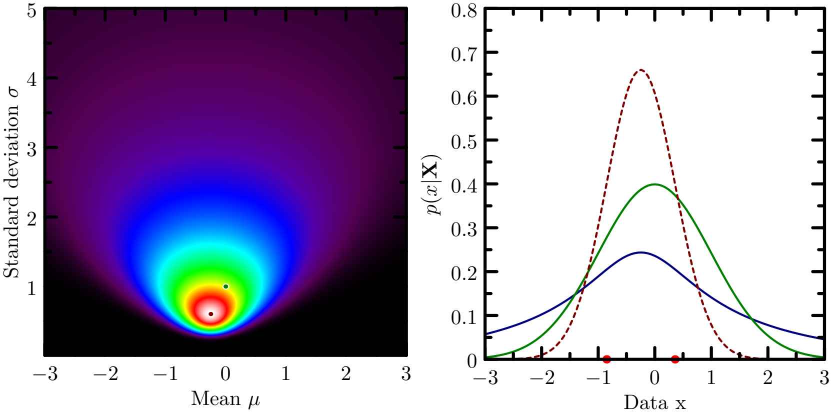
$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{x})d\boldsymbol{\theta}} \quad (11)$$

This requires us to place a prior over the parameter values

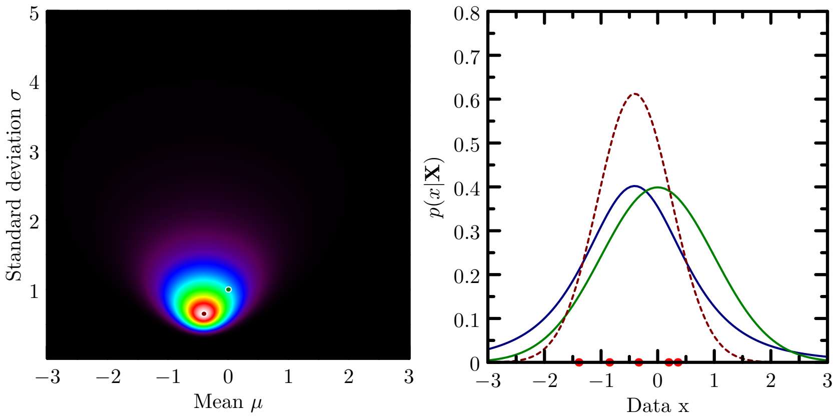
- Any prior is possible, choose prior to reflect prior knowledge
- If we use a Gaussian distribution with zero mean, this is equivalent to ML learning with parameter shrinkage
- The denominator is often intractable to compute but is constant, so that

$$\arg\max_{\boldsymbol{\theta}} \frac{p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{x})d\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \quad (12)$$

The Bayesian approach

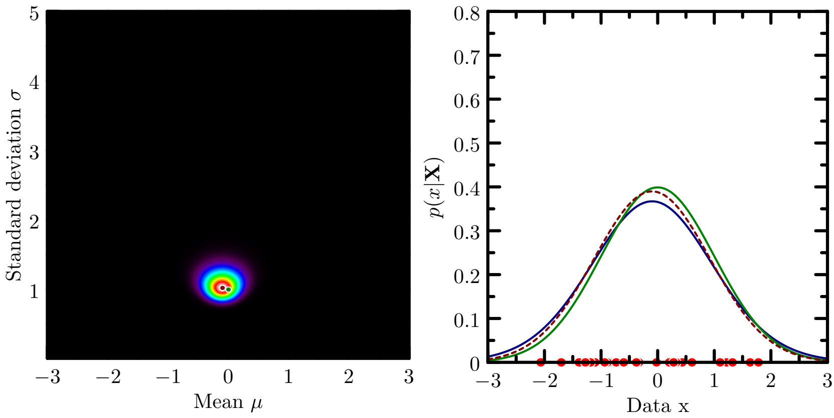


The Bayesian approach

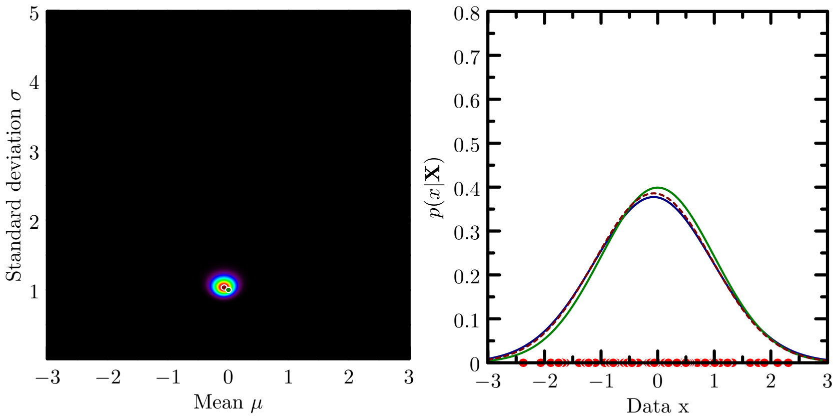


The Bayesian approach

University of Amsterdam



The Bayesian approach



The Bayesian approach

In fact, we're not really interested in knowing the original distribution that "generated" the data

- We'll never know that anyway

What we really want to do, is to use the knowledge that we have in an optimal way. That is, we want

$$p(t|\mathbf{x}, \mathbf{X}, \mathbf{t}) = \int p(t|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{t}) \mathrm{d}\boldsymbol{\theta} \tag{13}$$

In effect, we consider all the models (of the form that we have chosen beforehand) that could have generated the data, and weigh their prediction according to how probable they are.

Decision threshold

Classification: obtain a feature vector $\mathbf{x}$ and predict the corresponding class $\mathcal{C}$

Example

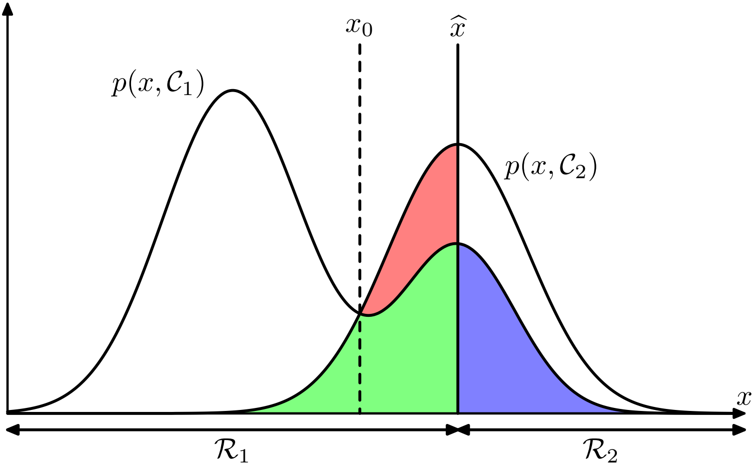
Given an X-ray image, predict the health state of the person

Bayesian decision rule: assign an observation $\mathbf{x}$ to class $\mathcal{C}_i$ if

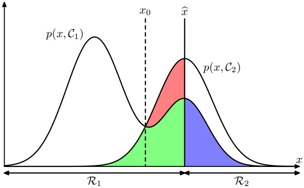
$$p(\mathcal{C}_i|\mathbf{x}) > p(\mathcal{C}_j|\mathbf{x}) \qquad \forall j \neq i \tag{14}$$

Minimising the misclassification rate

University of Amsterdam



Minimising the misclassification rate



From Bayes' rule, we have that

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)} \quad (15)$$

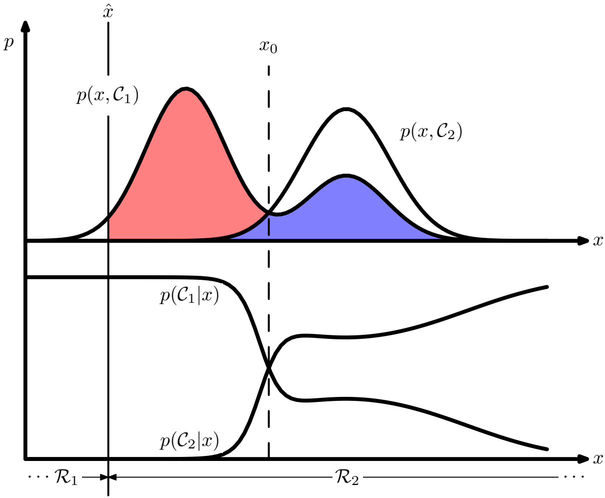
We want to minimise the probability of a mistake, that is:

$$p(\text{mistake}) = p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \quad (16)$$

$$= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2) d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1) \quad (17)$$

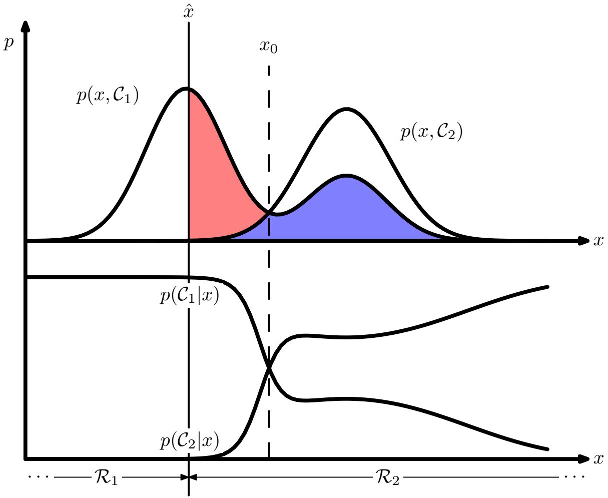
Since $p(\mathbf{x}, \mathcal{C}_k) = p(\mathcal{C}_k|\mathbf{x})p(\mathbf{x})$ and $p(\mathbf{x})$ is the same in both terms, $p(\text{mistake})$ is minimal if each point $\mathbf{x}$ is assigned to the class for which $p(\mathcal{C}_k|\mathbf{x})$ is largest.

Minimising the misclassification rate

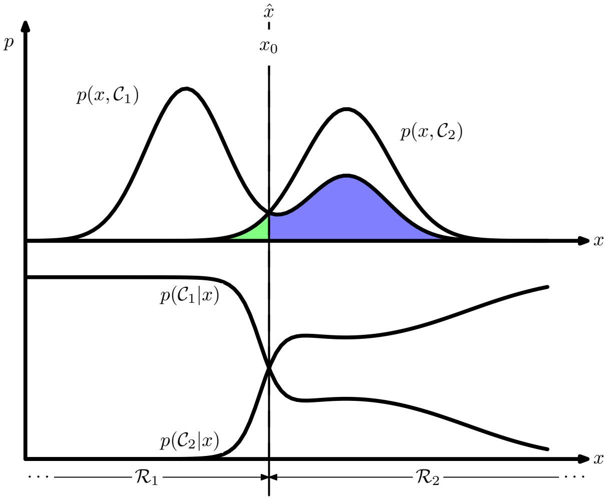


University of Amsterdam

Minimising the misclassification rate

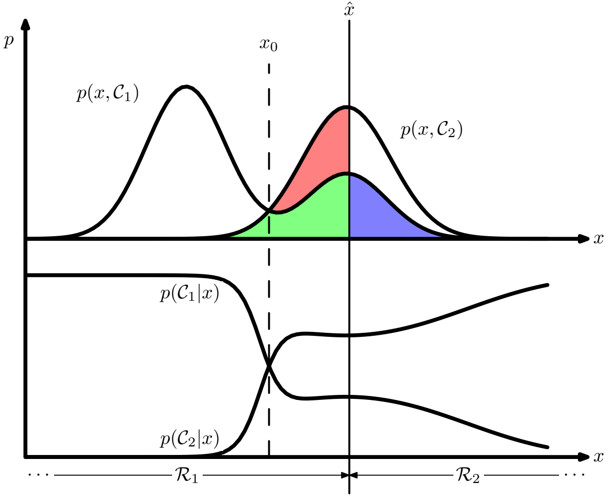


Minimising the misclassification rate

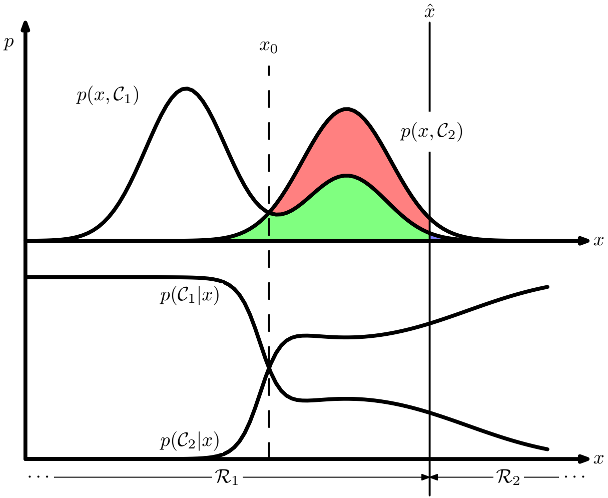


UNIVERSITY OF AMSTERDAM

Minimising the misclassification rate



Minimising the misclassification rate



Reject option

In some cases, the posterior probability $p(\mathcal{C}_k|\mathbf{x})$ of the most likely class may be far less than one.

- The regions where this is the case lead to most misclassifications

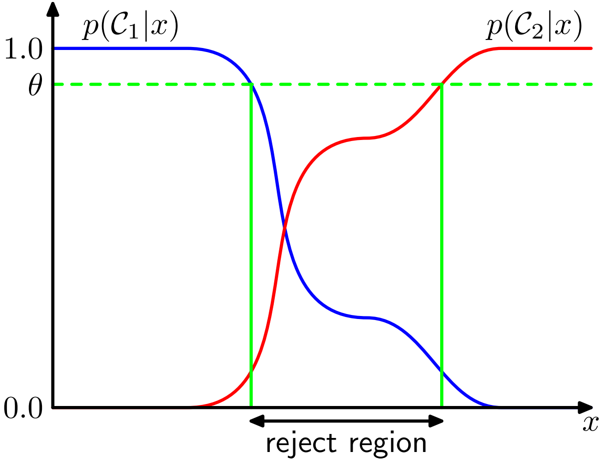
In some cases it is better to avoid making a decision when that is the case, in order to improve the performance on the examples for which a decision is made.

Example

In medical image classification, it may be suitable to automatically classify images for which we are very confident and leave the difficult cases for a human to evaluate.

The reject option

Achieved by choosing a threshold, $\boldsymbol{\theta}$, and rejecting datapoints for which the largest $p(\mathcal{C}_k|\mathbf{x}) \leqslant \boldsymbol{\theta}$.



Minimising the expected loss

In the case of unbalanced misclassification costs: loss matrix

Cancer classification example

$$L = \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \tag{18}$$

The expected loss is then given by

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) d\mathbf{x} \tag{19}$$

which is minimised by assigning each datapoint $\mathbf{x}$ to the class j for which

$$\sum_k L_{kj} p(\mathcal{C}_k | \mathbf{x}) \tag{20}$$

Measuring information

Information can be viewed as the "degree of surprise" on learning the value of a random variable. It can be quantified by considering:

- If we learn two unrelated (independent) random variables, the amount of information obtained should be the sum of the information gained by learning one of them.

$$h(x, y) = h(x) + h(y) \tag{21}$$

- From the probability of independent variables $p(x, y) = p(x)p(y)$ we have

$$h(x) \propto \log p(x) \tag{22}$$

From this we get

$$h(x) = -\log_2 p(x) \tag{23}$$

Entropy

Now suppose you transmit the value of a random variable. The average amount of information transmitted is given by

$$\mathrm{H}[x] = \mathbb{E}_{p(x)}[h(x)] = -\sum_{x} p(x) \log_2 p(x) \tag{24}$$

This is called the entropy of x. For continuous variables, this becomes the differential entropy:

$$\mathrm{H}[x] = -\int_{x} p(x) \log_2 p(x) \tag{25}$$

Properties of entropy

- The basis of the logarithm is arbitrary
  - Result differs by constant factor
  - $\log_2 \longrightarrow$ bits
  - $\ln \longrightarrow$ "nats"
- Entropy: lower bound on number of bits needed to transmit the value of a random variable (noiseless coding theorem)
- Discrete variables: maximal entropy if all possible states have the same probability

Lagrange multiplier

$$L = \sum_i p(x_i) \ln p(x_i) + \lambda\left(\sum_i p(x_i) - 1\right) \quad (26)$$

$$\Rightarrow \begin{cases} \ln p(x_i) + \frac{p(x_i)}{p(x_i)} + \lambda = 0 \\ \sum_i p(x_i) = 1 \end{cases} \quad (27)$$

Properties of entropy

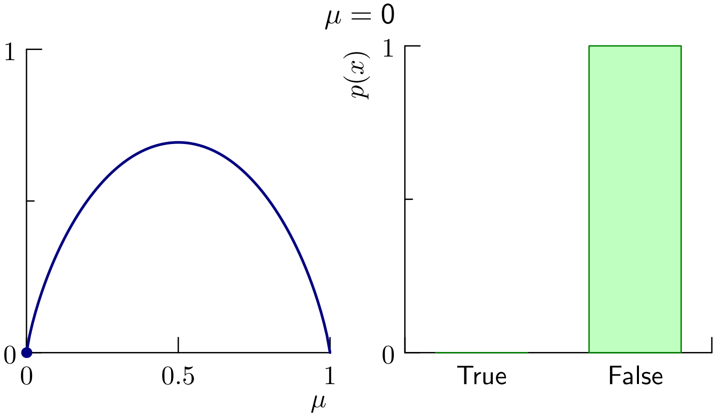
- The basis of the logarithm is arbitrary
  - Result differs by constant factor
  - $\log_2 \longrightarrow$ bits
  - $\ln \longrightarrow$ "nats"
- Entropy: lower bound on number of bits needed to transmit the value of a random variable (noiseless coding theorem)
- Discrete variables: maximal entropy if all possible states have the same probability

Lagrange multiplier

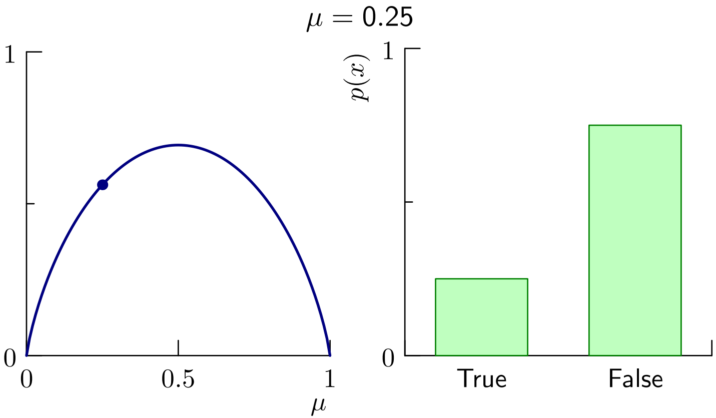
$$L = \sum_i p(x_i) \ln p(x_i) + \lambda(\sum_i p(x_i) - 1) \quad (26)$$

$$\Rightarrow \begin{cases} \ln p(x_i) + \frac{p(x_i)}{p(x_i)} + \lambda = 0 \\ \sum_i p(x_i) = 1 \end{cases} \quad (27)$$

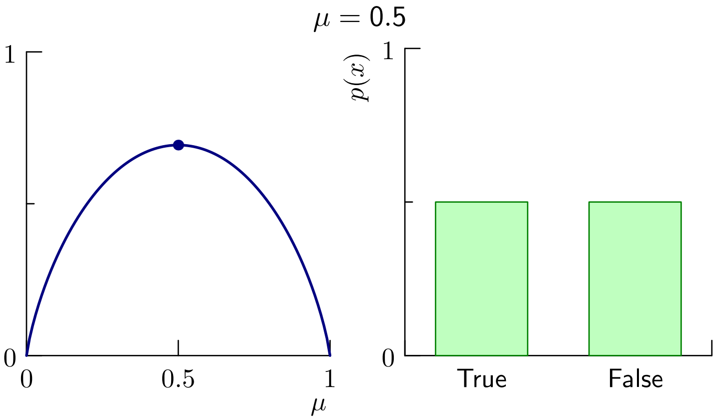
Entropy of a Bernoulli distribution



Entropy of a Bernoulli distribution

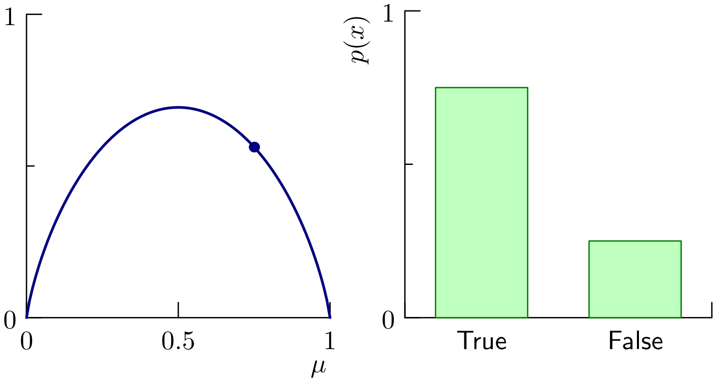


Entropy of a Bernoulli distribution

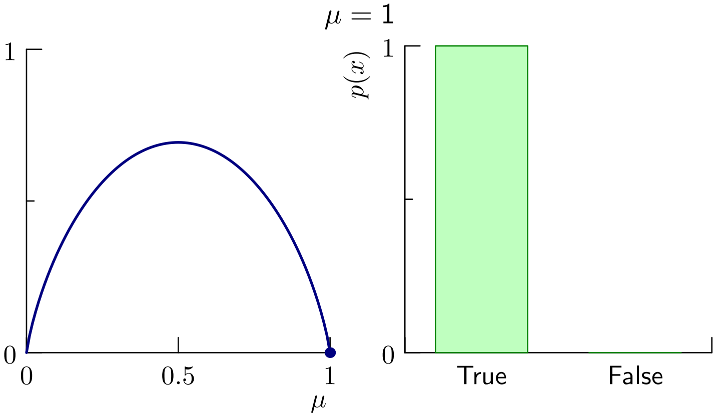


Entropy of a Bernoulli distribution

$\mu = 0.75$



Entropy of a Bernoulli distribution



Mutual Entropy

Consider two random variables, $\mathbf{x}$ and $\mathbf{y}$. If $\mathbf{x}$ is known, the additional information needed to specify $\mathbf{y}$ is given by

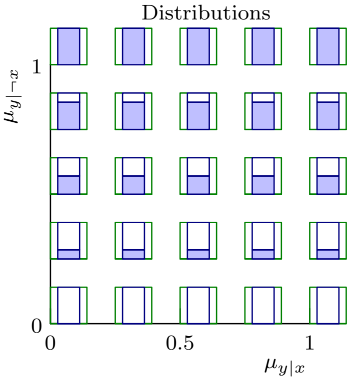
$$h(\mathbf{y}|\mathbf{x}) = -\ln p(\mathbf{y}|\mathbf{x}) \tag{28}$$

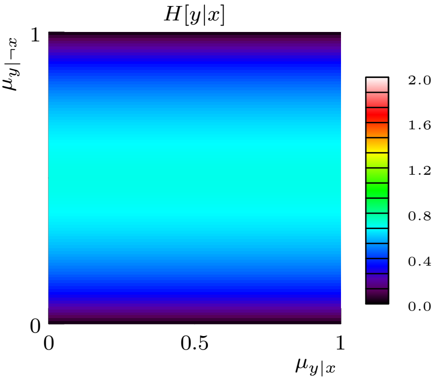
So that the average additional information needed to specify $\mathbf{y}$ is

$$\mathrm{H}[\mathbf{y}|\mathbf{x}] = -\iint p(\mathbf{x}, \mathbf{y}) \ln p(\mathbf{y}|\mathbf{x}) d\mathbf{y} d\mathbf{x} \tag{29}$$

so that

$$\mathrm{H}[\mathbf{x}, \mathbf{y}] = \mathrm{H}[\mathbf{y}|\mathbf{x}] + \mathrm{H}[\mathbf{x}] \tag{30}$$


Distributions
1
0
0
0.5
1
$\mu_{y|\neg x}$
$\mu_{y|x}$


$H[y|x]$
1
0
0
0.5
1
$\mu_{y|\neg x}$
$\mu_{y|x}$
2.0
1.6
1.2
0.8
0.4
0.0

KL Divergence

University of Amsterdam

Consider a random variable, $\mathbf{x}$, with unknown distribution $p(\mathbf{x})$. If we approximate this with $q(\mathbf{x})$ and use this distribution to transmit the value of $\mathbf{x}$, the additional (wasted) information used (in nats) is

$$\mathrm{KL}(p||q) = -\int p(\mathbf{x}) \ln q(\mathbf{x}) d\mathbf{x} - \left(-\int p(\mathbf{x}) \ln p(\mathbf{x}) d\mathbf{x} \right) \quad (31)$$

$$= -\int p(\mathbf{x}) \ln \left(\frac{q(\mathbf{x})}{p(\mathbf{x})} \right) d\mathbf{x} \quad (32)$$

This is the relative entropy or Kullback-Leibler divergence between $p(\mathbf{x})$ and $q(\mathbf{x})$.

KL Divergence

Properties

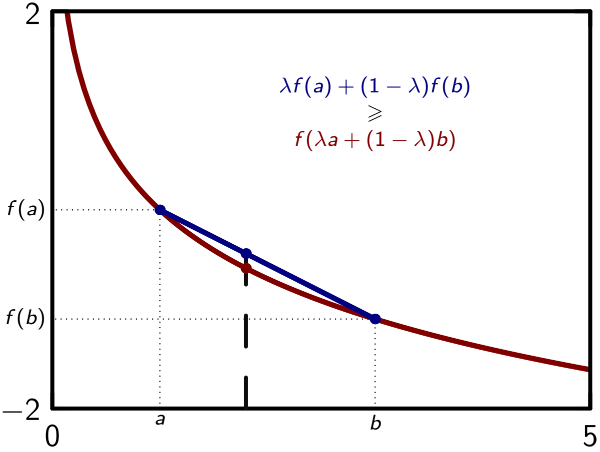
- Measure of difference between probability distributions
- Not a metric:

$$\mathrm{KL}(p||q) \neq \mathrm{KL}(q||p) \tag{33}$$

- Basis for approximations
  - Minimising $\mathrm{KL}(p||q)$ or $\mathrm{KL}(q||p)$ leads to different approximations

KL Divergence is positive

Convex function: $y = -\ln x$



KL Divergence is positive (II)

In general

$$\sum_i \lambda_i f(x_i) \geqslant f\left(\sum_i \lambda_i x_i\right) \text{ where } \sum_i \lambda_i = 1 \tag{34}$$

This is known as Jensen's inequality. If we take λ_i to be probabilities:

$$\mathbb{E}[f(x)] \geqslant f(\mathbb{E}[x]) \tag{35}$$

for any convex function $f(x)$. For KL-divergence:

$$-\int p(\mathbf{x}) \ln\left[\frac{q(\mathbf{x})}{p(\mathbf{x})}\right] d\mathbf{x} \geqslant -\ln \int p(\mathbf{x}) \left[\frac{q(\mathbf{x})}{p(\mathbf{x})}\right] d\mathbf{x} = \ln 1 = 0 \tag{36}$$

Mutual Information

Now imagine the distribution of $p(\mathbf{x}, \mathbf{y})$. If the variables are independent,

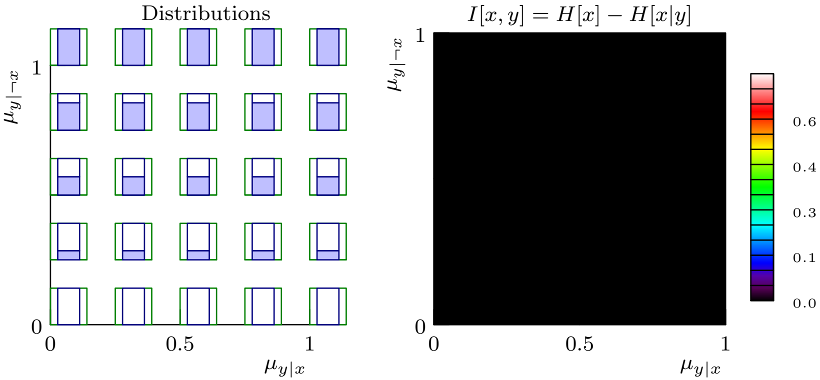
$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}). \quad (37)$$

If they are not independent, we can compute how "close" they are to being independent:

$$\begin{aligned} \mathrm{I}[\mathbf{x}, \mathbf{y}] &\equiv \mathrm{KL}(p(\mathbf{x}, \mathbf{y}) || p(\mathbf{x})p(\mathbf{y})) \quad (38) \\ &= -\int p(\mathbf{x}, \mathbf{y}) \ln \left(\frac{p(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x}, \mathbf{y})} \right) d\mathbf{x} d\mathbf{y} \quad (39) \end{aligned}$$

This is the mutual information between $\mathbf{x}$ and $\mathbf{y}$.

$$\mathrm{I}[\mathbf{x}, \mathbf{y}] = \mathrm{H}[\mathbf{x}] - \mathrm{H}[\mathbf{x}|\mathbf{y}] = \mathrm{H}[\mathbf{y}] - \mathrm{H}[\mathbf{y}|\mathbf{x}] \quad (40)$$


Distributions
1
0
0
0.5
1
$\mu_{y|\neg x}$
$\mu_{y|x}$
$I[x, y] = H[x] - H[x|y]$
1
0
0
0.5
1
$\mu_{y|\neg x}$
$\mu_{y|x}$
0.6
0.4
0.3
0.1
0.0

Wrap up

Today, we saw:

- Probabilistic modelling
- How to learn model parameters (Bishop, p. 28–31)
- How to make decision based on the model (Bishop, p. 38–42)
- Entropy, Conditional Entropy (Bishop, p. 48–52,54)
- KL Divergence, Mutual Information (Bishop, p. 55,57)

Coming up:

- Lagrange multipliers