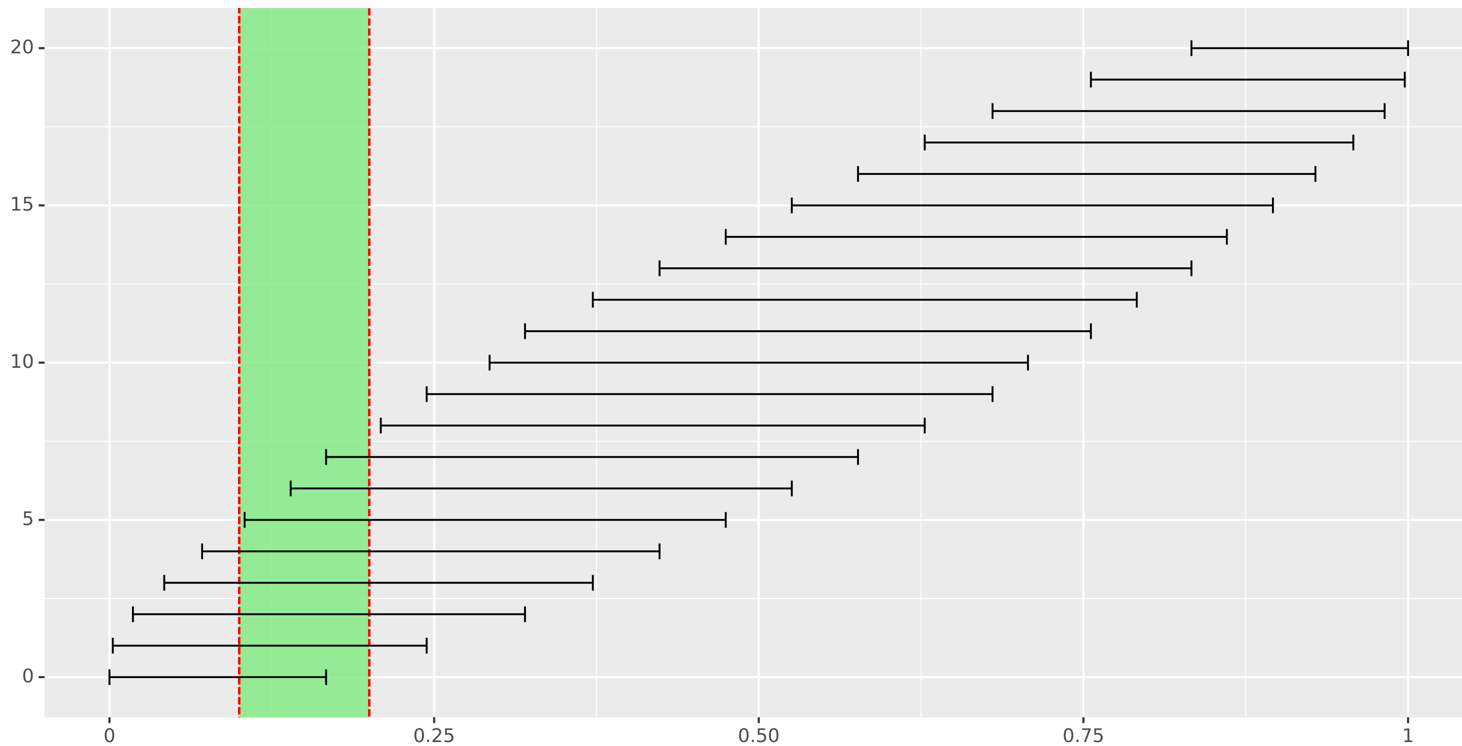
LTAT.02.004 Machine Learning II

Bayesian methods

Sven Laur
University of Tartu

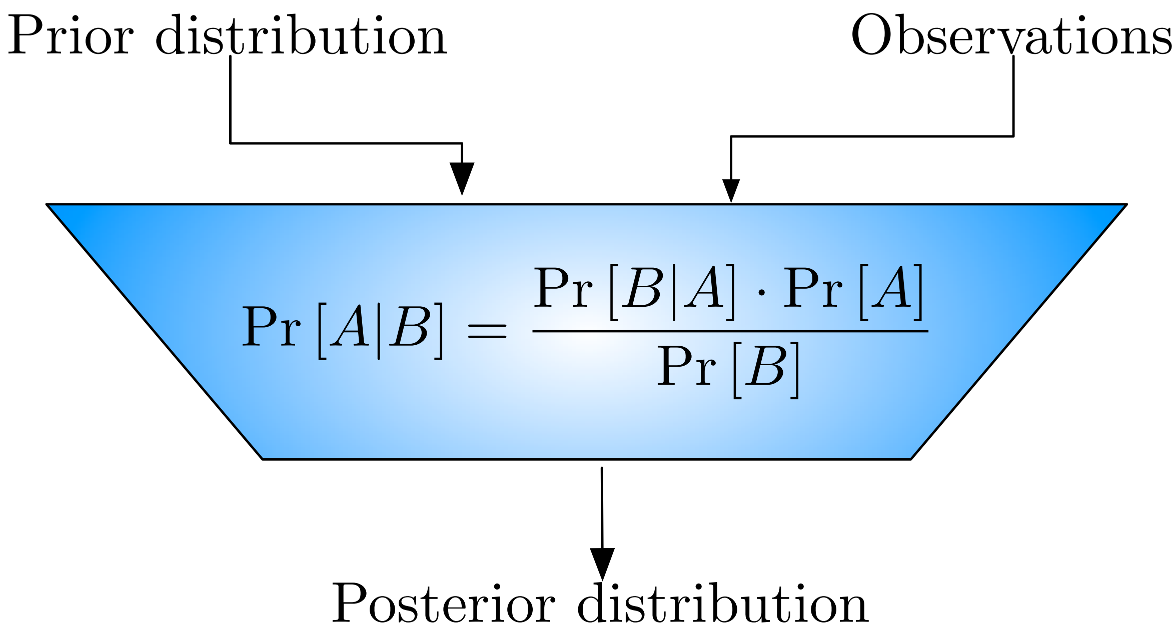
Bayesian methods

Confidence intervals vs background knowledge



- ▷ Confidence intervals do not capture background knowledge $p \in [0.1, 0.2]$.
- ▷ Thus we must accept absurd or suboptimal parameter estimations.

Bayesian inference procedure



- Prior distribution $\Pr\left[A\right]$ encodes the background knowledge
- The model $\Pr\left[B|A\right]$ determines how the posterior $\Pr\left[A|B\right]$ is updated

Prior and likelihood

Likelihood $\mathcal{L}(\mathcal{D}|\mathcal{M})$ is a probability of observations $\mathcal{D}$ when the data generation model $\mathcal{M}$ is fixed. The model is fixed by the set of parameters.

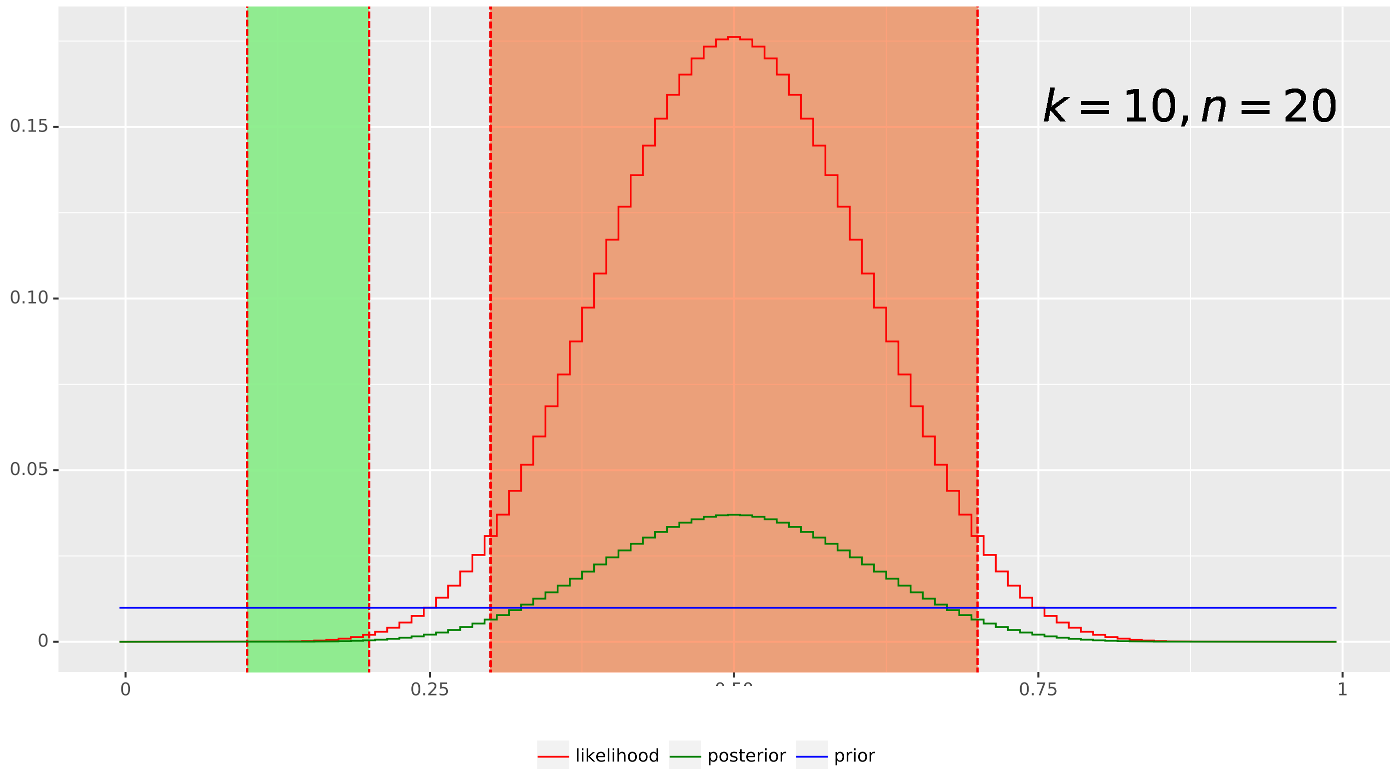
For coin flipping experiment the number of ones k is the observation and the coin bias p is the model paramater and thus

$$\mathcal{L}[k|p] = \binom{n}{k} p^k (1-p)^{n-k}$$

Prior is a distribution over models that encodes our preferences of models before we observe any data.

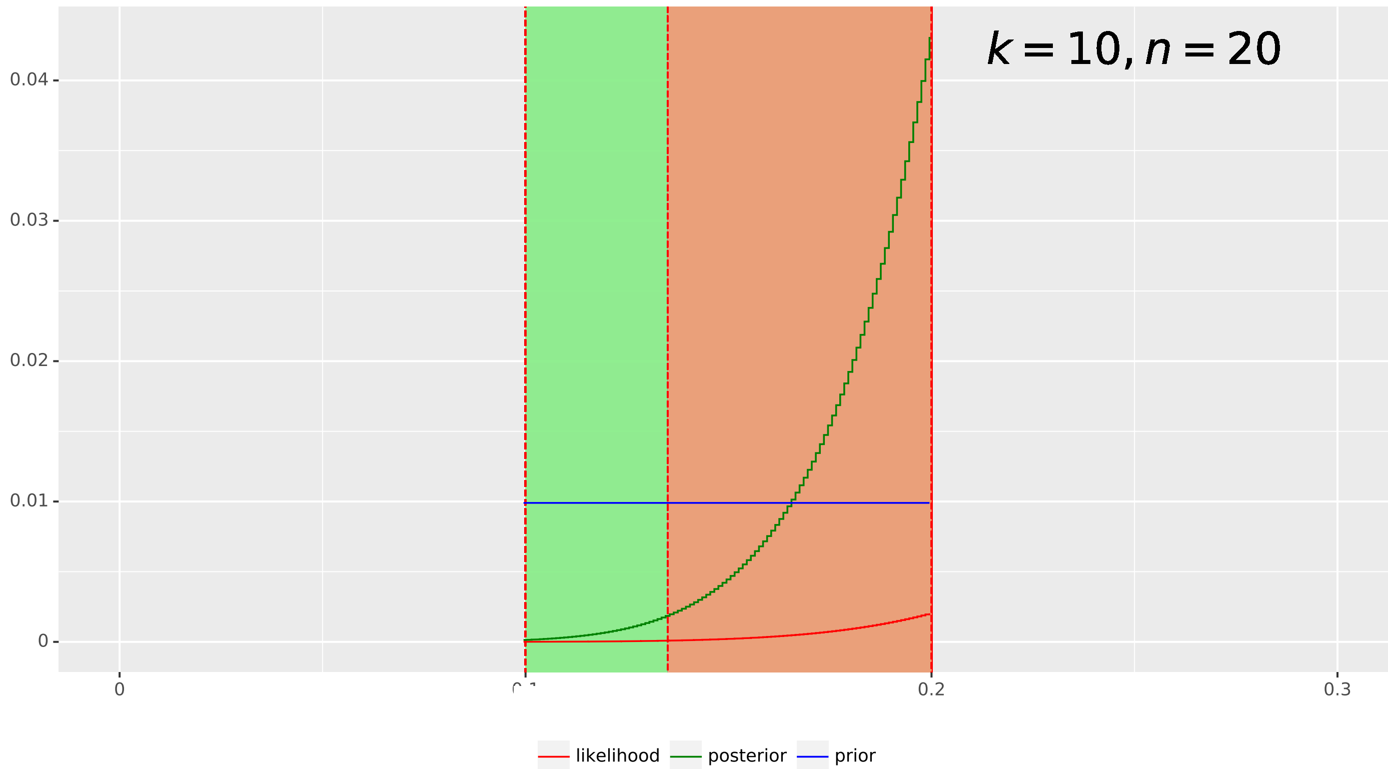
- ▷ Uninformative prior assigns uniform probability to all models.
- ▷ Uninformative prior is not well-defined for continuous parameters.

Posterior of an uninformed person



- ▷ With no preferences the posterior is concentrated around 0.5.
- ▷ Credibility interval $p \in [0.3, 0.7]$ contains 95% of posterior probability.

Posterior of an informed person



- ▷ With preferences the posterior is concentrated to the left of 0.2.
- ▷ Credibility interval $p \in [0.135, 0.2]$ contains 95% of posterior probability.

Beta distribution as a posterior

By increasing the number of grid points in the non-informative prior we reach a continuous distribution with a density function

$$p[p|k] = \frac{\Gamma(n+2)}{\Gamma(k+1)\Gamma(n-k+1)} \cdot p^k(1-p)^{n-k} \ .$$

This distribution is known as *beta distribution* Beta$(\alpha = k+1, \beta = n-k+1)$. The parameter value that maximises the posterior is

$$p_* = \frac{\alpha - 1}{\beta - \alpha} = \frac{k}{n} \ .$$

Maximum likelihood principle

If I have no background information to prefer one model to another then

$$\Pr\left[\mathcal{M}_i\right] = const$$

and thus

$$\Pr\left[\mathcal{M}_i|\mathcal{D}\right] = const \cdot \Pr\left[(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)|\mathcal{M}_i\right]$$

As a result I should choose a model that maximises *likelihood*

$$\Pr\left[(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)|\mathcal{M}_i\right]$$

The same principle is also applicable if the number of models is infinite.

Maximum a posteriori principle

Sometimes, we have extra background knowledge that makes some models more likely than the others:

$$\Pr\left[\mathcal{M}_i\right] \neq const$$

Then the model with largest likelihood is suboptimal choice and we should take a model with highest posterior probability

$$\Pr\left[\mathcal{M}_i|\mathcal{D}\right] \rightarrow \max \quad .$$

This method is known as *maximum a posteriori principle*.

In most cases, MAP estimates are defined so that they are *numerically and statistically more stable* than ML estimates.

Dice throwing vs coin flipping

A behaviour of a dice with faces $\{1, \ldots, m\}$ is determined by probabilities

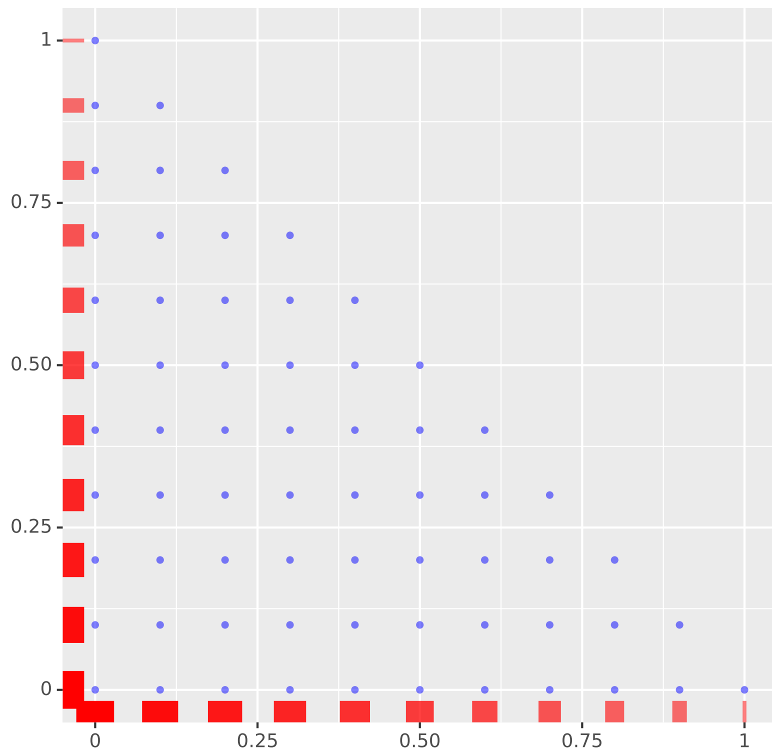
$$p_1 = \Pr[D_i = 1], \quad \ldots, \quad p_m = \Pr[D_i = m]$$

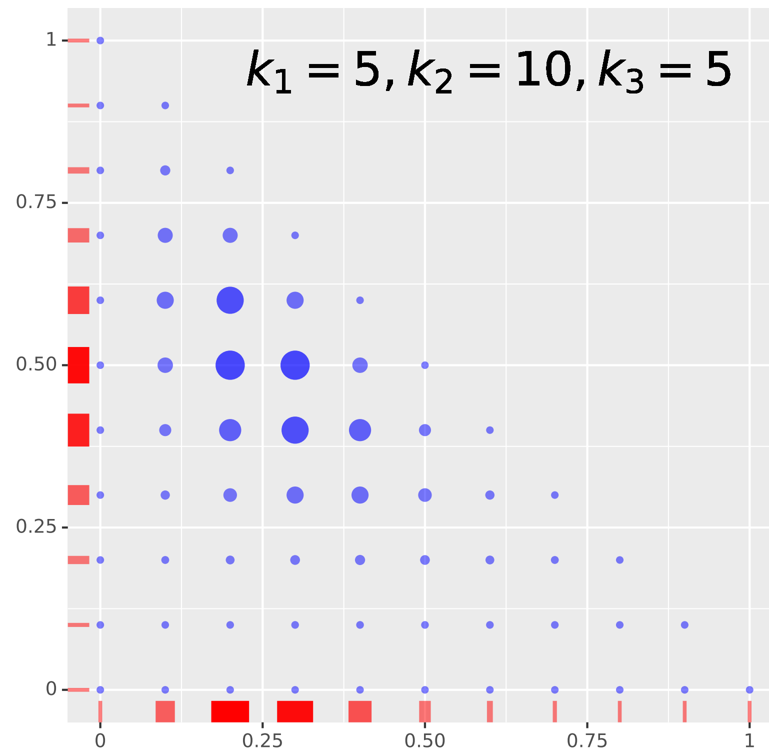
Reduction to coin flipping

- ▷ Let B_i denote the event that $D_i = j$.
- ▷ Then $B_1, \ldots, B_n$ is a coinflipping sequence with bias $\Pr[B_i = 1] = p_j$.
- ▷ ~~Non-informative prior for dice throwing goes to the non-informative prior.~~
- ▷ Informative priors can be marginalised to the right format.
- ▷ The same reduction can be done for all faces of the dice.

Caution: Marginal posteriors do not determine the full posterior in general.

Illustration





- ▷ Uniform prior over parameter pairs yields non-uniform marginal priors.
- ▷ The joint MAP estimate coincides with the marginal MAP estimates.

Dirichlet distribution as a posterior

By increasing the number of grid points in the non-informative prior over simplex we reach a continuous distribution with a density function

$$p[p_1, \ldots, p_m | k_1, \ldots, k_m] = \frac{\Gamma(n+m)}{\Gamma(k_1+1)\cdots\Gamma(k_m+1)} \cdot p_1^{k_1} \cdots p_m^{k_m} \enspace .$$

This distribution is known as *Dirichlet distribution*

$$\mathrm{Dirichlet}(\alpha_1 = k_1 + 1, \ldots, \alpha_m = k_m + 1) \enspace .$$

The parameter value that maximises the posterior is

$$p_i^* = \frac{\alpha_i - 1}{\alpha_1 + \ldots \alpha_m - m} = \frac{k_i}{n} \enspace .$$

Laplace smoothing

Assume that we throw a dice with m faces and B_i encodes the event that the dice lands on a specific face. Then it is natural to assign the maximum prior probability to the parameter value $p_* = \frac{1}{m}$.

Such prior can be defined through a following though experiment:

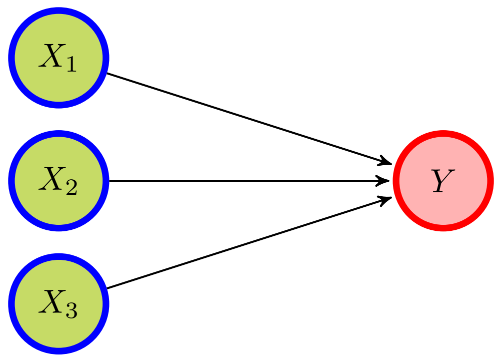
- ▷ We start with non-informative prior.
- ▷ We observe all possible outcomes of the dice α times.
- ▷ We use the resulting posterior as a prior for real observations.

Thus the posterior can be obtained by starting with non-informative prior and observing $k + \alpha$ ones among $n + m\alpha$ throws.

- ▷ The ratio $p = \frac{k+\alpha}{n+m\alpha}$ is the maximal aposteriori estimate for p.

Supervised learning

Model behind naive Bayes classifier



Underlying class value determines observed attributes

- ▷ Each attribute X_i is binary
- ▷ All variables are independent if class is fixed
- ▷ Sometimes we just ignore dependancies for easier modelling

Likelihood of the data

Let us assume that we know the probabilities

$$p_i = \Pr\left[X_i = 1 | Class = 0\right]$$
$$q_i = \Pr\left[X_i = 1 | Class = 1\right]$$

Then using the independence assumption we get

$$\Pr\left[X_1 = a_1, \ldots, X_n = a_n | Class = 0\right] = \prod_{i=1}^{n} p_i^{a_i}(1-p_i)^{1-a_i}$$
$$\Pr\left[X_1 = a_1, \ldots, X_n = a_n | Class = 1\right] = \prod_{i=1}^{n} q_i^{a_i}(1-q_i)^{1-a_i}$$

Prior and posterior for the class labels

Now it is straightforward to derive

$$\Pr\left[Class = 0 | \boldsymbol{X} = \boldsymbol{a}\right] = \frac{\prod_{i=1}^{n} p_i^{a_i}(1-p_i)^{1-a_i} \cdot \Pr\left[Class = 0\right]}{\Pr\left[\boldsymbol{X} = \boldsymbol{a}\right]}$$

$$\Pr\left[Class = 1 | \boldsymbol{X} = \boldsymbol{a}\right] = \frac{\prod_{i=1}^{n} q_i^{a_i}(1-q_i)^{1-a_i} \cdot \Pr\left[Class = 1\right]}{\Pr\left[\boldsymbol{X} = \boldsymbol{a}\right]}$$

which gives an *odd ratio*

$$\frac{\Pr\left[Class = 0 | \boldsymbol{X} = \boldsymbol{a}\right]}{\Pr\left[Class = 1 | \boldsymbol{X} = \boldsymbol{a}\right]} = \frac{\Pr\left[Class = 0\right]}{\Pr\left[Class = 1\right]} \cdot \frac{\prod_{i=1}^{n} p_i^{a_i}(1-p_i)^{1-a_i}}{\prod_{i=1}^{n} q_i^{a_i}(1-q_i)^{1-a_i}}$$

The resulting classifier is a linear classifer

By taking logarithm form the odd ratio we get

$$\log\left(\frac{\Pr\left[Class = 0|\boldsymbol{X} = \boldsymbol{a}\right]}{\Pr\left[Class = 1|\boldsymbol{X} = \boldsymbol{a}\right]}\right) = w_0 + \sum_{i=1}^{n} w_i a_i$$

where

$$w_0 = \log\left(\frac{\Pr\left[Class = 0\right]}{\Pr\left[Class = 1\right]}\right) + \sum_{i=1}^{n} \log\left(\frac{1 - p_i}{1 - q_i}\right)$$

$$w_i = \log\left(\frac{p_i}{1 - p_i} \cdot \frac{1 - q_i}{q_i}\right)$$

How to train the classifier?

A frequentistic approach is to fix probabilities from the training sample

$$p_i = \frac{\#\{\text{data points form class 0 with } X_i = 1\}}{\#\{\text{data points form class 0}\}}$$

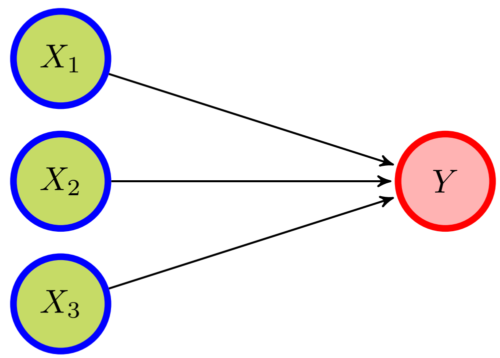
$$q_i = \frac{\#\{\text{data points form class 1 with } X_i = 1\}}{\#\{\text{data points form class 1}\}}$$

However if some value does not occur for X_i in the training sample we get overly confident results. Thus, Bayesian mean estimate is better alternative

$$p_i = \frac{\#\{\text{data points form class 0 with } X_i = 1\} + 1}{\#\{\text{data points form class 0}\} + 2}$$

$$q_i = \frac{\#\{\text{data points form class 1 with } X_i = 1\} + 1}{\#\{\text{data points form class 1}\} + 2}$$

Linear regression models

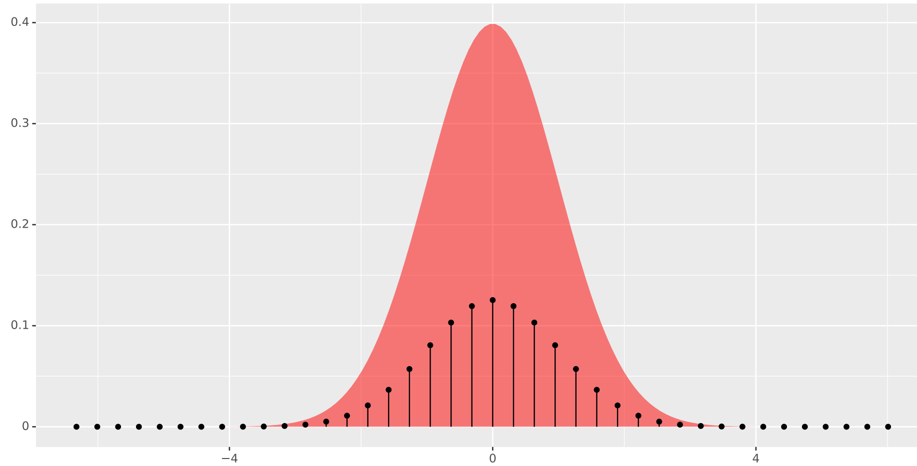


Repeated experiments with external controls

- ▷ We assume that y_i depends only on the values of $x_{i1}, \ldots, x_{i\ell}$
- ▷ A linear model assumes $y_i = w_1 x_{i1} + \cdots + w_\ell x_{i\ell} + w_0 + \varepsilon_i$.
- ▷ All error terms ε_i are assumed to be independent.
- ▷ All error terms ε_i are drawn from a normal distribution $\mathcal{N}(0, \sigma)$.

Univariate normal distribution

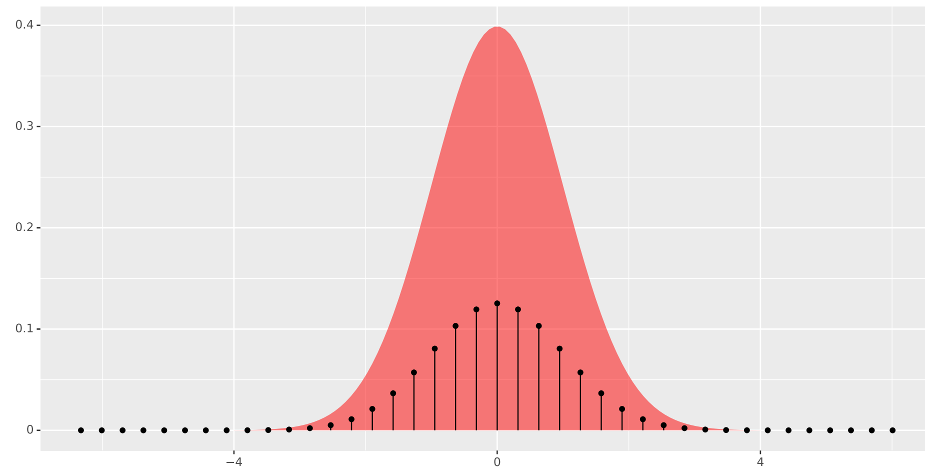
Probability density function



Definition. A real-valued random variable X comes from a continuous distribution with *a probability density function* $p : \mathbb{R} \to \mathbb{R}^+ \cup \{0\}$ if the following limit exists for any $x \in \mathbb{R}$:

$$p(x) = \lim_{\Delta x \to 0^+} \frac{\Pr\left[x - \Delta x \leq X \leq x + \Delta x\right]}{2 \cdot \Delta x} \enspace .$$

Probability mass function

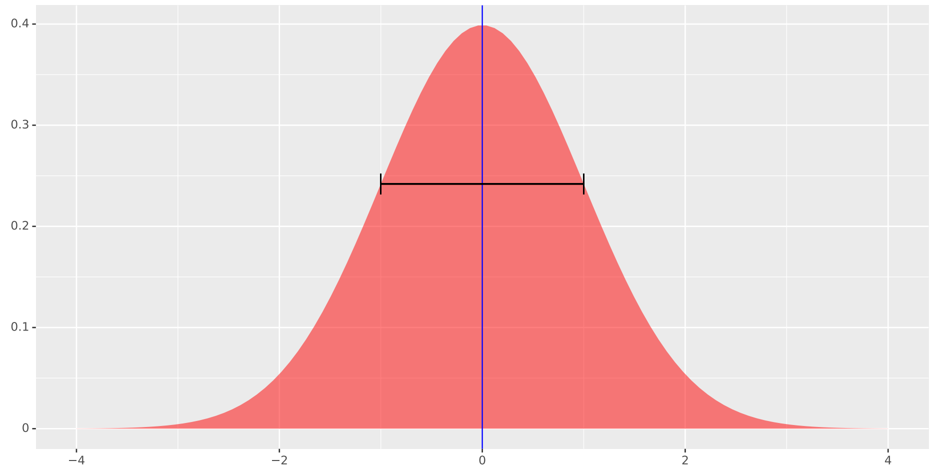


Definition. A real-valued random variable X comes from a discrete distribution with *a probability mass function* $p : \mathbb{R} \to \mathbb{R}^+ \cup \{0\}$ defined as

$$p(x) = \Pr[X = x] = \lim_{\Delta x \to 0^+} \Pr[x - \Delta x \leq X \leq x + \Delta x]$$

if there exist a sequence $(x_i)_{i=1}^{\infty}$ such that $p(x_1) + \ldots + p(x_i) + \ldots = 1$.

Standard normal distribution



Standard normal distribution $\mathcal{N}(\mu = 0, \sigma = 1)$ is a continuous distribution with a probability density function

$$p(x) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{x^2}{2}\right)$$

The mean value $\mu = 0$ and variance $\sigma^2 = 1$ for this distribution.

Univariate normal distribution

Definition. A random variable y is distributed according to a normal distribution $\mathcal{N}(\mu = a, \sigma = b)$ if it can be expressed

$$y = bx + a$$

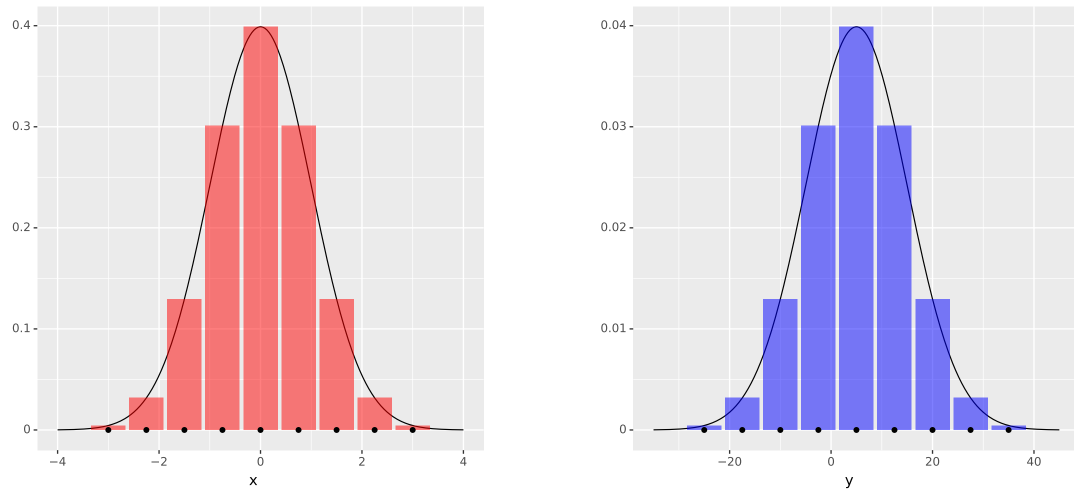
where x is distributed according to standardised normal distribution $\mathcal{N}(0, 1)$.

The corresponding probability density functions is

$$p[y|\mu, \sigma] = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right)$$

and the mean value μ and variance σ^2 for this distribution.

Density derivation



Let $y = ax + b$ the the relation between densities

$$p_x(x) = \sigma \cdot p_y(y)$$

follows form the fact that areas of red and blue columns must be the same.

Univariate linear regression

- Fix a set of inputs $x_1, \ldots, x_n \in \mathbb{R}$.
- A probabilistic model is defined by three coefficients $a, b, \sigma \in \mathbb{R}$.
- The model assigns a probability to outcomes $y_1, \ldots, y_n$ through the following observation generation mechanism

$$y_i = ax_i + b + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma)$$

- Consequently

$$p[\boldsymbol{y}|\boldsymbol{x}, a, b] = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{(y_i - ax_i - b)^2}{2\sigma^2}\right)$$

Maximum likelihood estimate

As usual we can find $a, b, \sigma \in \mathbb{R}$ that maximise the log-likelihood

$$\log p[\boldsymbol{y}|\boldsymbol{x}, a, b, \sigma] = const - n \log \sigma - \sum_{i=1}^{n} \frac{(y_i - ax_i - b)^2}{2\sigma^2}$$

and thus we can find a and b by minimising

$$\mathsf{MSE} = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - ax_i - b)^2 \enspace .$$

Residuals and the variance parameter

For fixed $a, b \in \mathbb{R}$ we can define predictions and residuals

$$\hat{y}_i = ax_i - b$$
$$r_i = y_i - \hat{y}_i$$

To find the optimal variance σ^2 we need to maximise

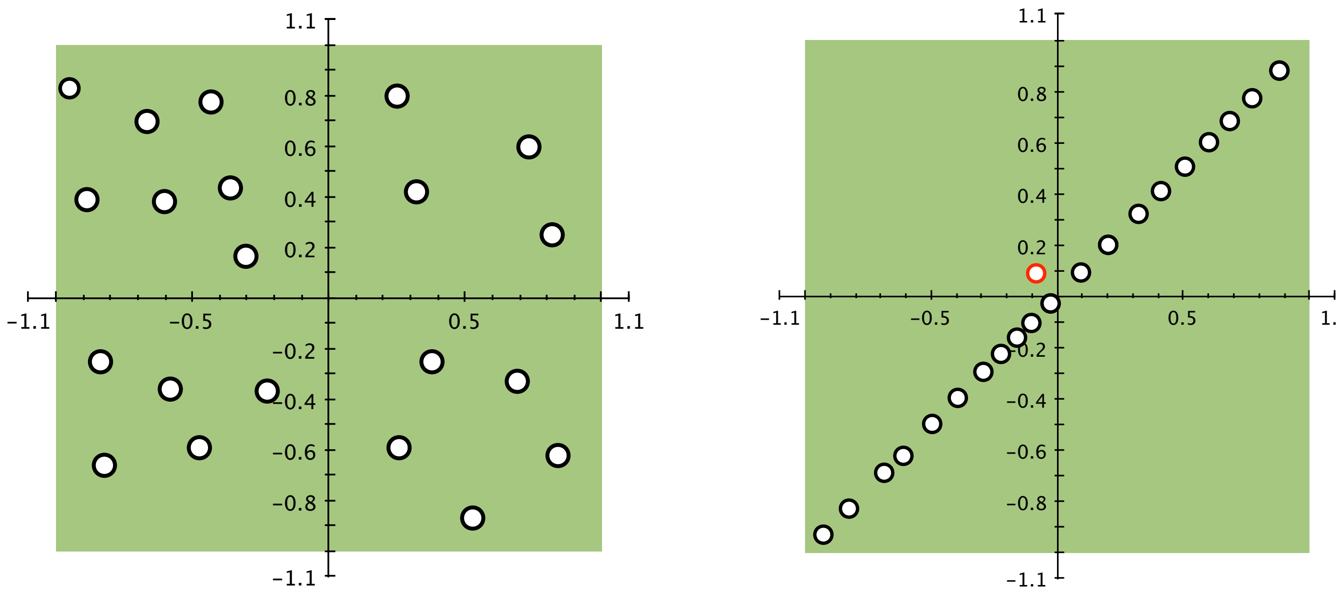
$$\log p[\boldsymbol{y}|\boldsymbol{x}, a, b, \sigma] = const - n \log \sigma - \sum_{i=1}^{n} \frac{r_i^2}{2\sigma^2}$$

The resulting solution is

$$\sigma^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Add fat tail distributions!

Input values and numerical stability

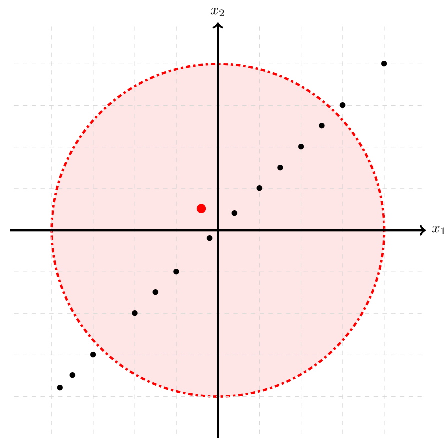


A small error in a point with big leverage can make linear regression function arbitrary large, which can lead to large test errors.

▷ In many case we know that the final output must be in fixed range.

Ridge regression

Let us seek the prediction as a function $f(\boldsymbol{x}) = w_1 x_1 + \cdots + w_k x_k$ with restriction $f(\boldsymbol{x}) \leq c$ inside a unit ball $\|\boldsymbol{x}\|_2^2 = x_1^2 + x_2^2 + \cdots + x_k^2 \leq 1$.



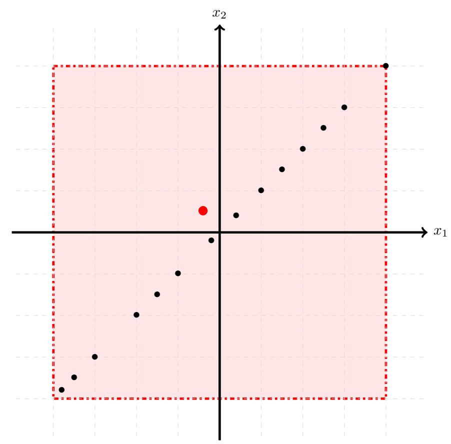
Then we should solve the following task instead:

$$\frac{1}{N} \cdot \sum_{i=1}^{N} \bigl(y_i - f(\boldsymbol{x}_i)\bigr)^2 \to \min$$

$$\text{s.t. } w_1^2 + \cdots + w_k^2 \leq c^2$$

LASSO regression

Let us seek the prediction as a function $f(\boldsymbol{x}) = w_1x_1 + \cdots + w_kx_k$ with restriction $f(\boldsymbol{x}) \leq c$ inside a unit ball $\|\boldsymbol{x}\|_\infty = \max\{|x_1|, \ldots, |x_k|\} \leq 1$.



Then we should solve the following task instead:

$$\frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i))^2 \to \min$$

$$\text{s.t.} \quad |w_1| + \cdots + |w_k| \leq c$$

Lagrange' trick

If we want to minimise $f(\boldsymbol{x})$ such that $g(\boldsymbol{x}) \leq c$ for a non-negative function $g(\cdot)$, then there exists $\lambda \geq 0$ such that the solution of the original problem is a minimum for a modified function

$$f_*(\boldsymbol{x}) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x})$$

Consequences

- ▷ We can use a penalty term $\lambda \left\|\boldsymbol{w}\right\|_1$ for rectangular area
- ▷ We can use a penalty term $\lambda \left\|\boldsymbol{w}\right\|_2^2$ for circular area