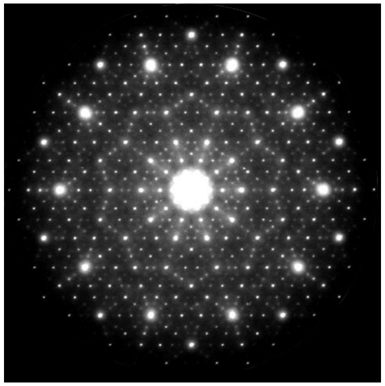
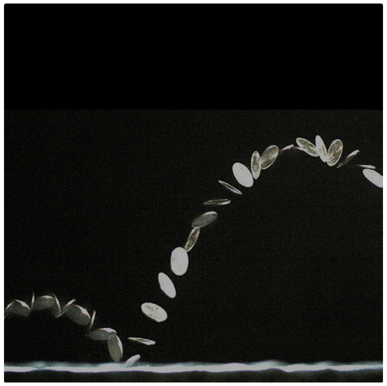
LTAT.02.004 Machine Learning II

Basics of probabilistic modelling

Sven Laur
University of Tartu

What is probability?





Probability is a measure of uncertainty which can rise in several ways

- ▷ Intrinsic uncertainty in the system
- ▷ Uncertainty caused by inherent instability of the system
- ▷ Uncertainty caused by lack of knowledge or control over the system

Frequentistic interpretation of probability

Probability is an average occurrence rate in long series of experiments.

- ▷ Law of large numbers
- ▷ Probability is a collective property
- ▷ Probabilities can be assigned only to future events

Bayesian interpretation of probability

Probability reflects persons individual beliefs on future or unknown events.

- ▷ Belief updates through the Bayes rule
- ▷ Probability is an inherently subjective property
- ▷ Probabilities can be assigned to past, present and future events

Ultra-frequentistic interpretation of probability

Events with small enough probability do not occur

- ▷ The main tool in classical statistics
- ▷ Errors in judgement does not matter if a gamma ray pulse kills us.
- ▷ One must avoid the lottery paradox in the reasoning

The goal of statistical inference

Frequentist goal

▷ The aim of statistics is to design algorithms that work well on average.

▷ For that one needs to specify probabilistic model for data sources.

▷ Confidence is the fraction of cases the algorithm works as specified.

Bayesian goal

▷ The aim of statistics is to design algorithms that allow *rational individuals* to reliably update their beliefs through Bayes formula

▷ Besides the data source model one has to provide model for initial beliefs.

▷ Correctness of an algorithm does not make sense.

Frequentistic methods

Causation between zero-one events

Assume that condition A causes the event $B = 1$ with probability p, i.e.,

$$\Pr\left[B = 1 | A\right] = p$$

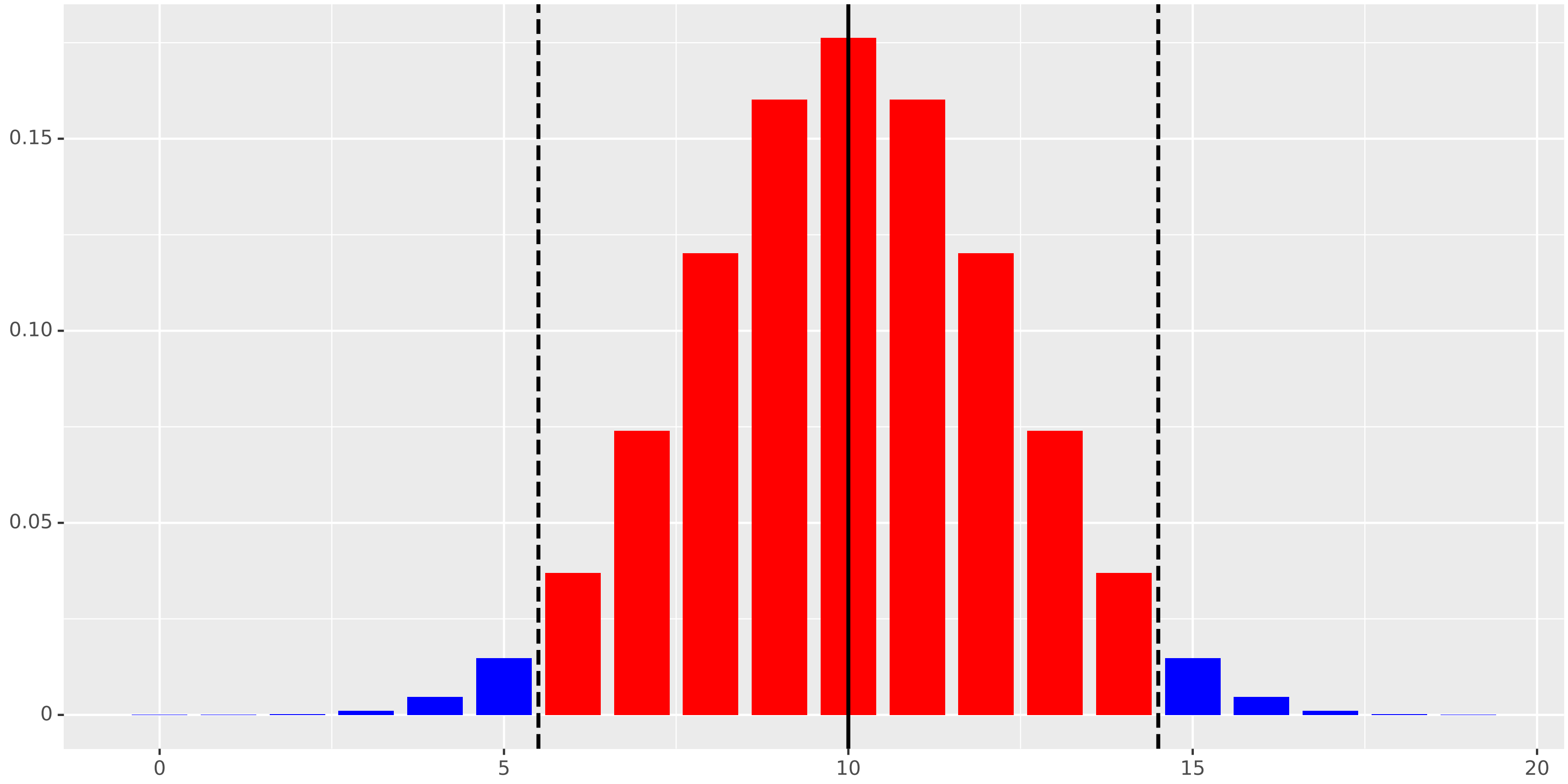
Then the probability is to get k ones in n independent trials is

$$\Pr\left[B_1 + \cdots + B_n = k | A\right] = \binom{n}{k} p^k (1-p)^{n-k}$$

The number of ones in known to have a *binomial distribution*

$$B_1 + \cdots + B_n \sim \mathsf{Bin}(n, p)$$

Illustration



The distribution of $B_1 + \ldots + B_n$ depends solely on the number of trials n and the probability p. Some values of $B_1 + \ldots + B_n$ are very unlikely.

How to build a statistical test

I. Null hypothesis:

▷ The probability of heads in a coinflip is $\Pr\left[B_i = 1\right] = p$.

II. Choose value to compute aka test statistic:

▷ Our test statistic will be $B_1 + \ldots + B_n$.

III. Consequences on the observations:

▷ The observed sum $B_1 + \ldots + B_n \sim \mathsf{Bin}(n = 20, p = 0.5)$.

▷ Limit on the tail probability $\Pr\left[|B_1 + \ldots + B_n - 10| \geq 6\right] \leq 5\%$

IV. Test procedure

▷ Reject null hypotesis at *significance level* 5% if $|B_1 + \ldots + B_n - 10| \geq 6$.

Properties of statistical tests

Statistical test is a classification algorithm designed to distinguish a fixed distribution of negative examples specified by a null hypothesis.

Any *static* classification algorithm can be converted to a statistical test by finding out the percentage of false positives aka *p-value*:

- ▷ There might exists a closed form solution.
- ▷ We can always estimate p-values using simulations.
- ▷ Observations must be compressed into a single decision value.

Testing several hypothesis in parallel increases the number of false positives. Several p-value adjustment methods are used to correct the issue:

- ▷ Bonferroni correction is almost optimal
- ▷ FDR correction controls the expected number false positives

How to build confidence intervals

I. Construct a family of statistical tests:

▷ Define a statistical test T_p for all possible parameter values p.

▷ All tests should share the same test statistic.

II. Perform multiple hypotesis testing for all parameter values:

▷ Accept all parameters values for which p-value is greater than $1 - \alpha$.

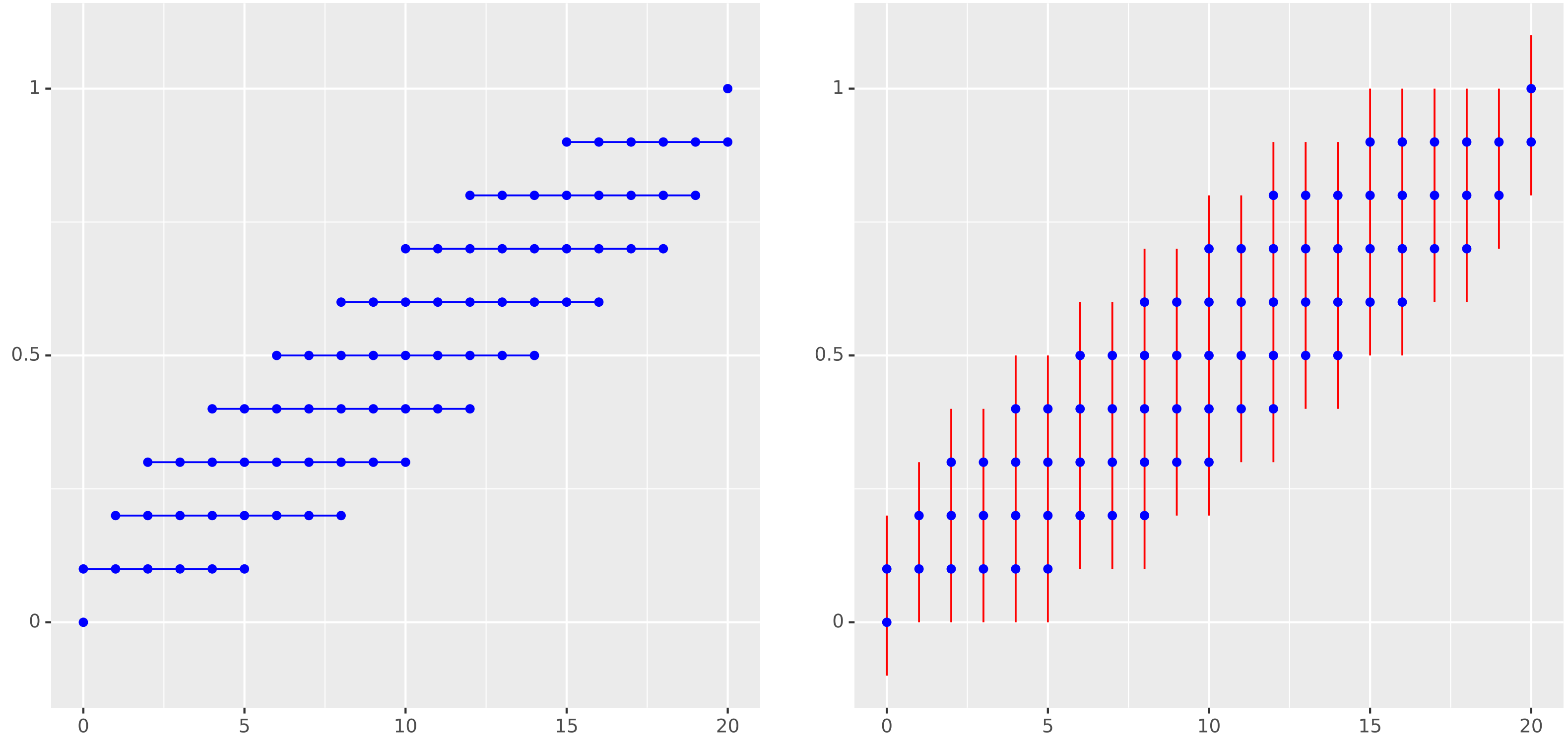
▷ Output a minimal interval that covers all accepted parameter values.

Rationale

▷ The true parameter value is accepted on α-fraction of possible observations.

▷ Otherwise, the true value is inside the predicted interval.

Illustration



- ▷ Acceptance ranges for different parameter values on the left.
- ▷ Extended parameter ranges covering all accepted parameters on the right.
- ▷ These ranges are the desired confidence intervals.

Interpretation of confidence intervals

Definition. Confidence interval for a parameter p is an outcome of an approximation algorithm. The algorithm must output an interval $[\hat{p}-\varepsilon, \hat{p}+\varepsilon]$ such that the true estimate is in the range on α-fraction of cases.

Paradoxical inapplicability

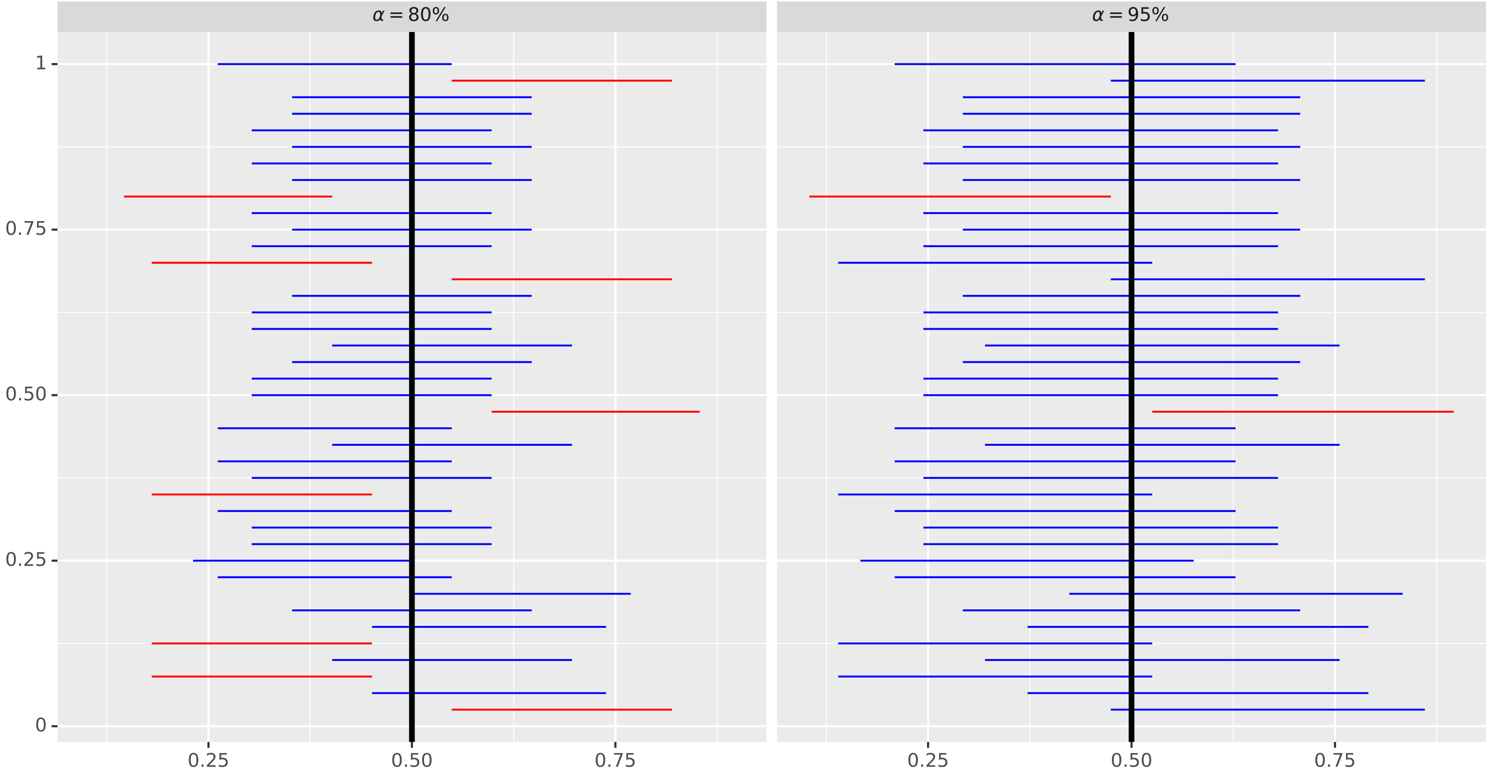
The definition does not state that the probability $p \in [\hat{p}-\varepsilon, \hat{p}+\varepsilon]$ is α!

- ▷ The statement $p \in [\hat{p} - \varepsilon, \hat{p} + \varepsilon]$ is either true or false.
- ▷ There is no probability left. We just *do not know* the answer!

Ultra-frequentistic resolution

- ▷ If $1 - \alpha$ is small enough say 5% then the algorithm is always correct.

Illustrative example



By increasing the length of the interval we increase the fraction of runs for which the true value of p lies in the interval.

Problems with confidence intervals

Inability to capture background knowledge

▷ What if I know that $p \in [0.1, 0.2]$ and observe $B_1 = \ldots = B_N = 1$?

▷ Then the estimate $[\hat{p} - \varepsilon, \hat{p} + \varepsilon]$ is clearly wrong although on average this confidence interval is reasonable.

Multiple hypothesis testing

▷ Using several confidence intervals in parallel increases the fraction of cases where some true estimate is out of the predicted range.

▷ We can use p-value adjustment methods are used to correct the issue.

Prediction intervals

Even if we know the true relation $y = f(\boldsymbol{x})$ we cannot predict the observation $y_i = f(\boldsymbol{x}_i) + \varepsilon_i$, as the noise term ε_i is not known ahead.

▷ We cannot give upper and lower bounds for y_i which always hold.

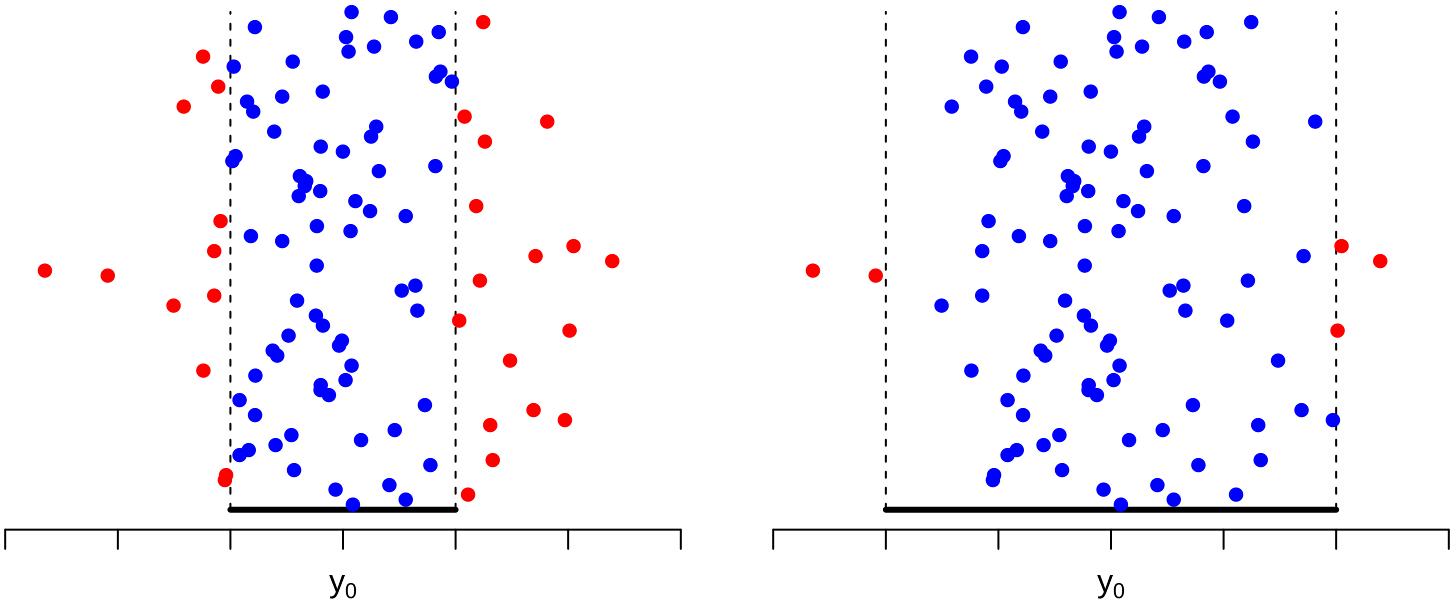
Instead, we can specify a prediction interval $[y_* - \varepsilon, y_* + \varepsilon]$ so that with probability 95% the resulting measurement y_i is in the range.

▷ Usually, the analysis is similar to confidence interval derivation.

Interpretation of prediction intervals is different from confidence intervals.

▷ The probability estimate holds for the particular interval.

Illustrative example



By increasing the length of the prediction interval we increase the fraction of future measurements which fall into interval.

Confidence envelopes

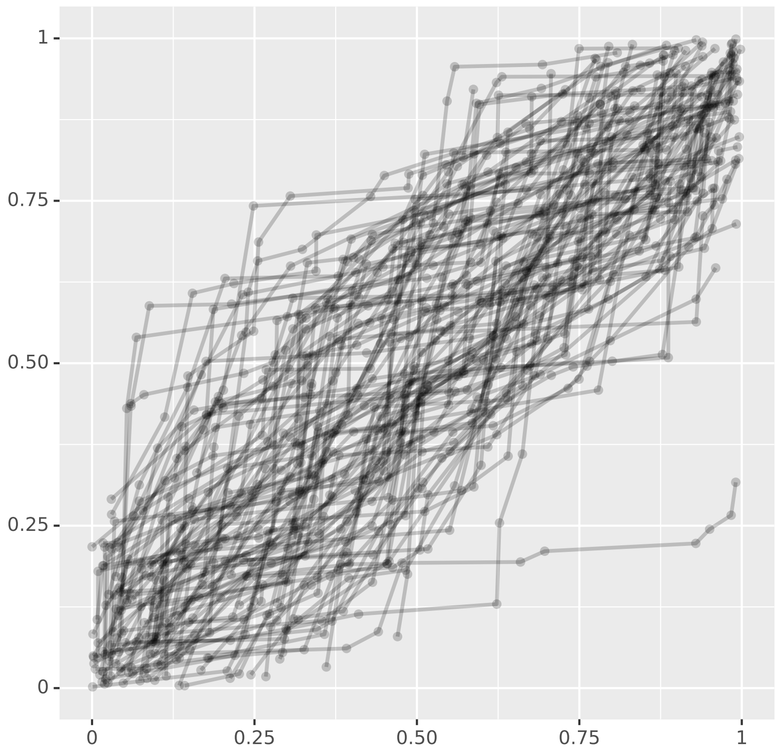
Confidence intervals is a good way to visualise uncertainty of a particular paramater. However, we are sometimes interested in the uncertainty many parameters or in the uncertainty of a function:

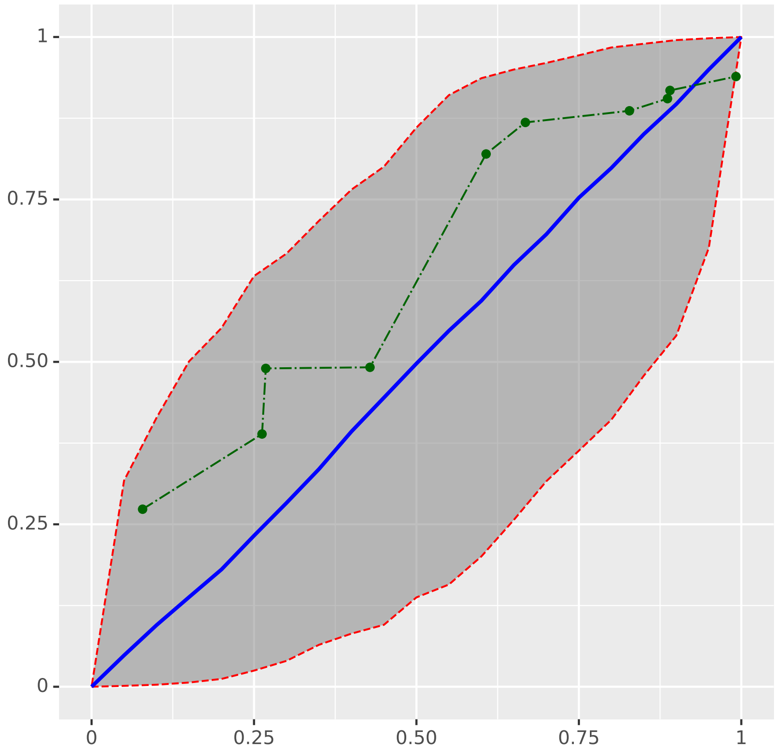
- ▷ How a predictor $f : [0,1] \to \mathbb{R}$ depends on the training set
- ▷ How a ROC curve $\textsc{Roc} : [0,1] \to [0,1]$ depends on the test set
- ▷ How should a quantile-quantile plot be distributed.

Confidence bands are generalisations of confidence intervals

- ▷ Pointwise confidence band is a collection of confidence intervals
- ▷ Simultaneous confidence band must enclose α-fraction of functions.
- ▷ Simultaneous confidence bands are much wider than pointwise bands.

Illustrative example





▷ Distribution of qq-lines visualised through a sample on the left.

▷ A simulation based pointwise 95% confidence envelope on the right.

▷ The significance level that qq-line is inside the envelope is ca 50%.

Permutation tests

Baseline problem:

- ▷ Achievable accuracy depends on the data distribution.
- ▷ Artefacts in the dataset may bias performance measures.

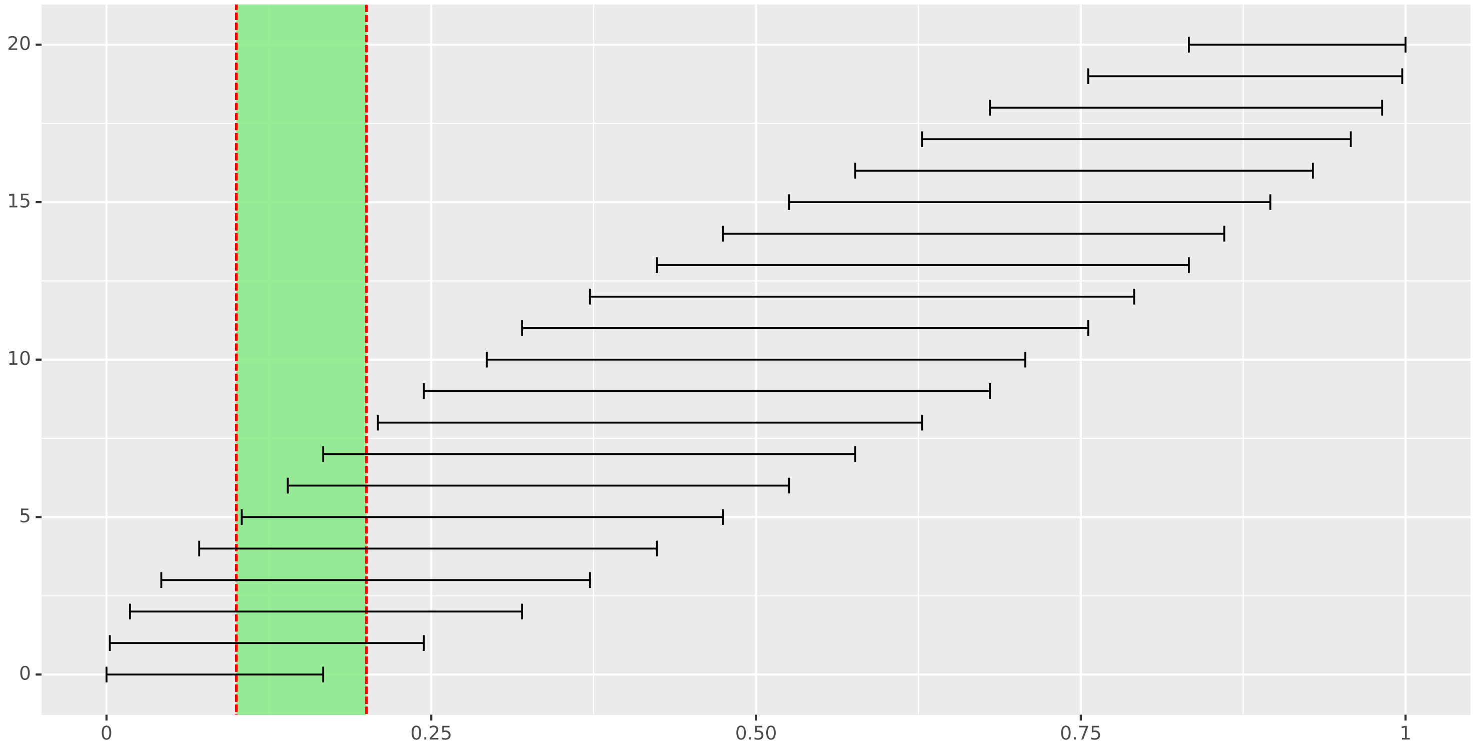
Label permutation. A random permutation π on outputs y_i destroys correlations between input-output pairs $(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})$ but preserves marginal distribution of inputs and outputs.

Permutation test. Estimate how probable is to achieve equal or higher accuracy than was observed on the real data.

- ▷ If this probability is small then there must be signal in the data.
- ▷ The test completely neglect the effect size, i.e., how much results differ.
- ▷ Statistical significance does not imply utility!

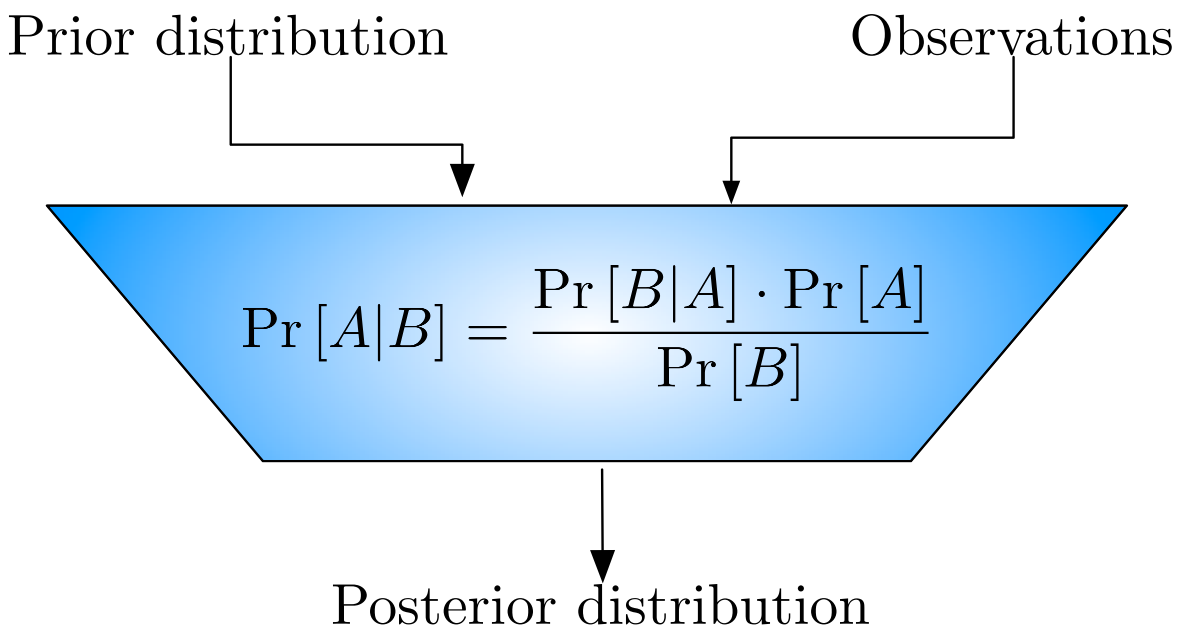
Bayesian methods

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



$$\Pr\left[A|B\right] = \frac{\Pr\left[B|A\right]\cdot\Pr\left[A\right]}{\Pr\left[B\right]}$$

- 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.

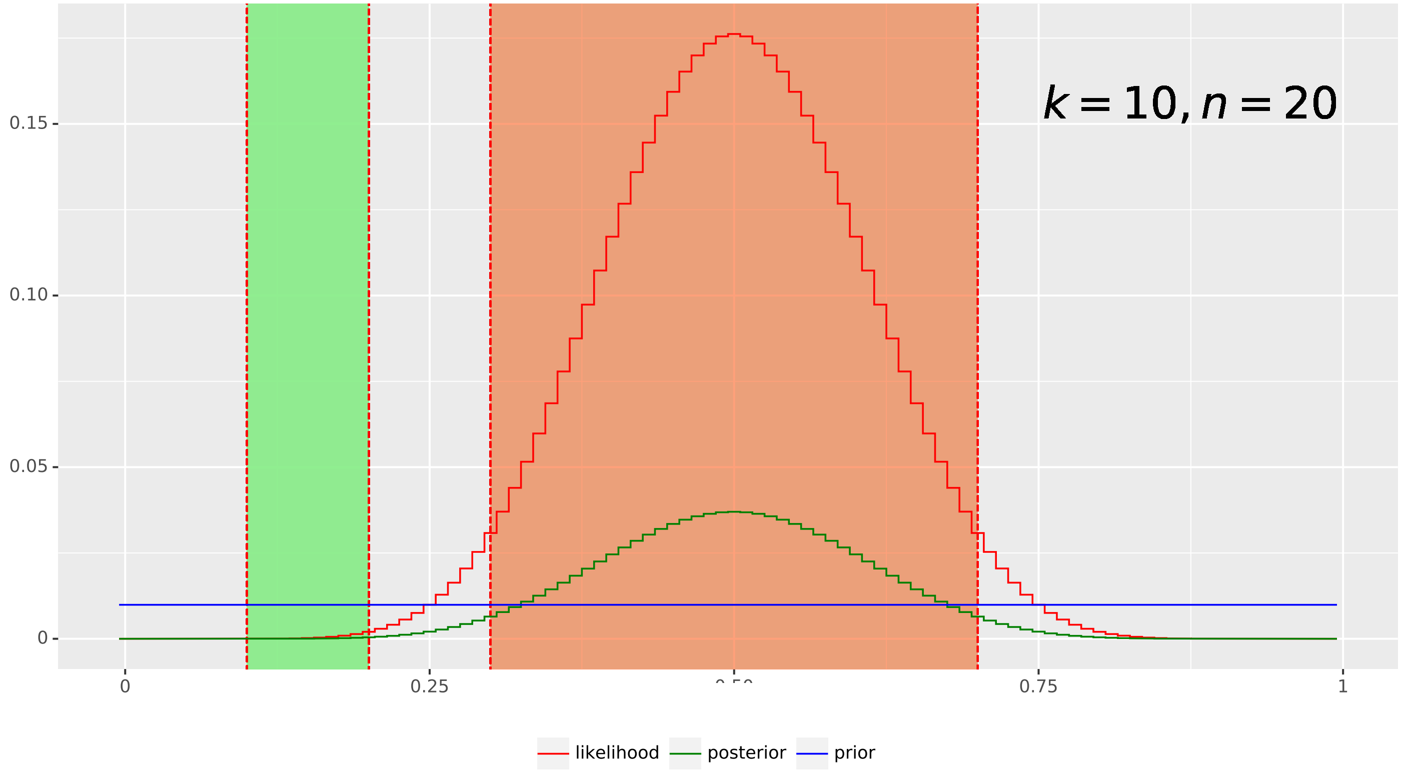
For coin flipping experiment the number of ones k is the observation and coin bias p is the model paramater and thus

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

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

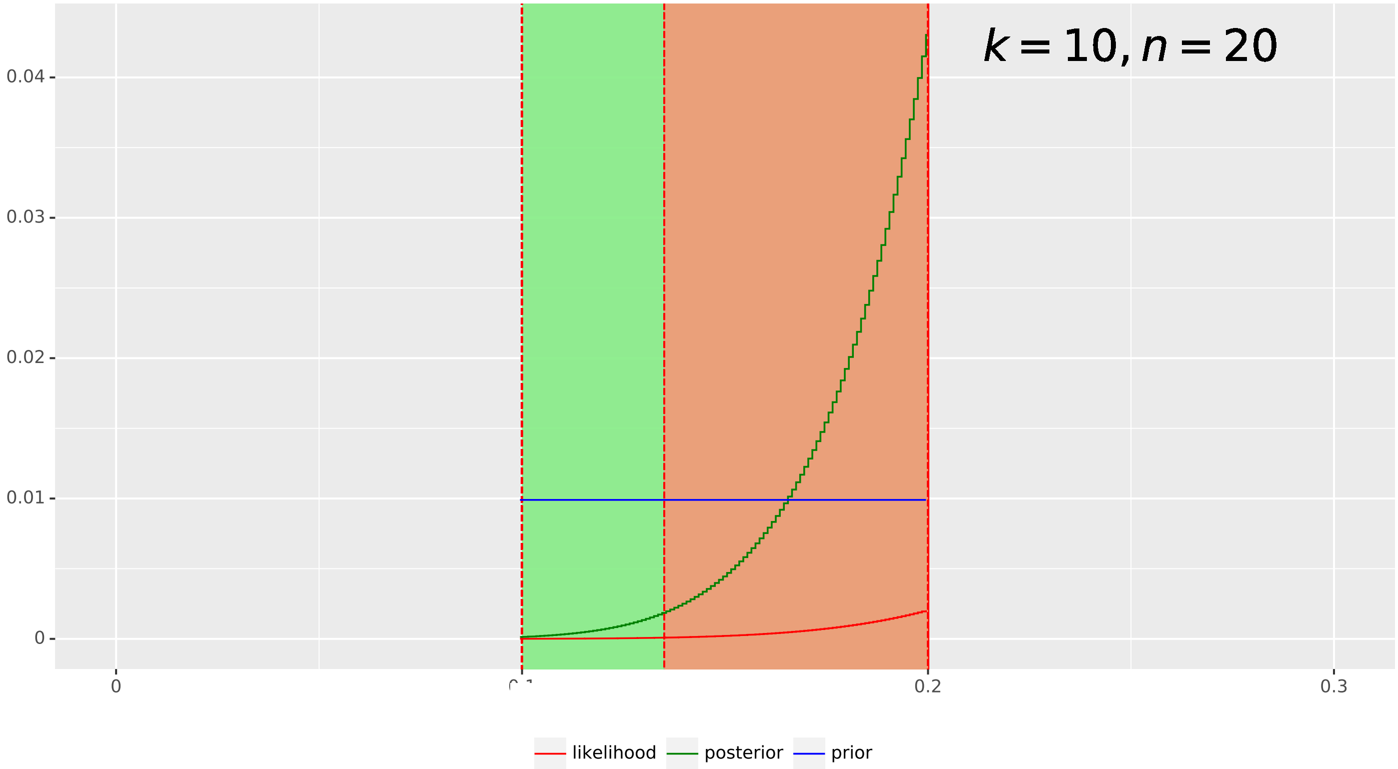
- ▷ Uninformative prior assigns uniform probability to all models.
- ▷ Uninformative prior is not well-defined for continuous parameters.

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

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

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

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

Dice throwing vs coinflipping

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

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

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.

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

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.

Markov chains

Definition. Markov chain with order m is an outcome of a process that outputs correlated observations $X_1, X_2, \ldots$ in such a way that the probability of the observation X_{m+i} depends only on the observations $X_{m+i-1}, \ldots, X_m$

Log-likelihood. Let $\boldsymbol{x} = (x_1, \ldots, x_{m+n})$ be a sequence of observations. Then the log-likelihood $\ell[\boldsymbol{x}]$ can be expressed

$$\ell[\boldsymbol{x}] = \log \underbrace{\Pr\left[x_1, \ldots, x_m\right]}_{\beta[x_1,\ldots,x_m]} + \sum_{i=1}^{n} \log \underbrace{\Pr\left[x_{m+i} | x_{m+i-1}, \ldots, x_i\right]}_{\alpha[x_i,\ldots,x_{i+m-1},x_{m+i}]}$$

where

- ▷ the tensor $\beta[\ldots]$ determines initial probabilities;
- ▷ the tensor $\alpha[\ldots]$ determines transition probabilities.

Reduction to the dice throwing experiment

Let $k(u_1,\ldots,u_{m+1})$ is the count of subsequences $u_1,\ldots,u_{m+1}$ then

$$\ell[\boldsymbol{x}] = \log\beta[x_1,\ldots,x_m] + \sum_{\boldsymbol{u}} k(\boldsymbol{u})\log\alpha[\boldsymbol{u}]$$

Let us assume that the posterior is defined

- ▷ fixing probabilities for slices $\alpha[u_1,\ldots,u_m,*]$
- ▷ multiplying these probabilities to get the probability of $\alpha[\ldots]$

The logarithm of the posterior decomposes into sum of independent terms

$$\sum_{u_{m+1}} k(u_1,\ldots,u_{m+1})\cdot\log\alpha[u_1,\ldots,u_{m+1}] + \log p(\alpha[u_1,\ldots,u_m,*])$$

This is equivalent to inferring probabilities in a dice throws

Hidden Markov models

Definition. Let $X_1, X_2, \ldots$ be hidden states that form a Markov chain and let $Y_1, Y_2, \ldots$ be observations that the probability of Y_i depends only on the state X_i. Then the entire process is known as Hidden Markov Model.

Log-likelihood. Let $\boldsymbol{y} = (y_1, \ldots, y_{m+n})$ and $\boldsymbol{x} = (x_1, \ldots, x_{m+n})$ be the observations and hidden states. Then the complete log likelihood is

$$\ell[\boldsymbol{x}, \boldsymbol{y}] = \log \beta[x_1, \ldots, x_m] + \sum_{i=1}^{n} \log \alpha[x_i, \ldots, x_{i+m-1}, x_{m+i}] + \log \delta[x_i, y_i]$$

where

- ▷ the tensor $\beta[\ldots]$ determines initial probabilities;
- ▷ the tensor $\alpha[\ldots]$ determines transition probabilities;
- ▷ the tensor $\delta[\ldots]$ determines emission probabilities.

Reduction to the previous building blocks

The problem simplifies when we know the vector of hidden states $\boldsymbol{x}$:

- ▷ Inference of emission probabilities $\delta[\ldots]$ reduce to dice throwing.
- ▷ Inference of the chain parameters $\alpha[\ldots]$ and $\beta[\ldots]$ is also possible.

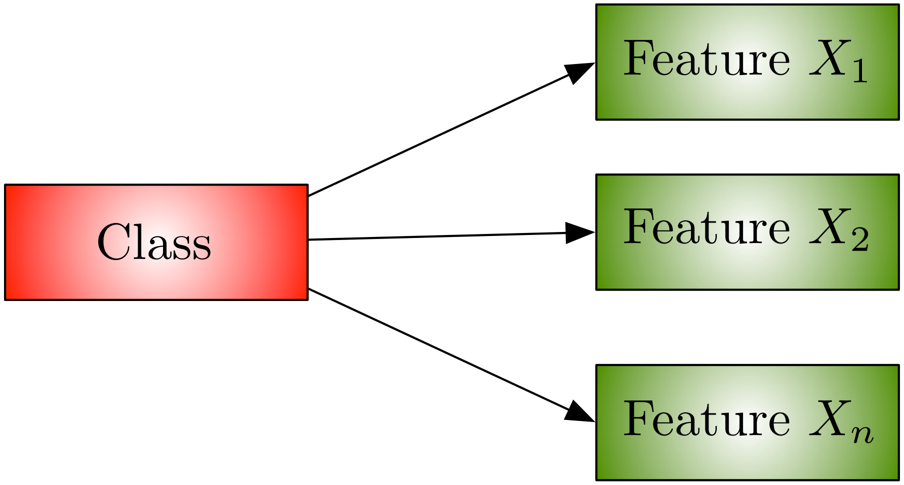
We can use Viterbi algorithm for finding the most probable hidden state $\boldsymbol{x}$ is easy if all parameters are known.

Naive inference algorithm: Fix random parameters and repeat steps:

- ▷ Given parameters α, β, δ learn the most probable hidden state $\boldsymbol{x}$.
- ▷ Given the most probable hidden state $\boldsymbol{x}$ learn model parameters α, β, δ.

This algorithm overfits as all hidden states can have similar probability.

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

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

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

which gives an *odd ratio*

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

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

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