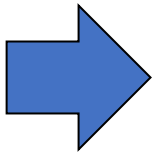
Introduction to Bayesian Statistics - 8

Edoardo Milotti

Università di Trieste and INFN-Sezione di Trieste

Our next important topic: Bayesian estimates often require complex numerical integrals. How do we confront this problem?



enter the Monte Carlo methods!

1. acceptance-rejection sampling
2. importance sampling
3. statistical bootstrap
4. Bayesian methods in a sampling-resampling perspective
5. Introduction to Markov chains and to Random Walks (RW)
6. Simulated annealing
7. The Metropolis algorithm
8. Markov Chain Monte Carlo (MCMC)
9. The Gibbs sampler
10. The efficiency of MCMC algorithms
11. Affine-invariant MCMC algorithms (EMCEE)

9. The Gibbs sampler

(adapted from Casella and George, *Explaining the Gibbs sampler* Am.Stat. 46 (1992) 167)

Let's start with an example, and consider the following joint distribution:

$$f(x,y) \propto \binom{n}{x} y^{x+\alpha-1}(1-y)^{n-x+\beta-1}, \quad x = 0,\ldots,n \quad 0 \le y \le 1$$

We see that

$$f(x|y) \sim \text{Binomial}(n,y)$$
$$f(y|x) \sim \text{Beta}(x+\alpha, n-x+\beta)$$

Next we set up a simple Markov chain procedure ...

We generate a "Gibbs sequence" of random variables

$$Y'_0, X'_0, Y'_1, X'_1, Y'_2, X'_2, \ldots, Y'_k, X'_k$$

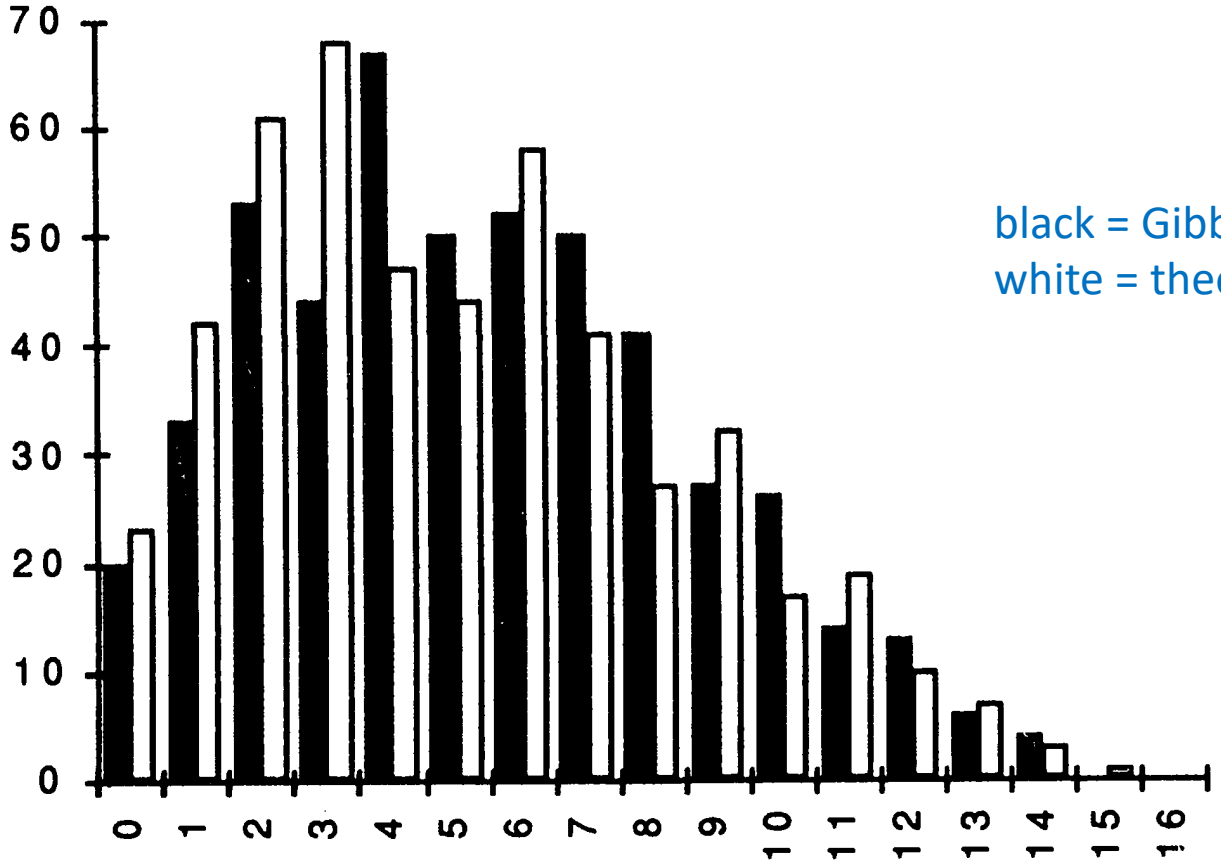
where the initial value is specified and the others are computed with the rule

$$X'_j \sim f(x \mid Y'_j = y'_j)$$

$$Y'_{j+1} \sim f(y \mid X'_j = x'_j)$$

(Gibbs sampling).

We observe that for large enough *k*, the final *X* values have a fixed distribution that corresponds to the marginal pdf of the *x* variate.



black = Gibbs sampling
white = theoretical expectation

Figure 1. Comparison of Two Histograms of Samples of Size $m = 500$ From the Beta-Binomial Distribution With $n = 16$, $\alpha = 2$, and $\beta = 4$. The black histogram sample was obtained using Gibbs sampling with $k = 10$. The white histogram sample was generated directly from the beta-binomial distribution.

Should we expect this result?

Consider the following expectation value

$$E_y[f(x|y)] = \int_Y f(x|y)f(y)dy = \int_Y f(x,y)dy = f(x)$$

therefore we can estimate *f*(*x*) with the sum

$$\hat{f}(x) = \frac{1}{m}\sum_{i=1}^{m} f(x \mid y_i)$$

where the y's are generated according to their marginal distribution; finally the Gibbs sampling provides representative samples that correspond to the marginal distribution of the x's. (for a mathematically accurate proof, check the paper by Casella&George)

Does Gibbs sampling converge?

We consider the following case: two discrete random variables with marginally Bernoulli distributions and with a joint probability distribution described by this matrix

Y \ X	0	1
0	p_1	p_2
1	p_3	p_4

$$p_i \geq 0, p_1 + p_2 + p_3 + p_4 = 1$$

$$\begin{bmatrix} f_{x,y}(0,0) & f_{x,y}(1,0) \\ f_{x,y}(0,1) & f_{x,y}(1,1) \end{bmatrix} = \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix}$$

$$\begin{bmatrix} f_{x,y}(0,0) & f_{x,y}(1,0) \\ f_{x,y}(0,1) & f_{x,y}(1,1) \end{bmatrix} = \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix}$$

$$f_x = [f_x(0) \quad f_x(1)] = [p_1 + p_3 \quad p_2 + p_4]$$

marginal distribution

from the usual formula for conditional probabilities

$$f_{y|x}(y|x) = \frac{f(x,y)}{f_x(x)}$$

$$A_{y|x} = \begin{bmatrix} \frac{p_1}{p_1 + p_3} & \frac{p_3}{p_1 + p_3} \\ \frac{p_2}{p_2 + p_4} & \frac{p_4}{p_2 + p_4} \end{bmatrix}$$

$$A_{x|y} = \begin{bmatrix} \frac{p_1}{p_1 + p_2} & \frac{p_2}{p_1 + p_2} \\ \frac{p_3}{p_3 + p_4} & \frac{p_4}{p_3 + p_4} \end{bmatrix}$$

transition probabilities

Since we are only interested in the X sequence

$$P(X'_1 = x_1 \mid X'_0 = x_0) = \sum_y P(X'_1 = x_1 \mid Y'_1 = y) \times P(Y'_1 = y \mid X'_0 = x_0).$$

the transition matrix for the X sequence is

$$A_{x|x} = A_{y|x} A_{x|y}$$

From the theory of Markov chains, we know that iterating this produces a fixed probability distribution, i.e., our marginal distribution for X.

So, what's the use of all this?

Consider the case where we want to compute the marginal pdf

$$f(x) = \int \ldots \int f(x, y_1, \ldots, y_p)\, dy_1 \ldots dy_p$$

in a situation where the multidimensional integral can be hard to compute.

The Gibbs sampler completely bypasses the calculation of the multidimensional integral and affords an easy path to marginalization.

Indeed, the procedure can be easily extended to multidimensional distributions, for example with two nuisance variables we produce the sequence

$$Y'_0, Z'_0, X'_0, Y'_1, Z'_1, X'_1, Y'_2, Z'_2, X'_2, \ldots$$

by means of the conditional PDFs

10. The efficiency of MCMC methods

The effective use of MCMC programs requires the fine-tuning of many aspects

- Duration of burn-in (initialization)
- Number of parallel chains (random walkers)
- Selection of jumping rule (proposal function)
- Selection of convergence rule
- …

Statistical Science
1992, Vol. 7, No. 4, 457–511

Inference from Iterative Simulation Using Multiple Sequences

Andrew Gelman and Donald B. Rubin

Abstract. The Gibbs sampler, the algorithm of Metropolis and similar iterative simulation methods are potentially very helpful for summarizing multivariate distributions. Used naively, however, iterative simulation can give misleading answers. Our methods are simple and generally applicable to the output of any iterative simulation; they are designed for researchers primarily interested in the science underlying the data and models they are analyzing, rather than for researchers interested in the probability theory underlying the iterative simulations themselves. Our recommended strategy is to use several independent sequences, with starting points sampled from an overdispersed distribution. At each step of the iterative simulation, we obtain, for each univariate estimand of interest, a distributional estimate and an estimate of how much sharper the distributional estimate might become if the simulations were continued indefinitely. Because our focus is on applied inference for Bayesian posterior distributions in real problems, which often tend toward normality after transformations and marginalization, we derive our results as normal-theory approximations to exact Bayesian inference, conditional on the observed simulations. The methods are illustrated on a random-effects mixture model applied to experimental measurements of reaction times of normal and schizophrenic patients.

Key words and phrases: Bayesian inference, convergence of stochastic processes, EM, ECM, Gibbs sampler, importance sampling, Metropolis algorithm, multiple imputation, random-effects model, SIR.

1.3 Our Approach

Our method is composed of two major steps. First, an estimate of the target distribution is created, centered about its mode (or modes, which are typically found by an optimization algorithm) and "overdispersed" in the sense of being more variable than the target distribution. The approximate distribution is then used to start several independent sequences of the iterative simulation. The second major step is to analyze the multiple sequences to form a distributional estimate of what is known about the target random variable, given the simulations thus far. This distributional estimate, which is in the form of a Student's t distribution for each scalar estimand, is somewhere between its starting and target distributions and provides the basis for an estimate of how close the simulation process is to convergence – that is, how much sharper the distributional estimate might become if the simulations were run longer.

The G-R monitoring index

- *m* chains (walkers) and *n* samples/chain
- summary variable ψ with mean μ and st. dev. σ under the target distribution
- ψ_{ji} is the value of the summary variable at the *i*-th iteration within the *j*-th chain
- we evaluate

$$W = \frac{1}{m(n-1)} \sum_{j=1}^{m} \sum_{i=1}^{n} (\psi_{ji} - \bar{\psi} j.)^2$$

within-sequence variance

$$B/n = \frac{1}{m-1} \sum_{j=1}^{m} (\psi_{ji} - \bar{\psi}..)^2$$

between-sequence variance

- σ can be estimated from these variances

$$\hat{\sigma}^2 = \frac{n-1}{n} W + \frac{B}{n}$$

The G-R monitoring index (ctd)

- the variance

$$\hat{\sigma}^2 = \frac{n-1}{n}W + \frac{B}{n}$$

- would be an unbiased estimate of the true variance if the starting points were drawn from the target distribution, but is an overestimate if the starting distribution is overdispersed
- taking the variability of the mean into account yields a different estimate of the variance

$$\hat{V} = \hat{\sigma}^2 + \frac{B}{mn}$$

- the so-called potential scale reduction factor (PSRF) can be interpreted as a convergence diagnostic (when large it can be further decreased by continuing the simulation, when close to 1 it shows that the set of simulations is close to the target distribution)

$$\hat{R} = \frac{\hat{V}}{W} = \frac{m+1}{m}\frac{\hat{\sigma}^2}{W} - \frac{n-1}{mn}$$

An Introduction to Probability Theory and Its Applications

WILLIAM FELLER

Eugene Higgins Professor of Mathematics

Princeton University

VOLUME I

THIRD EDITION

CHAPTER III*

Fluctuations in Coin Tossing and Random Walks

This chapter digresses from our main topic, which is taken up again only in chapter V. Its material has traditionally served as a first orientation and guide to more advanced theories. Simple methods will soon lead us to results of far-reaching theoretical and practical importance. We shall encounter theoretical conclusions which not only are unexpected but actually come as a shock to intuition and common sense. They will reveal that commonly accepted notions concerning chance fluctuations are without foundation and that the implications of the law of large numbers are widely misconstrued. For example, in various applications it is assumed that observations on an individual coin-tossing game during a long time interval will yield the same statistical characteristics as the observation of the results of a huge number of independent games at one given instant. This is not so. Indeed, using a currently popular jargon we reach the conclusion that in a population of normal coins the majority is necessarily maladjusted. [For empirical illustrations see section 6 and example (4.*b*).]

6. AN EXPERIMENTAL ILLUSTRATION

Figure 4 represents the result of a computer experiment simulating 10,000 tosses of a coin; the same material is tabulated in example I, (6.*c*). The top line contains the graph of the first 550 trials; the next two lines represent the entire record of 10,000 trials the scale in the horizontal direction being changed in the ratio 1:10. The scale in the vertical direction is the same in the two graphs.

When looking at the graph most people feel surprised by the length of the intervals between successive crossings of the axis. As a matter of fact, the graph represents a rather mild case history and was chosen as the mildest among three available records. A more startling example is obtained by looking at the same graph in the *reverse* direction; that is, reversing the order in which the 10,000 trials actually occurred (see section 8). Theoretically, the series as graphed and the reversed series are equally legitimate as representative of an ideal random walk. The reversed random walk has the following characteristics. Starting from the origin

the path stays on the	
negative side	*positive side*
for the first 7804 *steps*	*next* 8 *steps*
next 2 *steps*	*next* 54 *steps*
next 30 *steps*	*next* 2 *steps*
next 48 *steps*	*next* 6 *steps*
next 2046 *steps*	
Total of 9930 steps	*Total of* 70 steps
Fraction of time: 0.993	*Fraction of time:* 0.007

This *looks* absurd, and yet the probability that in 10,000 tosses of a perfect coin the lead is at one side for more than 9930 trials and at the other for fewer than 70 exceeds $\frac{1}{10}$. In other words, on the average *one record out of ten will look worse than the one just described.* By contrast, the probability of a balance better than in the graph is only 0.072.

The original record of figure 4 contains 78 changes of sign and 64 other returns to the origin. The reversed series shows 8 changes of sign and 6 other returns to the origin. Sampling of expert opinion revealed that even trained statisticians expect much more than 78 changes of sign in 10,000 trials, and nobody counted on the possibility of only 8 changes of sign. Actually the probability of not more than 8 changes of sign exceeds 0.14, whereas the probability of more than 78 changes of sign is about 0.12. As far as the number of changes of sign is concerned the two records stand on a par and, theoretically, neither should cause surprise. If they seem startling, this is due to our faulty intuition and to our having been exposed to too many vague references to a mysterious "law of averages."

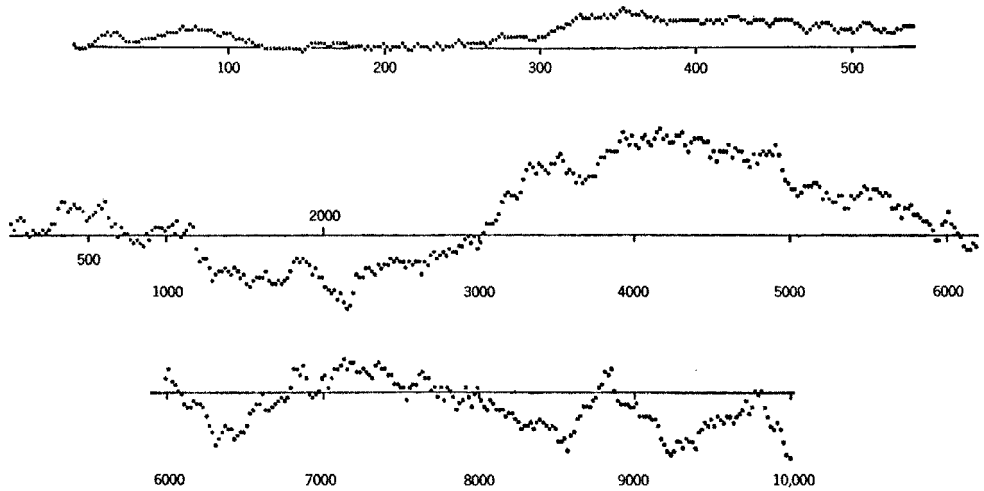
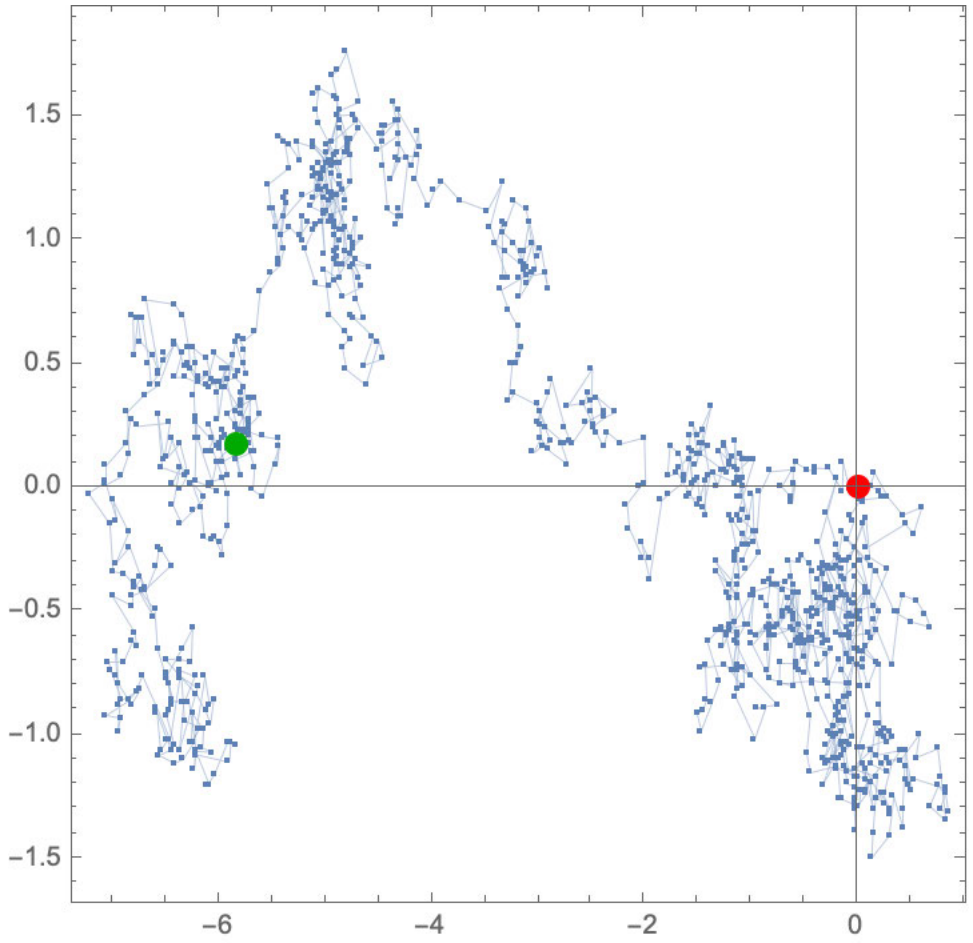
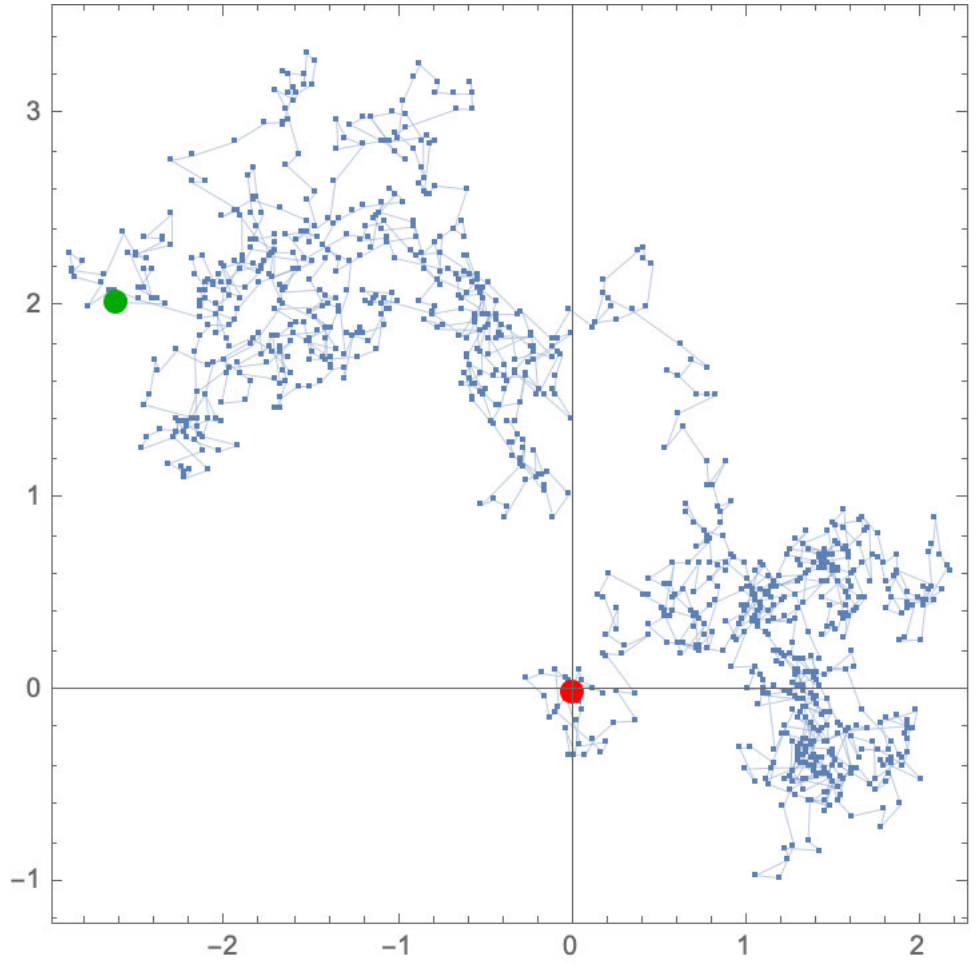
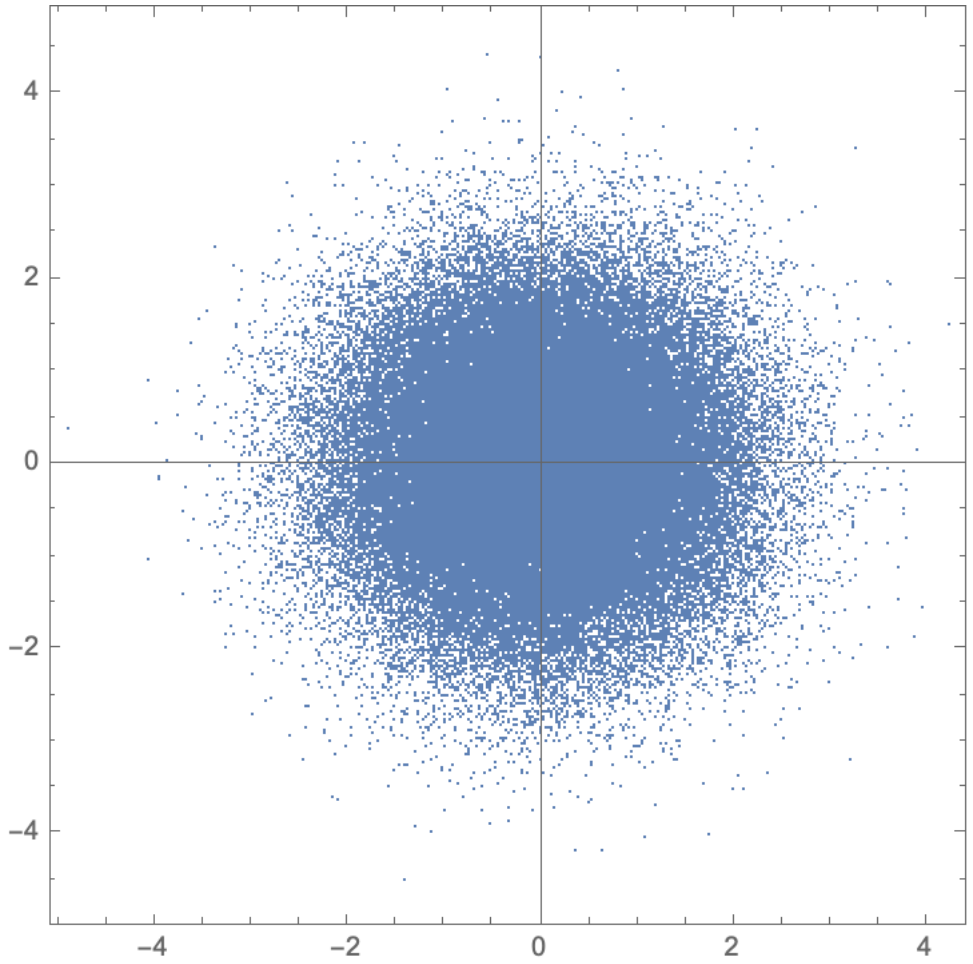


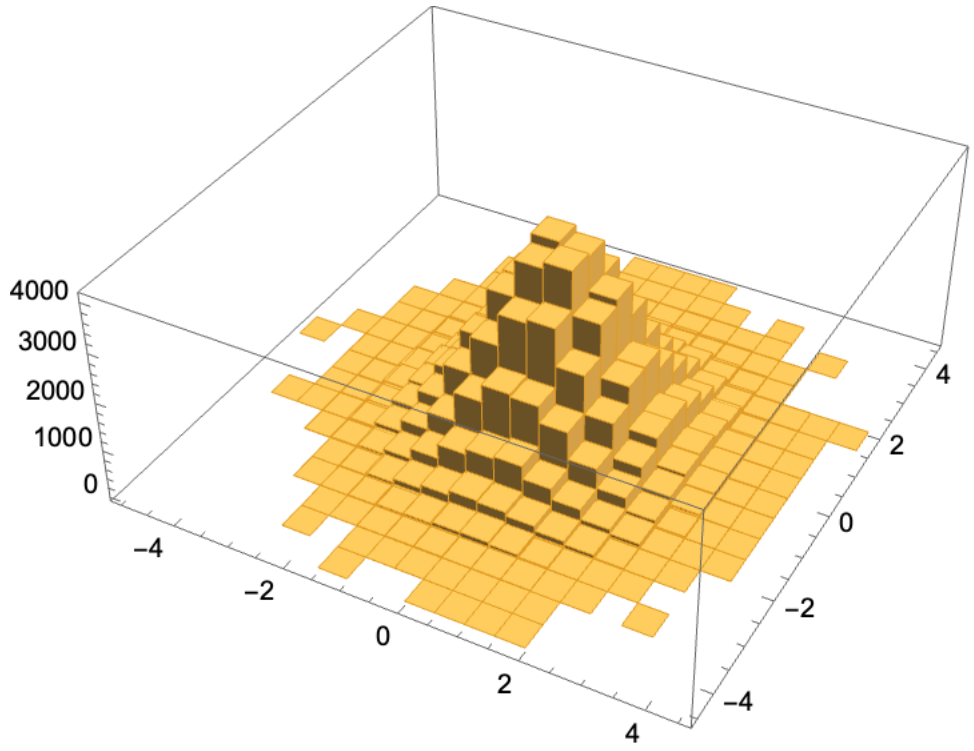
Figure 4. The record of 10,000 tosses of an ideal coin (described in section 6).

- start
- end






4
2
0
−2
−4
−4
−2
0
2
4


4000
3000
2000
1000
0
−4
−2
0
2
4
4
2
0
−2
−4

11. Affine-invariant MCMC algorithms

PUBLICATIONS OF THE ASTRONOMICAL SOCIETY OF THE PACIFIC, **125**:306–312, 2013 March
© 2013. The Astronomical Society of the Pacific. All rights reserved. Printed in U.S.A.

emcee: The MCMC Hammer

DANIEL FOREMAN-MACKEY,[1] DAVID W. HOGG,[1,2] DUSTIN LANG,[3,4] AND JONATHAN GOODMAN[5]

Received 2013 January 09; accepted 2013 January 30; published 2013 February 25

ABSTRACT. We introduce a stable, well tested Python implementation of the affine-invariant ensemble sampler for Markov chain Monte Carlo (MCMC) proposed by Goodman & Weare (2010). The code is open source and has already been used in several published projects in the astrophysics literature. The algorithm behind `emcee` has several advantages over traditional MCMC sampling methods and it has excellent performance as measured by the autocorrelation time (or function calls per independent sample). One major advantage of the algorithm is that it requires hand-tuning of only 1 or 2 parameters compared to $\sim N^2$ for a traditional algorithm in an N-dimensional parameter space. In this document, we describe the algorithm and the details of our implementation. Exploiting the parallelism of the ensemble method, `emcee` permits *any* user to take advantage of multiple CPU cores without extra effort. The code is available online at http://dan.iel.fm/emcee under the GNU General Public License v2.

- Optimizing a MCMC in a given parameter space often means that we use a proposal distribution that is tuned to the target distribution.

- This proposal distribution is often a multivariate Gaussian with an *n* x *n* covariance matrix that must be tuned accordingly

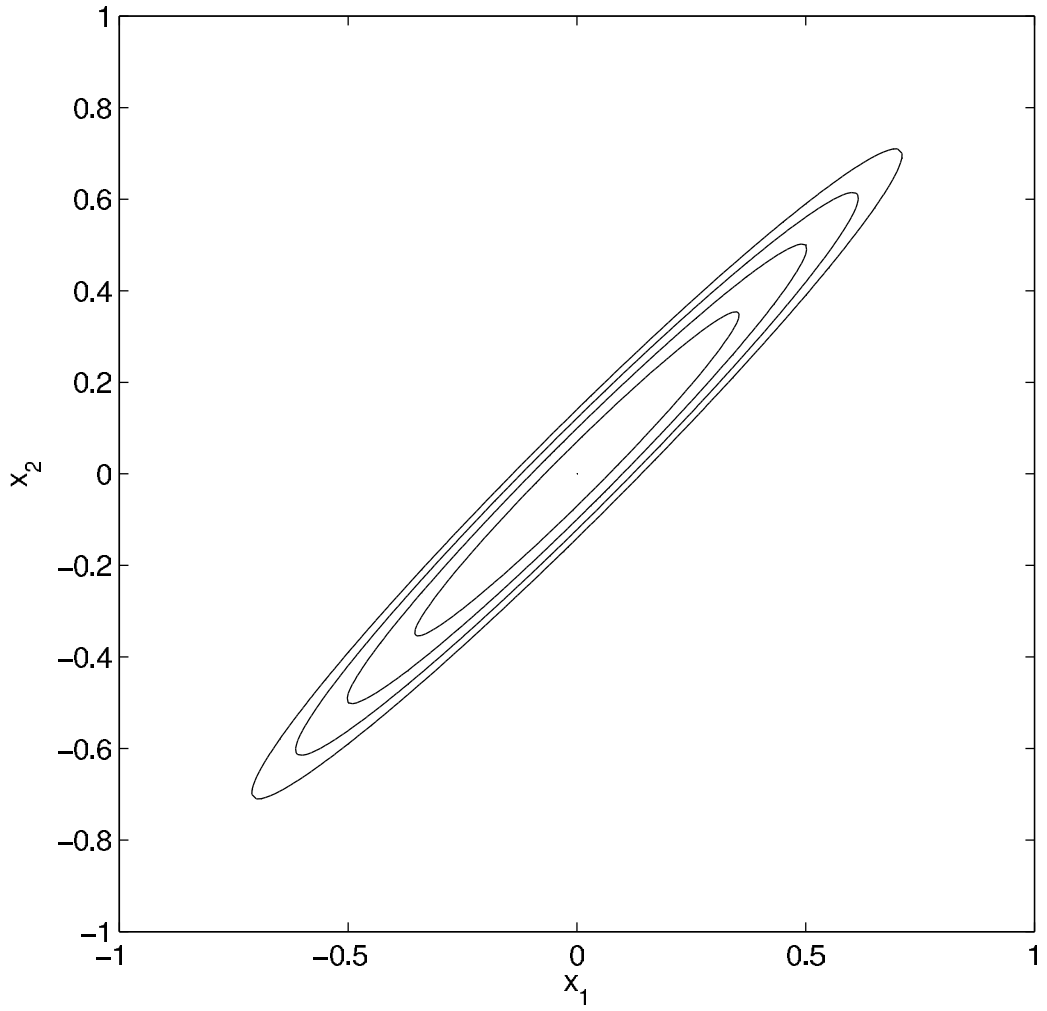
$$p(x_1, \ldots, x_n) = \frac{1}{(2\pi)^{n/2} |\det V|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}^T V^{-1} \mathbf{x}\right)$$

with

$$V = \begin{pmatrix} \sigma_1^2 & \rho_{1,2}\sigma_1\sigma_2 & \cdots & \rho_{1,n}\sigma_1\sigma_n \\ \rho_{1,2}\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2,n}\sigma_2\sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1,n}\sigma_1\sigma_n & \rho_{2,n}\sigma_2\sigma_n & \cdots & \sigma_n^2 \end{pmatrix}$$

- The covariance matrix has $\frac{1}{2}n(n+1)$ indipendent elements;

  tuning the proposal distribution means tuning these independent elements (hyperparameters) with a long (and computationally expensive) burn-in phase.



Consider the following highly anisotropic pdf

$$p(\mathbf{x}) \propto \exp\left(-\frac{(x_1 - x_2)^2}{2\sqrt{\epsilon}} - \frac{(x_1 + x_2)^2}{2}\right)$$

With the variable transformation

$$y_1 = \frac{x_1 - x_2}{\sqrt{\epsilon}}, \quad y_2 = x_1 + x_2$$

which has the Jacobian

$$J(\mathbf{y}, \mathbf{x}) = \begin{vmatrix} \frac{1}{\sqrt{\epsilon}} & -\frac{1}{\sqrt{\epsilon}} \\ 1 & 1 \end{vmatrix} = \frac{2}{\sqrt{\epsilon}}$$

Then, we find that this affine transformation transforms the original Gaussian into the simpler Gaussian

$$p(\mathbf{y}) \propto \exp\left(-\frac{y_1^2}{2} - \frac{y_2^2}{2}\right)$$

In the n-dimensional parameter space there are only 2 hyperparameters to tune (mean, variance) instead of $\frac{1}{2}n(n+1)$

The affine transformation is implemented as follows, with an ensemble of K walkers $\{X_k\}$

$$X_k(t) \rightarrow Y = X_j + Z[X_k(t) - X_j]$$

where the index *j* is drawn at random from the set of all the all the indexes excluding *k*, and Z is a random variable from a distribution *g* such that

$$g(z) \propto \begin{cases} \frac{1}{\sqrt{Z}} & \text{if } z \in \left[\frac{1}{a}, a\right] \\ 0 & \text{otherwise} \end{cases}$$

and acceptance probability

$$q = \min\left(1, Z^{N-1} \frac{p(Y)}{p(X_k(t))}\right)$$

which satisfies detailed balance.

https://emcee.readthedocs.io/en/stable/

emcee

emcee is an MIT licensed pure-Python implementation of Goodman & Weare's Affine Invariant Markov chain Monte Carlo (MCMC) Ensemble sampler and these pages will show you how to use it.

This documentation won't teach you too much about MCMC but there are a lot of resources available for that (try this one). We also published a paper explaining the emcee algorithm and implementation in detail.

emcee has been used in quite a few projects in the astrophysical literature and it is being actively developed on GitHub.

Basic Usage

If you wanted to draw samples from a 5 dimensional Gaussian, you would do something like:

```
import numpy as np
import emcee

def log_prob(x, ivar):
    return -0.5 * np.sum(ivar * x ** 2)

ndim, nwalkers = 5, 100
ivar = 1. / np.random.rand(ndim)
p0 = np.random.randn(nwalkers, ndim)

sampler = emcee.EnsembleSampler(nwalkers, ndim, log_prob, args=[ivar])
sampler.run_mcmc(p0, 10000)
```

A more complete example is available in the Quickstart tutorial.

Model selection

The generic purpose of a model selection statistic is to set up a tension between the predictiveness of a model (for instance indicated by the number of free parameters) and its ability to fit observational data. Oversimplistic models offering a poor fit should of course be thrown out, but so should more complex models that offer poor predictive power.

There are two main types of model selection statistic that have been used in the literature so far. Information criteria look at the best-fitting parameter values and attach a penalty for the number of parameters; they are essentially a technical formulation of "chi-squared per degrees of freedom" arguments. By contrast, the Bayesian evidence applies the same type of likelihood analysis familiar from parameter estimation, but at the level of models rather than parameters. It depends on goodness of fit across the entire model parameter space.

(Liddle & al., 2006)

Akaike Information Criterion (AIC).

This was derived by Hirotugu Akaike in 1974, and takes the form

$$\mathrm{AIC} = -2\ln \mathcal{L}_{\max} + 2k$$

where k is the number of parameters in the model. The subscript "max" indicates that one should find the parameter values yielding the highest possible likelihood within the model. ***This second term acts as a kind of "Occam factor"; initially, as parameters are added, the fit to data improves rapidly until a reasonable fit is achieved, but further parameters then add little and the penalty term 2k takes over.*** *The generic shape of the AIC as a function of number of parameters is a rapid fall, a minimum, and then a rise. The preferred model sits at the minimum.*

The AIC was derived from information-theoretic considerations, specifically an approximate minimization of the Kullback–Leibler information entropy which measures the distance between two probability distributions.

(Liddle & al., 2006)

Bayesian Information Criterion (BIC).

This was derived by Gideon Schwarz in 1978, and strongly resembles the AIC. It is given by

$$\mathrm{BIC} = -2 \ln \mathcal{L}_{\max} + k \ln N$$

where N is the number of datapoints. Since a typical dataset will have lnN > 2, the BIC imposes a stricter penalty against extra parameters than the AIC.

It was derived as an approximation to the Bayesian evidence, … but the assumptions required are very restrictive and unlikely to hold in practice, rendering the approximation quite crude.

(Liddle & al., 2006)

Bayesian evidence

Model selection aims to determine which theoretical models are most plausible given some data, without necessarily considering preferred values of model parameters.

Ideally, we would like to estimate posterior probabilities on the set of all competing models using Bayes' theorem:

$$P(M_i|D,I) = \frac{P(D|M_i,I)P(M_i|I)}{\sum_k P(D|M_k,I)P(M_k|I)}$$

and select the best model using the odds ratio

$$\mathcal{O}_{i,j} = \frac{P(M_i|D,I)}{P(M_i|D,I)} = \frac{P(D|M_i,I)P(M_i|I)}{P(D|M_j,I)P(M_j|I)}$$

or the Bayes factor, if we assume equal prior probabilities for the different models:

$$B_{i,j} = \frac{P(D|M_i,I)}{P(D|M_j,I)}$$

Thus, we see that the Bayes factor is a ratio of evidences

$$B_{i,j} = \frac{P(D|M_i, I)}{P(D|M_j, I)}$$

As usual, each evidence is obtained by marginalizing the likelihood with respect to the (potentially different) parameters:

$$P(D|M_i, I) = \int_{\Theta_i} P(D|\boldsymbol{\theta}_i, M_i, I) p(\boldsymbol{\theta}_i|M_i, I) d\boldsymbol{\theta}_i$$

The evidence of a model is thus the average likelihood of the model in the prior.

Unlike the AIC and BIC, it does not focus on the best-fitting parameters of the model, but asks "of all the parameter values you thought were viable before the data came along, how well on average did they fit the data?". Literally, it is the likelihood of the model given the data.

The evidence rewards predictability of models, provided they give a good fit to the data, and hence gives an axiomatic realization of Occam's razor.

A model with little parameter freedom is likely to fit data over much of its parameter space, whereas a model that could match pretty much any data that might have cropped up will give a better fit to the actual data but only in a small region of its larger parameter space, pulling the average likelihood down.

(Liddle & al., 2006)

Which statistics?

Of these statistics, we would advocate using – wherever possible – the Bayesian evidence, which is a full implementation of Bayesian inference and can be directly interpreted in terms of model probabilities. It is computationally challenging to compute, being a highly peaked multidimensional integral, but recent algorithm development has made it feasible in cosmological contexts.

If the Bayesian evidence cannot be computed, the BIC can be deployed as a substitute. *It is much simpler to compute as one need only find the point of maximum likelihood for each model.* ***However, interpreting it can be difficult. Its main usefulness is as an approximation to the evidence, but this holds only for Gaussian likelihoods and provided the datapoints are independent and identically distributed.*** *The latter condition holds poorly for the current global cosmological dataset, though it can potentially be improved by binning of the data, hence decreasing the N in the penalty term.*

The AIC has been widely used outside astrophysics but is of debatable utility. ***It has been shown to be "dimensionally inconsistent", meaning that it is not guaranteed to give the right result even in the limit of infinite unbiased data.*** *It may be useful for checking the robustness of conclusions drawn using the BIC.* ***The evidence and BIC are dimensionally consistent.***

(Liddle & al., 2006)

Bayesian classification

data X, classes C

this likelihood is defined by training data

$$P(C|X) = \frac{P(X|C)}{P(X)} P(C)$$

the prior is also defined by training data

we can use the prior learning to assign a class to new data

$$C_k = \arg\max_{C_k} \frac{P(X|C_k)}{P(X)} P(C_k) = \arg\max_{C_k} P(X|C_k) P(C_k)$$

Consider a vector of *N* attributes given as Boolean variables $\mathbf{x} = \{x_i\}$ and classify the data vectors with a single Boolean variable.

The learning procedure must yield:

$P(y)$ it is easy to obtain it as an empirical distribution from an histogram of training class data: y is Boolean, the histogram has just two bins, and a hundred examples suffice to determine the empirical distribution to better than 10%.

$P(\mathbf{x}|y)$ there is a bigger problem here: the arguments have 2^{N+1} different values, and we must estimate $2(2^N-1)$ parameters ... for instance, with N = 30 there are more than 2 billion parameters!

How can we reduce the huge complexity of learning?

we assume the conditional independence of the x_n's:
naive Bayesian learning

for instance, with just two attributes

$$P(x_1, x_2 | y) = P(x_1 | x_2, y) P(x_2 | y) = P(x_1 | y) P(x_2 | y)$$

conditional independence assumption

with more than 2 attributes

$$P(\mathbf{x} | y) \approx \prod_{k=1}^{N} P(x_k | y)$$

Therefore:

$$P(y_k|\mathbf{x}) = \frac{P(\mathbf{x}|y_k)}{P(\mathbf{x})}P(y_k) = \frac{P(\mathbf{x}|y_k)}{\sum_j P(\mathbf{x}|y_j)P(y_j)}P(y_k)$$

$$\approx \frac{\prod_{n=1}^{N} P(x_n|y_k)}{\sum_j P(y_j)\prod_{n=1}^{N} P(x_n|y_j)}P(y_k)$$

and we assign the class according to the rule (MAP)

$$y = \arg\max_{y_k} \frac{\prod_{n=1}^{N} P(x_n|y_k)}{\sum_j P(y_j)\prod_{n=1}^{N} P(x_n|y_j)}P(y_k)$$

More general discrete inputs

If any of the N x variables has J different values, e if there are K classes, then we must estimate in all $NK(J\text{-}1)$ free parameters with the Naive Bayes Classifier (this includes normalization) (compare this with the $K(J^N\text{-}1)$ parameters needed by a complete classifier)

Continuous inputs and discrete classes – the Gaussian case

$$P\left(x_n \middle| y_k\right) = \frac{1}{\sqrt{2\pi\sigma_{nk}^2}} \exp\left[-\frac{\left(x_n - \mu_{nk}\right)^2}{2\sigma_{nk}^2} \right]$$

here we must estimate 2*NK* parameters + the shape of the distribution *P*(*y*) (this adds up to another *K*-1 parameters)

Gaussian special case with class-independent variance and Boolean classification (two classes only):

$$P\left(y=0\middle|\mathbf{x}\right)=\frac{P\left(\mathbf{x}\middle|y=0\right)P\left(y=0\right)}{P\left(\mathbf{x}\middle|y=0\right)P\left(y=0\right)+P\left(\mathbf{x}\middle|y=1\right)P\left(y=1\right)}$$

$$P\left(x_n\middle|y=0\right)=\frac{1}{\sqrt{2\pi\sigma_n^2}}\exp\left[-\frac{\left(x_n-\mu_{n0}\right)^2}{2\sigma_n^2}\right]$$

$$P\left(x_n\middle|y=1\right)=\frac{1}{\sqrt{2\pi\sigma_n^2}}\exp\left[-\frac{\left(x_n-\mu_{n1}\right)^2}{2\sigma_n^2}\right]$$

$$P(y=0|\mathbf{x}) = \frac{P(\mathbf{x}|y=0)P(y=0)}{P(\mathbf{x}|y=0)P(y=0)+P(\mathbf{x}|y=1)P(y=1)}$$

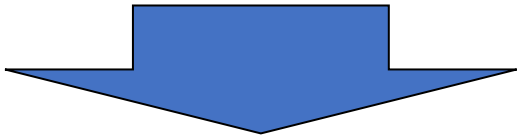
$$= \frac{1}{1+\frac{P(\mathbf{x}|y=1)P(y=1)}{P(\mathbf{x}|y=0)P(y=0)}}$$

$$= \frac{1}{1+\frac{P(y=1)}{P(y=0)}\prod_{n=1}^{N}\exp\left[-\frac{(x_n-\mu_{n1})^2}{2\sigma_n^2}+\frac{(x_n-\mu_{n0})^2}{2\sigma_n^2}\right]}$$

$$= \frac{1}{1+\exp\left\{\ln\left(\frac{P(y=1)}{P(y=0)}\right)+\sum_{n=1}^{N}\left[\frac{(\mu_{n1}-\mu_{n0})x_n}{\sigma_n^2}+\frac{\mu_{n0}^2-\mu_{n1}^2}{2\sigma_n^2}\right]\right\}}$$

$$w_0 = \ln\left(\frac{P(y=1)}{P(y=0)}\right) + \sum_{n=1}^{N}\left[\frac{\mu_{n0}^2 - \mu_{n1}^2}{2\sigma_n^2}\right]$$

$$w_n = \frac{(\mu_{n1} - \mu_{n0})}{\sigma_n^2}$$



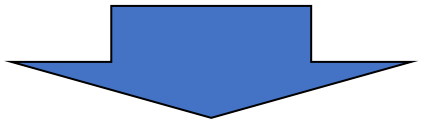
logistic shape

$$P(y=0|\mathbf{x}) = \frac{1}{1+\exp\left(w_0 + \sum_{n=1}^{N} w_n x_n\right)}$$

$$P(y=1|\mathbf{x}) = 1 - P(y=0|\mathbf{x}) = \frac{\exp\left(w_0 + \sum_{n=1}^{N} w_n x_n\right)}{1+\exp\left(w_0 + \sum_{n=1}^{N} w_n x_n\right)}$$

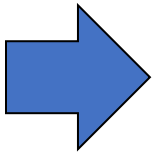
Finally, an input vector belongs to class y = 0 if

$$\frac{P(y=0|\mathbf{x})}{P(y=1|\mathbf{x})} > 1$$

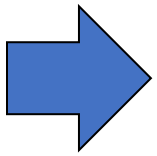


$$P(y=0|\mathbf{x}) = \frac{1}{1+\exp\left(w_0 + \sum_{n=1}^{N} w_n x_n\right)}$$

$$P(y=1|\mathbf{x}) = \frac{\exp\left(w_0 + \sum_{n=1}^{N} w_n x_n\right)}{1+\exp\left(w_0 + \sum_{n=1}^{N} w_n x_n\right)}$$



$$\exp\left(w_0 + \sum_{n=1}^{N} w_n x_n\right) < 1$$



$$w_0 + \sum_{n=1}^{N} w_n x_n < 0$$