Statistics for Machine Learning

Moritz Wolter

March 15, 2023

High-Performance Computing and Analytics Lab, University of Bonn

Overview

Foundational Statistical Concepts

Gaussian mixture models

Why statistics?

- Its useful can help us make decisions when outcomes are uncertain.
- Like getting a vaccination.
- Statistics is also an integral part of machine learning. Without it, we won't understand many machine learning methods.
- Neural networks, for example, model class probabilities in the classification case.

Today's talk is mostly based on [Has22] and some [Unp22].

Foundational Statistical Concepts

Foundational Statistical Concepts

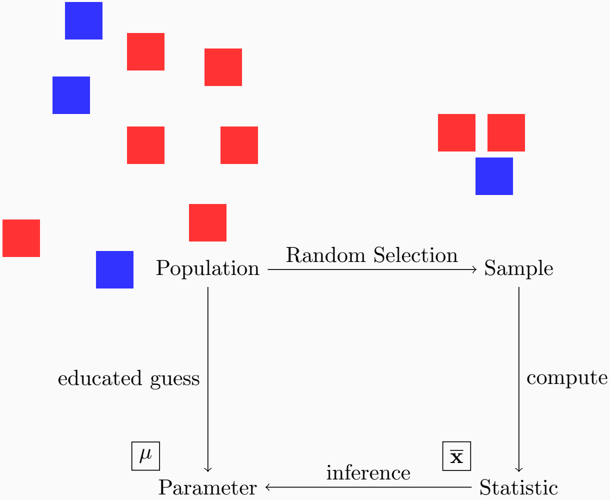
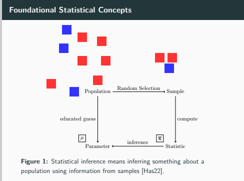


Figure 1: Statistical inference means inferring something about a population using information from samples [Has22].



Population includes all of the elements from a set of data. Sample consists of one or more observations from the population.

Parameter Characteristic of a distribution describing a population, such as the mean or standard deviation of a normal distribution. Often notated using Greek letters.

Statistic A numerical value that represents a property of a random sample. Examples of statistics are

- the mean value of the sample data.
- the range of the sample data.
- deviation of the data from the sample mean.

Important definitions [Has22]

Random Variable

A random variable X is an uncertain quantity. Its value depends on random events. A good example is the result of a dice roll.

Probability Distribution

Probability density functions are a mathematical tool to describe the randomness of data in populations and samples.

Mean

Typically everyone means the arithmetic mean when speaking about the mean,

$$\hat{\mu}_x = \frac{\sum_{i=1}^{n} x_i}{n}. \tag{1}$$

For the sample size $n \in 0, 1, 2, 3, \dots$ or $\mathbb{N}$.

`np.mean` allows you to compute the mean.

Variance

Variance measures the spread in the measurements of a random variable. It is defined as:

$$\hat{\sigma}_x^2 = \frac{\sum_{i=1}^{n}(x_i - \hat{\mu}_x)^2}{n-1}. \tag{2}$$

Again $n \in \mathbb{N}$ denotes the sample size. `np.var` implements this. The standard deviation is defined as the square root if the variance. its main advantage is that it has the same dimension as the original data [Has22],

$$\hat{\sigma}_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{\mu}_x)^2}{n-1}}. \tag{3}$$

`np.std` implements the computation of the standard deviation.

[Has22] uses $\overline{x}$ for $\hat{\mu}_x$ and s for $\hat{\sigma}_x$. Our notation is consistent width [McN16].

The Probability Density Function

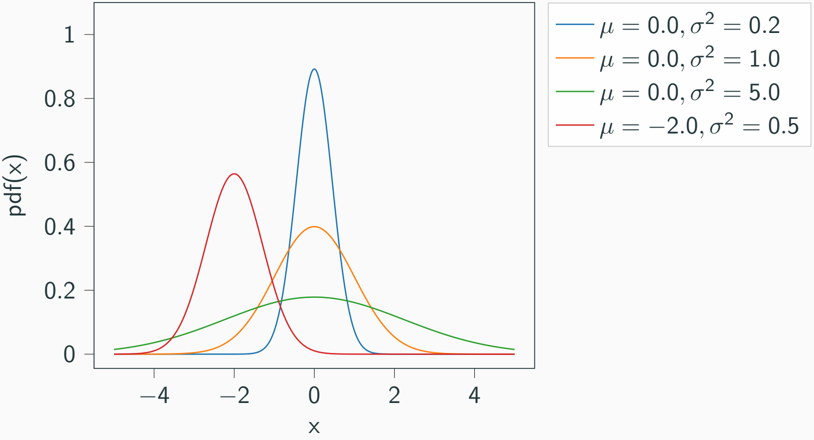


Figure 2: Normal distribution densitiy functons for different values of μ and σ. Integrating between two points on x tells us how likely the random variable will end up between those two points.

The Probability Density Function

Let p(x) be the Probability Density Function (PDF) of a random variable X. The integral over p(x) between a and b represents the probability of finding the value of X in that range [Has22].

The Probability Density Function

More formally, pdfs $p(x)$ are always positive

$$p(x) \geq 0 \; \forall x \in \mathbb{R}, \tag{4}$$

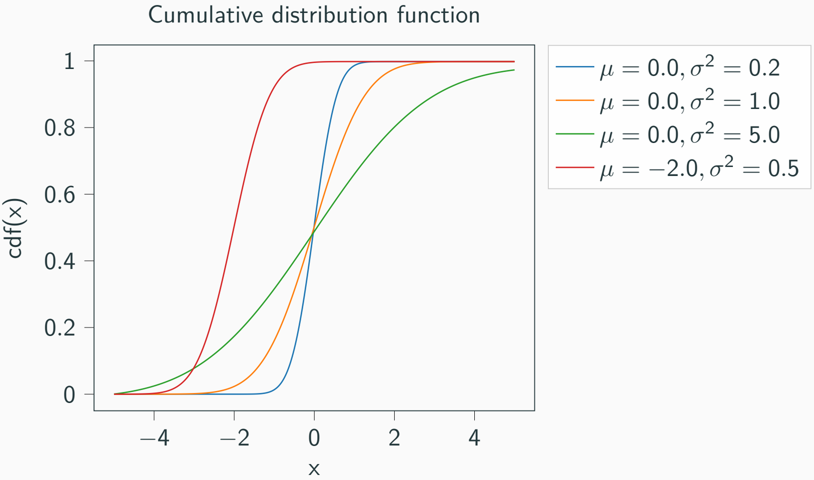
The probability for a value to end up between a and b is

$$p(a < x < b) = \int_a^b p(x) dx, \tag{5}$$

and the area under its curve must sum up to one,

$$\int_{-\infty}^{\infty} p(x) dx = 1. \tag{6}$$

The Cumulative distribution function



The Cumulative distribution function

The cumulative distribution function $P(x)$ allows us the compute the probability for a random variable X to be in a certain range.

$$P[a < X < b] = \int_a^b p(x)dx = P(b) - P(a). \tag{7}$$

Gaussian Distribution

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{8}$$

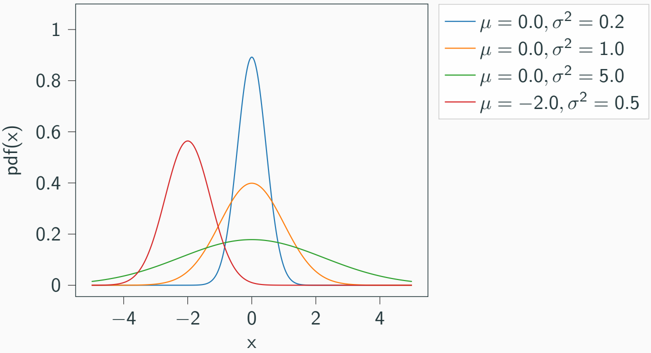


Figure 3: Plot of a Gaussian probability density function.

Uniform Distribution

$$f(x) = \begin{cases} 1/(b-a) & \text{for } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

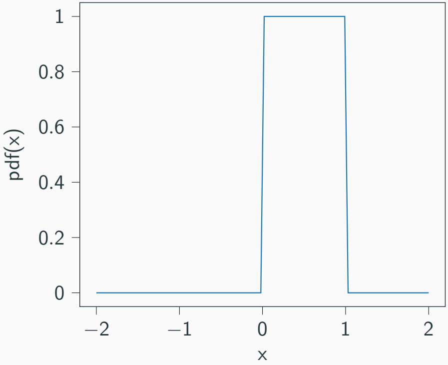


Figure 4: Plot of a uniform probability densitiy function.

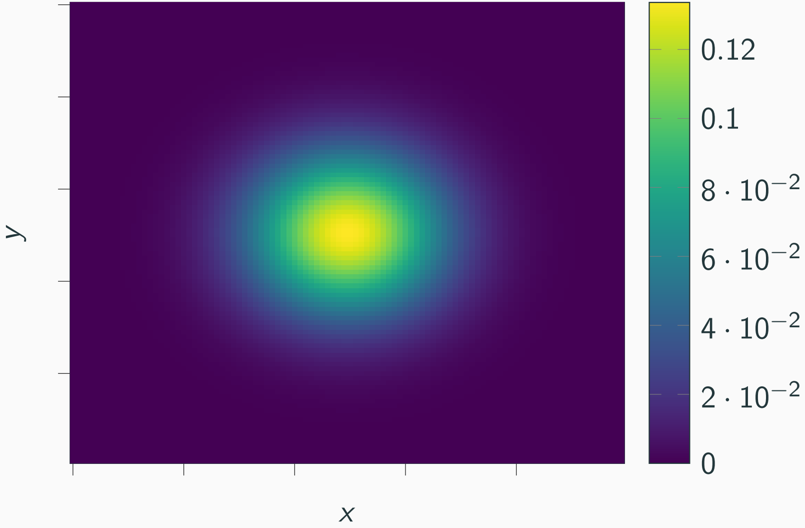
Multidimensional Gaussians

N-dimensional gaussian pdf are defined as [McN16],

$$\phi_2(\mathbf{x}|\mu_g, \Sigma_g) = \frac{1}{\sqrt{(2\pi)^N \|\Sigma_g\|}} \exp(-\frac{1}{2}(\mathbf{x} - \mu_g)^T \Sigma_g^{-1}(\mathbf{x} - \mu_g)). \tag{10}$$

$\mu_g \in \mathbb{R}^N$ denots the mean vector, $\Sigma_g \in \mathbb{R}^{N \times N}$ the covariance matrix, $^{-1}$ the matrix inverse, T the transpose and $g \in \mathbb{N}$ the number of the distrubtion, which will be important later.

The Bell curve in two dimensions



Covariance

Covariance describes how two random variables ”vary together”[Has22]. More formally,

$$\hat{\sigma}_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu}_x)(y_i - \hat{\mu}_y) \tag{11}$$

For two n sized samples x and y and real numbers x, y and μ.

Covariance Matrix

The covariance matrix of a multidimensonal variables is filled with individual variables, consider the two-dimensional case:

$$\Sigma = \begin{pmatrix} \hat{\sigma}_{xx} & \hat{\sigma}_{xy} \\ \hat{\sigma}_{yx} & \hat{\sigma}_{yy} \end{pmatrix} \tag{12}$$

Covariance Matrix

The covariance matrix of a multidimensonal variables is filled with individual variables, consider the two-dimensional case:

$$\Sigma = \begin{pmatrix} \hat{\sigma}_{xx} & \hat{\sigma}_{xy} \\ \hat{\sigma}_{yx} & \hat{\sigma}_{yy} \end{pmatrix} \tag{12}$$

$$\Sigma = (\mathbf{x}_i - \hat{\mu}_g)(\mathbf{x}_i - \hat{\mu}_g)^T \tag{13}$$

$$\Sigma = (\mathbf{x}_i - \hat{\mu}_g)(\mathbf{x}_i - \hat{\mu}_g)^T \tag{14}$$

$$= \begin{pmatrix} (\mathbf{x}_1 - \mu_1)^2 & (\mathbf{x}_1 - \mu_1)(\mathbf{x}_2 - \mu_2) & \dots & (\mathbf{x}_1 - \mu_1)(\mathbf{x}_n - \mu_n) \\ (\mathbf{x}_2 - \mu_2)(\mathbf{x}_1 - \mu_1) & (\mathbf{x}_2 - \mu_2)^2 & \dots & (\mathbf{x}_2 - \mu_2)(\mathbf{x}_n - \mu_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{x}_n - \mu_n)(\mathbf{x}_1 - \mu_1) & (\mathbf{x}_n - \mu_n)^2 & \dots & (\mathbf{x}_n - \mu_n)(\mathbf{x}_n - \mu_n) \end{pmatrix} \tag{15}$$

$$= \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{pmatrix} \tag{16}$$

Correlation

Correlation tells us how much the relationship between to random variables is linearly connected [Has22]

$$r_{xy} = \frac{\hat{\sigma}_{xy}}{\hat{\sigma}_x \hat{\sigma}_y} \tag{17}$$

$$= \frac{1}{(n-1)\hat{\sigma}_x \hat{\sigma}_y} \sum_{i=1}^{n} (x_i - \hat{\mu}_x)(y_i - \hat{\mu}_y). \tag{18}$$

Auto-Correlation

Auto-correlation [Has22] is correlation of a time delayed signal with itself. The operation is typically written as a function of the delay.

$$c_k = \frac{1}{N} \sum_{t=1}^{N-k} (x_t - \hat{\mu}_x)(x_{t+k} - \hat{\mu}_x) \tag{19}$$

For a signal of length N. To allow k to move to all possible positions zeros are typically added on both sides. In the engineering literature the normalization is typically dropped [Has22].

Auto-Correlation

autocorrelation

Gaussian mixture models

Gaussian mixture models

A Gaussian mixture model has the density [McN16]

$$f(\mathbf{x}|\theta) = \sum_{g=1}^{G} \rho_g \phi(\mathbf{x}|\mu_g, \Sigma_g). \tag{20}$$

With the normal distribution ϕ defined as before. ρ_g denotes the global probability with which a data value could originate from gaussian g. The gs number the gaussians, and G is the total number of Gaussians in the mix. We will use two. ϕ denotes the parameters μ_g and Σ_g.

Gaussian mixture models

A Gaussian mixture model has the density [McN16]

$$f(\mathbf{x}|\theta) = \sum_{g=1}^{G} \rho_g \phi(\mathbf{x}|\mu_g, \Sigma_g). \tag{20}$$

With the normal distribution ϕ defined as before. ρ_g denotes the global probability with which a data value could originate from gaussian g. The gs number the gaussians, and G is the total number of Gaussians in the mix. We will use two. ϕ denotes the parameters μ_g and Σ_g.

Typically we want as many g as we have classes in the data. I.e. one for healthy and one for diabetic. The data vectors are p dimensional $\mathbf{x} \in \mathbb{R}^p$

Likelihood

Likelihood models the probability of data originating from a distribution as a function of the parameters. The gaussian case is modelled by [McN16]

$$\mathcal{L}_c(\theta) = \prod_{i=1}^{n} \prod_{g=1}^{G} [\rho_g \phi(\mathbf{x}_i | \mu_g, \Sigma_g)]^{z_{ig}}. \tag{21}$$

We want to maximize the likelihood.
In other words, we want to transform the bells in such a way, that they explain the points as plausible as possible.

Likelihood

Likelihood models the probability of data originating from a distribution as a function of the parameters. The gaussian case is modelled by [McN16]

$$\mathcal{L}_c(\theta) = \prod_{i=1}^{n} \prod_{g=1}^{G} [\rho_g \phi(\mathbf{x}_i | \mu_g, \Sigma_g)]^{z_{ig}}. \tag{21}$$

We want to maximize the likelihood.
In other words, we want to transform the bells in such a way, that they explain the points as plausible as possible.

To maximize *phi* it needs to sit on top of the points it labels. When a gaussian sits on top of many points it's ρ_g should be large. Finally, when this works well we want a big weight from z_{ig}.

Log-Likelihood

The log-likelihood is easier to work with consider,

$$l_c(\theta) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} [\log \rho_g + \log \phi(\mathbf{x}_i | \mu_g, \Sigma_g)]. \qquad (22)$$

Now the exponent is gone, and the products turned into sums. The logs rescale the bells but do not change their maxima.

Clustering using a GMM

After guessing an initial choice for all $\hat{\mu}_g$ and $\hat{\Sigma}_g$ [McN16],

$$\hat{z}_{ig} = \frac{\rho_g \phi(\mathbf{x_i}|\hat{\mu}_g, \hat{\Sigma}_g)}{\sum_{h=1}^{G} \rho_h \phi(\mathbf{x_i}|\hat{\mu}_h, \hat{\Sigma}_h)} \tag{23}$$

tells us the probability with which point x_i came from gaussian g. It creates an association between the data points and the Gaussians. Numerically evaluation results in a matrix $\mathbf{Z} \in \mathbb{R}^{\mathsf{G}\times\mathsf{n}}$. Use it's output to select the points which belong to each class.

Clustering using a GMM

After guessing an initial choice for all $\hat{\mu}_g$ and $\hat{\Sigma}_g$ [McN16],

$$\hat{z}_{ig} = \frac{\rho_g \phi(\mathbf{x_i} | \hat{\mu}_g, \hat{\Sigma}_g)}{\sum_{h=1}^{G} \rho_h \phi(\mathbf{x_i} | \hat{\mu}_h, \hat{\Sigma}_h)} \tag{23}$$

tells us the probability with which point x_i came from gaussian g. It creates an association between the data points and the Gaussians. Numerically evaluation results in a matrix $\mathbf{Z} \in \mathbb{R}^{G \times n}$. Use it's output to select the points which belong to each class.

The z_{ig} are the true labels, $\hat{z}_{ig}$ is our estimation. The $\hat{z}_{ig}$ are the expected value of the complete data log-likelihood. Why? ϕ is a pdf. A pdf can be interpreted as providing a relative likelihood that the value of the random variable would be equal to that sample [1]. We ask for all gaussians and every point and normalize.

Fitting a GMM

Use its output to select the points which belong to each class. Optimizing the gaussian parameters θ, requires four steps per gaussian and iteration,

1. update $\hat{z}_{ig}$.
2. update $\hat{\rho}_g = n_g/n$.
3. update $\hat{\mu}_g = \frac{1}{n_g}\sum_{i=1}^{n} \hat{z}_{ig}\mathbf{x}_i$.
4. update $\hat{\Sigma}_g = \frac{1}{n_g}(\mathbf{x}_i - \hat{\mu}_g)(\mathbf{x}_i - \hat{\mu}_g)^T$.

Above n_g denotes the number of points in class g. These four steps must be repeated until the solution is good enough.

Fitting a GMM

Gauss optimization

References

[Has22] Thomas Haslwanter. *An Introduction to Statistics with Python With Applications in the Life Sciences*. 2nd ed. Springer, 2022.

[McN16] Paul D McNicholas. *Mixture model-based classification*. Chapman and Hall/CRC, 2016.

[Unp22] José Unpingco. *Python for probability, statistics, and machine learning*. 3rd ed. Springer, 2022.