Probabilistic methods

Jianxin Wu

LAMDA Group
National Key Lab for Novel Software Technology
Nanjing University, China
wujx2001@gmail.com

April 16, 2018

Contents

We will discuss a few probabilistic methods in this chapter. As the name suggests, we estimate probability functions (p.m.f. or p.d.f.) in such models, and use these probability functions to guide our decisions (e.g., classification).

This is a huge topic and tons of methods exist within this category. We will only touch the most basic concepts and methods, and provide only brief introductions (e.g., one or two sentences) to some other methods. The main purpose of this chapter is to introduce the terminology, a few important concepts and methods, and the probabilistic way of inference and decision.

1 The probabilistic way of thinking

The first thing to do is introducing the terminology, which is a little bit different from those we use in other parts of this book.

1.1 Terminology

Suppose we are given a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with training examples $\boldsymbol{x}_i$ and their associated labels y_i for all $1 \le i \le n$. Our task is to find a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$, in which $\mathcal{X}$ is the input domain, $\boldsymbol{x}_i \in \mathcal{X}$; $\mathcal{Y}$ is the domain for labels or predictions, $y_i \in \mathcal{Y}$. During the testing time, we are given any example $\boldsymbol{x} \in \mathcal{X}$, and need to predict a value $y \in \mathcal{Y}$ for it.

In the probabilistic world, we use random variables (or random vectors) X and Y to denote the above-mentioned input examples and labels. The training examples $\boldsymbol{x}_i$ are treated as samples drawn from the random vector X and in most cases, these examples are considered as sampled i.i.d. (independent and identically distributed) from X. In other words, one can treat each $\boldsymbol{x}_i$ as a random variable, but they follow the same distribution as X, and are independent to each other. This view is useful in analyzing probabilistic models and their properties. However, in this chapter, we can simply treat $\boldsymbol{x}_i$ as an example (or instantiation) drawn from the distribution following X in the i.i.d. manner. The same applies to any test example $\boldsymbol{x}$.

Since we are given (i.e., have access to, or can observe) the values of $\boldsymbol{x}_i$, the random vector X is called observable or observed. When we use diagrams or graphs to illustrate probabilistic models, an observable random variable is often drawn as a filled circle.

Similarly, the labels y_i or prediction y are samples drawn from Y. Because they are the variable(s) we want to predict (i.e., have no access to, or cannot directly observe), they are called hidden or latent variables, and are drawn as circled nodes.

The values of random variables can be of different types. For example, the label Y can be categorical. A categorical variable is also called a nominal variable, which can take value from some (two ore more) categories. For example, if $\mathcal{Y} = \{$‘male’,‘female’$\}$, then Y is categorical, with ‘male’ and ‘female’ denoting two categories, respectively. It is important to remember that these categories are orderless, that is, you cannot find a natural or intrinsic ordering of these categories. When Y is categorical, we say the task is *classification*, and the mapping f is a classification model (or a classifier).

Alternatively, Y can be real-valued, e.g., $\mathcal{Y} = \mathbb{R}$. In this case, the task is called *regression*, the mapping f is called a regression model. In statistical regression, X is also called the independent variables, and Y the dependent variable.

Y can also be a random vector, which may comprise of both discrete and continuous random variables. However, in this book, we focus on the classification task. Hence, Y is always a discrete random variable, and is always categorical (unless in rare cases where Y is explicitly specified differently.)

1.2 Distributions and inference

Let $p(X, Y)$ be the *joint* distribution for X and Y. Since we assume that Y can be somehow predicted based on X, there must be some relationships between X and Y. In other words, X and Y cannot be independent—which means that we should expect that

$$p_{X,Y}(X, Y) \neq p_X(X)p_Y(Y) \,. \tag{1}$$

If instead we have

$$p_{X,Y}(X, Y) = p_X(X)p_Y(Y) \,,$$

then knowing X is not helpful for predicting Y at all and we cannot learn any meaningful model.

The marginal distribution $p_X(x)$ is measuring the density of data X without considering the effect of Y (or, having the effect of Y integrated out from the joint).[1] It is called the *marginal likelihood*.

The marginal distribution $p_Y(y)$ is the *prior* distribution of Y when X is not considered (or not observed yet). It reflects the prior knowledge we know about Y (e.g., through domain knowledge) before any input is observed.

After we observe X, because of the relationship between X and Y, we can estimate the value of Y more precisely. That is, $p_{Y|X}(Y|X)$ (or simply $p(Y|X)$ when the meaning can be safely deduced from its context) is a better estimate of Y than $p_Y(Y)$. This distribution is called the *posterior* distribution. When given more *evidence* (samples of X), we can update our *belief* on Y. The updated belief, i.e., the posterior or the conditional distribution $p(Y|X)$ acts as the best estimate we have for Y given X.

The procedure of updating the belief (i.e., updating the posterior distribution) using the evidence is called probabilistic *inference*. We also need to decide what can we do after obtaining the posterior, hence the *decision* process follows. Classification is a typical type of decision.

[1] In the Bayesian perspective, this is the likelihood of the observed data marginalized over the parameters. We will postpone our brief introduction of the Bayesian view to a later section.

1.3 Bayes' theorem

Inference can be performed through Bayes' theorem (or Bayes' rule)

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \,. \tag{2}$$

$p(X|Y)$ is called the *likelihood*. It is also a conditional distribution. If we know $Y = y$, then the distribution of X will be different than its prior $p(X)$. For example, if we want to decide the gender of a person from his/her height, then the distribution (likelihood) of males $p(\text{height}|Y = \text{`male'})$ or females $p(\text{height}|Y = \text{`female'})$ will definitely be different from the distribution (marginal likelihood) of all people $p(\text{height})$.

Since we consider only the classification problem, $p(X|Y)$ are also the *class conditional* distributions. In short, the Bayes' theorem states that

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \,. \tag{3}$$

One thing is worth mentioning about Bayes' rule. Since the denominator $p(X)$ does not depend on Y, we can write

$$\begin{aligned} p(Y|X) &\propto p(X|Y)p(Y) \qquad (4) \\ &= \frac{1}{Z} p(X|Y)p(Y) \,, \qquad (5) \end{aligned}$$

in which $\propto$ means "proportional to" and $Z = p(X) > 0$ is a normalization constant which makes $p(Y|X)$ a valid probability distribution.

2 Choices

Now it seems we just need to estimate $p(Y|X)$ and that distribution alone will give us sufficient information to make decisions relevant to Y (given the evidence from X). There are, however, several questions remaining, e.g.,

- Shall we estimate $p(Y|X)$ using Bayes' theorem, i.e., by first estimating $p(X|Y)$ and $p(Y)$? This is equivalent to estimate the joint $p(X, Y)$. Is there any other way?
- How do we represent the distributions?
- How do we estimate the distributions?

At first glance, these questions may seem unnecessary or even trivial. However, different answers to these questions lead to different solutions or decisions, or even different conception of the world.

Next we will discuss a few important options for these questions. No matter what option is chosen, parameter estimation is the key in probabilistic methods. When the considered distribution is continuous, we use the phrase *density estimation* to refer to the estimation of the continuous distribution's density function.

2.1 Generative vs. discriminative models

If one directly models the conditional / posterior distribution $p(Y|X)$, this is a *discriminative model*. Discriminative models, however, cannot draw (or generate) a sample pair $(\boldsymbol{x}, y)$ that follows the underlying joint distribution. Generating a sample is important in some applications. Hence, one can also model the joint distribution $p(X, Y)$, which leads to a *generative model*.

In terms of classification, usually the prior distribution $p(Y)$ and the class conditional distribution $p(X|Y)$ are modeled instead. This is equivalent to model $p(X, Y)$, as

$$p(X, Y) = p(Y)p(X|Y)\,.$$

When the capability to sample from the joint (i.e., to generate instances from the joint) is not important, a discriminative model is applicable and it usually has higher classification accuracy than a generative model in practice. However, if the goal is to model the data generation process rather than classification, a generative model is necessary.

There are other options, too. We do not necessarily need to interpret the world probabilistically. To find the classification boundary (which is also called the discriminant functions) directly without considering probabilities sometimes leads to better results even than the discriminative model.

2.2 Parametric vs. nonparametric

A natural way to represent a distribution is to assume it has a specific parametric form. For example, if we assume a distribution is a normal distribution, then the p.d.f. has a fixed functional form

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)\,, \tag{6}$$

and is completely specified by two parameters: the mean $\boldsymbol{\mu}$ and the covariance matrix Σ. Hence, estimating the distribution is estimating its parameters.

Given a dataset $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ and assuming a multivariate normal distribution, we will soon show that the best maximum likelihood (ML) estimation of the parameters are

$$\boldsymbol{\mu}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i \tag{7}$$

$$\Sigma_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu}_{\text{ML}})(\boldsymbol{x}_i-\boldsymbol{\mu}_{\text{ML}})^T\,. \tag{8}$$

There are, however, different criteria for the "best" estimation. For example, when the maximum a posteriori (MAP) estimation is used, the best estimation of $\boldsymbol{\mu}$ and Σ will be different. We will discuss ML and MAP estimations later in this chapter. These methods can also be used to estimate the parameters of discrete distributions, i.e., to estimate the probability mass functions.

This family of methods for density estimation are called the *parametric* methods, because we assume specific functional forms (e.g., normal or exponential p.d.f.) and only estimate the parameters in these functions. Parametric estimation is a powerful tool when domain knowledge can hint us on the particular form of a p.d.f.

When the functional form of a continuous distribution is unknown, we can use a GMM (Gaussian Mixture Model) instead:

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} \alpha_i N(\boldsymbol{x}; \boldsymbol{\mu}_i, \Sigma_i)\,, \tag{9}$$

in which $\alpha_i \geq 0$ $(1 \leq i \leq K)$ are the mixing weights satisfying $\sum_{i=1}^{K} \alpha_i = 1$, and $N(\boldsymbol{x}; \boldsymbol{\mu}_i, \Sigma_i)$ is the i-th component multivariate Gaussian with mean $\boldsymbol{\mu}_i$ and covariance matrix Σ_i. This GMM distribution is a valid continuous distribution.

So long as we can use as many Gaussian components as we require, the GMM is a universal approximator, in the sense that it can accurately approximate any continuous distribution with high accuracy. In practice, however, it is never easy to estimate the parameters in a GMM (the α_i, $\boldsymbol{\mu}_i$, Σ_i parameters). This is a non-convex problem and we can only find a local minimum in the ML or MAP estimation.[2] A more serious problem is an accurate enough estimation may require a large K, which is computationally infeasible and requires too many training examples. And, we do not know what K value fits a particular density estimation problem.

Another family of density estimation methods are called the *nonparametric* methods, because no specific functional form is assumed for the density function. Nonparametric methods use the training examples to estimate the density at any particular point of the domain. Note that the word "nonparametric" means no parameterized functional form is assumed, but does *not* mean parameters are not needed. In fact, all training examples are parameters in such methods in addition to other possible parameters, and the number of parameters can grow towards infinity in nonparametric models.

The number of parameters in a nonparametric model usually increases when the number of training examples increases. Hence, we do not need to manually control the model complexity (such as choosing a proper K value in a GMM model). Nonparametric methods, however, usually suffer from the high computational cost. We will present a simple nonparametric method in this chapter, the kernel density estimation.

2.3 What is a parameter?

When we say the ML estimate of a Gaussian's mean is $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$, we have an implicit assumption (or we view the parameter $\boldsymbol{\mu}$ in this way): the parameter $\boldsymbol{\mu}$ is a vector whose values are fixed (i.e., without randomness), but we do not know the exact values. Hence, the maximum likelihood method uses the

[2] For example, use the EM method, which will be discussed in Chapter 14.

training data to find this set of fixed but unknown parameter values. The same interpretation applies to the MAP estimation method. This view is associated with the *frequentist view of probability.* The estimation result of these methods is a fixed point (without randomness) in the space of possible parameter values, and are called *point estimations*.

The *Bayesian interpretation of probability* interprets parameters and parameter estimation in a different manner. The parameters (e.g., $\boldsymbol{\mu}$) are also considered as random variables (or random vectors). Hence, what we should estimate are not a fixed set of values, but distributions.

Since $\boldsymbol{\mu}$ is also a random vector, it must have a prior distribution (before we observe the training set). If that prior distribution is a multivariate Gaussian, Bayes' rule leads to a Bayesian estimation of $\boldsymbol{\mu}$, which is an *entire Gaussian distribution* $N(\boldsymbol{\mu}_n, \Sigma_n)$, rather than a single fixed point. Bayesian estimation is a complex topic. We will only introduce a very simple example of Bayesian estimation in this chapter. The focus, however, is not the technique itself, but the different interpretations of these two lines of methods.

3 Parametric estimation

Since parameter estimation is the key in all sorts of parametric methods, in this section we introduce three types of parameter estimation methods: ML, MAP and Bayesian. We mainly use simple examples to introduce the ideas. Interested readers can refer to advanced textbooks for more technical details.

3.1 Maximum likelihood

The maximum likelihood (ML) estimation method is probably the simplest parameter estimation method.

Suppose we have a set of scalar training examples $D = \{x_1, x_2, \dots, x_n\}$. Furthermore, we assume they are drawn i.i.d. from a normal distribution $N(\mu, \sigma^2)$. The parameters to be estimated are denoted as $\boldsymbol{\theta}$, where $\boldsymbol{\theta} = (\mu, \sigma^2)$. The ML method estimates the parameters depending on the answer to this question:

> Given two parameters $\boldsymbol{\theta}_1 = (\mu_1, \sigma_1^2)$ and $\boldsymbol{\theta}_2 = (\mu_2, \sigma_2^2)$, how do we judge whether $\boldsymbol{\theta}_1$ is better than $\boldsymbol{\theta}_2$ (or the reverse)?

A concrete example is as follows. If

$$D = \{5.67, 3.79, 5.72, 6.63, 5.49, 6.03, 5.73, 4.70, 5.29, 4.21\}$$

follows the normal distribution with $\sigma^2 = 1$, and $\mu_1 = 0$, $\mu_2 = 5$, which one is a better choice for the μ parameter?

For a normal distribution, we know that the probability of a point bigger than 3σ plus the mean is less than 0.0015.[3] Hence, if $\mu = \mu_1 = 0$, then the probability we observe any single point in D (which are all more than 4σ away

[3] Let Φ be the c.d.f. of the standard normal distribution ($\mu = 0$ and $\sigma^2 = 1$). Then, this

from the mean) is less than 0.0015. Because these points are i.i.d. sampled, assuming $\mu = 0$, the chance or *likelihood* we will observe D is extremely small: smaller than $0.0015^{10} < 5.8 \times 10^{-29}$!

For another candidate $\mu = \mu_2 = 5$, we see that all values in D are around 5, and we will compute a much higher likelihood of observing D if $\mu = 5$. Hence, it is natural to determine $\mu_2 = 5$ is better than $\mu_1 = 0$, when we are given the dataset D and $\sigma^2 = 1$.

Formally, given a training set D and a parametric density p, we define

$$p(D|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(x_i|\boldsymbol{\theta})\,. \tag{10}$$

In our normal distribution example, we further have

$$p(D|\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)\,. \tag{11}$$

The term $p(D|\boldsymbol{\theta})$ is called the likelihood (of observing the training data D when the parameter value is fixed to $\boldsymbol{\theta}$).

However, because $\boldsymbol{\theta}$ is not a random vector, $p(D|\boldsymbol{\theta})$ is *not* a conditional distribution. This notation can be a little bit confusing in some cases. Hence, it is common to define a *likelihood function* $\ell(\boldsymbol{\theta})$:

$$\ell(\boldsymbol{\theta}) = \prod_{i=1}^{n} p(x_i|\boldsymbol{\theta})\,, \tag{12}$$

which clearly indicates that the likelihood is a function of $\boldsymbol{\theta}$. Because the function exp is involved in many density, the logarithm of $\ell(\boldsymbol{\theta})$ is very useful. It is called the log-likelihood function, and defined as

$$\ell\ell(\boldsymbol{\theta}) = \ln \ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p(x_i|\boldsymbol{\theta})\,. \tag{13}$$

If the observations are vectors, we can use the notation $\boldsymbol{x}_i$ to replace x_i.

As its name suggests, the maximum likelihood estimation solves the following optimization

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \ell\ell(\boldsymbol{\theta})\,. \tag{14}$$

The logarithm function is a monotonically increasing function, so applying it to $\ell(\boldsymbol{\theta})$ will not change the optimal estimation.

Returning to our normal distribution example, it is easy to solve the above optimization by setting the partial derivatives to 0, and get

$$\mu_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} x_i\,, \tag{15}$$

probability is

$$1 - \Phi(3) \approx 0.0013\,,$$

because we only consider the one-sided range $(\mu + 3\sigma, \infty)$.

$$\sigma_{\mathrm{ML}}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{\mathrm{ML}})^2\,. \tag{16}$$

Generalizing it to the multivariate normal distribution, Equations 7 and 8 are the ML estimates for $\boldsymbol{\mu}$ and Σ, respectively.

However, the optimization in ML estimation is not always as easy as in the above example. The ML estimation for a GMM model, for example, is non-convex and difficult. Advanced techniques such as expectation-maximization (EM) has to be adopted, which we introduce in Chapter 14.

3.2 Maximum a posteriori

The ML estimate can be accurate if we have enough examples. However, when there are only a small number of training examples, the ML estimate might suffer from inaccurate results. One remedy is to incorporate our domain knowledge about the parameters.

For example, if we know the mean μ should be around 5.5, this knowledge can be translated into a prior distribution[4]

$$p(\boldsymbol{\theta}) = p(\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma_0}\exp\left(-\frac{(\mu-5.5)^2}{2\sigma_0^2}\right)\,,$$

in which σ_0 is a relatively large number. In this example, we assume no prior knowledge about σ, and assume *a priori* that μ follows a Gaussian distribution whose mean is 5.5 and the variance σ_0^2 is large (i.e., the prior distribution is flat.)

The Maximum a posteriori (MAP) estimation then solves the following

$$\arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta})\ell(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \left\{\ln p(\boldsymbol{\theta}) + \ell\ell(\boldsymbol{\theta})\right\}\,. \tag{17}$$

The optimization is similar to what is in the ML estimate.

MAP takes into account both the prior knowledge and training data. When the number of training data (n) is small, the prior $\ln p(\boldsymbol{\theta})$ may play an important role, especially when the sampled examples are unluckily not a representative set of samples from $p(\boldsymbol{x};\boldsymbol{\theta})$. However, when there are a large number of training examples, $\ell\ell(\boldsymbol{\theta})$ will be much larger than $\ln p(\boldsymbol{\theta})$ and the effect of the prior knowledge is diluted.

Both ML and MAP are point estimation methods, which return one single optimal value for $\boldsymbol{\theta}$. In a generative model, after estimating the parameters of the joint distribution $p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta})$, we are able to calculate $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ and make decisions about $\boldsymbol{y}$. In a discriminative model, after estimating the parameters of the distribution $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$, we can also make decisions based on $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$.

[4] The phrases *a priori* and *a posteriori* are two Latin phrases, meaning conclusions that come before and after we sense observations, respectively. One example is: "This is something one knows *a priori*". In probability, our belief before sensing observations are encoded in the *prior* distribution, and the belief are updated to form the *posterior* distribution after the observations are factored in.

Note that in $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$, the inclusion of $\boldsymbol{\theta}$ after the "|" sign only indicates this density function is computed using the estimated parameter value $\boldsymbol{\theta}$, but $\boldsymbol{\theta}$ is not a random variable. Hence, $\boldsymbol{\theta}$ is put after a ";" sign.

3.3 Bayesian

In the Bayesian point of view and the Bayesian parameter estimation method, $\boldsymbol{\theta}$ is a random variable (or random vector), which means its best estimate is no longer a fixed value (vector), but an entire distribution. Hence, the output of the Bayesian estimation is $p(\boldsymbol{\theta}|D)$. Now, this is a valid p.d.f. because both $\boldsymbol{\theta}$ (the parameters) and D (with one instance sampled from D being the training set) are random vectors.

Bayesian estimation is a complex topic and we only work on a simplified example.

Given a dataset $D = \{x_1, x_2, \ldots, x_n\}$, it is interpreted as: there are n random variables $X_1, X_2, \ldots, X_n$, which are i.i.d. and x_i is sampled from X_i. Hence, D is one sample of an array of random variables. If we assume X_i is normally distributed, these random variables will follow the same normal distribution $N(\mu, \sigma^2)$. To simplify the problem, we assume σ is *known* and we only need to estimate μ. Hence, $\boldsymbol{\theta} = \mu$.

Because $\boldsymbol{\theta}$ (μ) is a random variable, it should have a prior distribution, which we assume is

$$p(\mu) = N(\mu; \mu_0, \sigma_0^2)\,.$$

To further simplify our introduction, we assume both μ_0 and σ_0 are *known*. σ_0 is usually set to a large value because the prior knowledge cannot be very certain.

We need to estimate $p(\mu|D)$. As the name suggests, Bayes' rule is key to this estimate:

$$\begin{aligned} p(\mu|D) &= \frac{p(D|\mu)p(\mu)}{\int p(D|\mu)p(\mu)\,\mathrm{d}\mu} && (18) \\ &= \alpha p(D|\mu)p(\mu) && (19) \\ &= \alpha \prod_{i=1}^{n} p(x_i|\mu)p(\mu)\,, && (20) \end{aligned}$$

in which $\alpha = \frac{1}{\int p(D|\mu)p(\mu)\,\mathrm{d}\mu}$ is a normalization constant which does not depend on μ.

This estimate involves the product of several normal p.d.f. According to the properties of normal distributions, we have[5]

$$p(\mu|D) = N(\mu_n, \sigma_n^2)\,, \tag{21}$$

in which

$$\left(\sigma_n^2\right)^{-1} = \left(\sigma_0^2\right)^{-1} + \left(\frac{\sigma^2}{n}\right)^{-1}\,, \tag{22}$$

[5] Please refer to Chapter 13 for details about this derivation.

$$\mu_n = \frac{\sigma^2/n}{\sigma_0^2 + \sigma^2/n}\mu_0 + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2/n}\mu_{\mathrm{ML}} \,. \tag{23}$$

The reciprocal of σ_n^2 is the sum of two terms: the reciprocal of the variance (uncertainty) in the prior (σ_0^2) and a weighted version of the uncertainty in the distribution ($\frac{1}{n} \times \sigma^2$). It is equivalent to:

$$\sigma_n^2 = \frac{\sigma_0^2 \times \frac{\sigma^2}{n}}{\sigma_0^2 + \frac{\sigma^2}{n}} \,. \tag{24}$$

When there are only few examples, the prior plays an important role; however, when $n \to \infty$, we have $\frac{\sigma^2}{n} \to 0$ and $\sigma_n^2 < \frac{\sigma^2}{n} \to 0$; i.e., as we have more training examples, the uncertainty about μ is reduced towards 0.

μ_n is also a weighted average of the prior mean μ_0 and the sample mean μ_{ML}, and the weights are the two uncertainties: $\frac{\sigma^2}{n}$ for μ_0 and σ_0^2 for μ_{ML}. When n is small, both the prior and the data are important components of the estimation; however, when $n \to \infty$, $\frac{\sigma^2}{n} \to 0$ and the effect of the prior disappears.

Hence, these Bayesian estimates at least match our intuition: when there are only few training examples, a proper prior distribution is helpful; when there are enough examples, the prior distribution can be safely disregarded.

An example of Bayesian estimation is shown in Figure 1. In this example, we estimate the μ parameter of a normal distribution whose $\sigma^2 = 4$. The prior for μ is $N(10, 25)$, i.e., $\mu_0 = 10$, $\sigma_0 = 5$. The training data D is generated using $\mu = 5$, and contains n examples.

As shown in Figures 1, when $n = 20$, the Bayesian estimation is quite accurate: its mode is close to 5 and the variance is small. However, when $n = 2$ the estimated density has a mean which is almost the same as that of the prior. These observations match the intuitions we deduce from the equations.

A lot can be talked about Bayesian estimation. However, we only discuss a few issues qualitatively because more detailed explanations are beyond the scope of this book.

- In the above example, when $p(D|\mu)$ is a normal distribution, we *choose* the prior for μ to be a normal distribution, then the posterior $p(\mu|D)$ is also a normal distribution, i.e., in the same functional form as the prior. This fact makes our derivation easier.

  In general, when the likelihood function $p(D|\boldsymbol{\theta})$ follows a particular distribution A, the prior $p(\boldsymbol{\theta})$ follows distribution B (B can be the same as A or different), if the posterior $p(\boldsymbol{\theta}|D)$ is also in the distribution family B (i.e., has the same functional form as the prior), we say B is a *conjugate prior* for the likelihood function A. For example, the conjugate prior of a Gaussian is again a Gaussian.

- Bayesian estimation has nice theoretical underpinnings; and the mathematics involved are usually beautiful. However, its derivation is usually much more complex than point estimation methods such as ML and MAP.

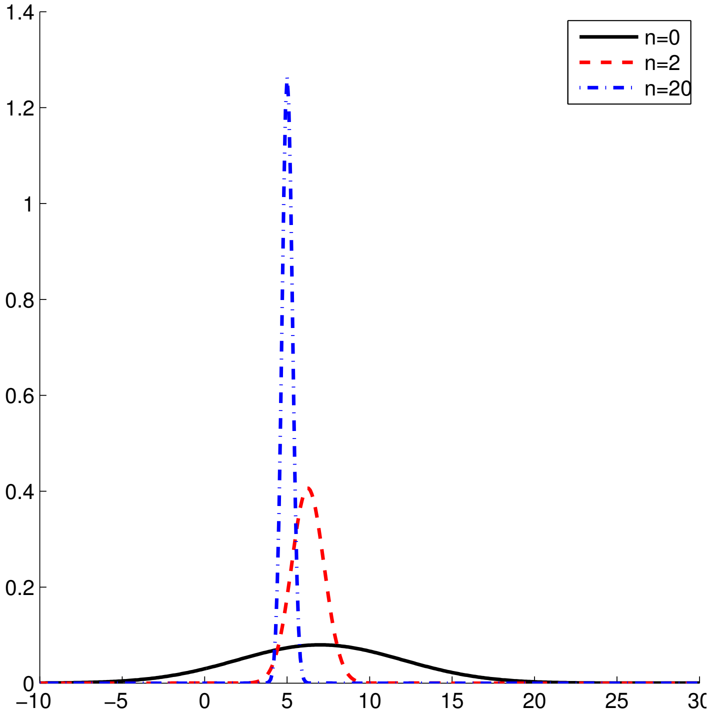


Figure 1: An illustration of the Bayesian parameter estimation. The black solid curve is the prior distribution for μ. The red dashed curve is the Bayesian estimation when $n = 2$. The blue dash-dotted curve is the Bayesian estimation when $n = 20$.

Although the conjugate prior of some likelihood distributions are known, it is hard to find the conjugate prior for an arbitrary density function.

- Integrations are involved in Bayesian estimation. When closed-form solutions are not available as a proper conjugate prior, the integration has to be done numerically or through methods such as MCMC (Markov chain Monte Carlo), which is computationally very expensive. This fact limits the scale of problems that Bayesian estimation can handle.

- When the Bayesian estimation is used in a decision process, we need to find $p(\boldsymbol{y}|D, \boldsymbol{\theta})$, which means another integration over $\boldsymbol{\theta}$ is required. Hence, it is not as convenient as the decision process using point estimation methods.

  Note that $p(\boldsymbol{y}|D, \boldsymbol{\theta})$ is a distribution (called the posterior predicative distribution). Then, $\mathbb{E}[p(\boldsymbol{y}|D, \boldsymbol{\theta})]$ can be used to guide the decision process. In some cases, the uncertainty of our decision is also required, and can be measured by $\sqrt{\mathrm{Var}(p(\boldsymbol{y}|D, \boldsymbol{\theta}))}$.

- In the Bayesian view of probability, parameters of the prior (such as μ_0 and σ_0) are also random variables. Sometimes it is necessary to define prior distributions for these parameters, and these prior distributions will in turn have parameters which have to be modeled. Hence, a Bayesian model can be hierarchical and rather complex.

- Bayesian estimation and decision are very useful when there are a small number of training examples. When we have ample training examples, its performance (e.g., accuracy in a classification problem) is usually lower than other methods (such as a discriminant function).

- Sometimes the prior distribution can be an *uninformative distribution*, which does not carry useful information in it. For example, if we know μ is between 0 and 10 (but know no further information beyond that), the prior of μ can be a uniform distribution on $[0, 10]$ (i.e., $p(\mu) = 0.1$ for $\mu \in [0, 10]$ and $p(\mu) = 0$ for $\mu < 0$ and $\mu > 10$), which does not favor any particular point in this range.

  One extreme situation is that we know nothing about μ, and set $p(\mu) = \text{const}$. This type of prior assumes uniform distribution in $\mathbb{R}$, but is not a valid probability density function; hence, it is called an *improper prior*.

4 Nonparametric estimation

Nonparametric estimation does not assume any functional form of the density. Different intuitions and ideas can lead to different nonparametric estimation approaches. In this section, we only talk about classic nonparametric density estimation, and the more advanced nonparametric Bayesian concept will not be touched.

We introduce nonparametric estimation for continuous distributions, and start from the simple one-dimensional distribution case.

4.1 A one-dimensional example

Given a set of scalar values $D = \{x_1, x_2, \ldots, x_n\}$ drawn i.i.d. from a random variable X with an underlying density $p(x)$, we want to estimate this density function.

The histogram is an excellent visualization tool that helps us examine the distribution of values in the one dimensional space. We draw 400 hundred examples from the following two-component Gaussian Mixture Model (GMM):

$$0.25N(x;0,1) + 0.75N(x;6,4)\,, \tag{25}$$

and compute three histograms with 10, 20, and 40 histogram bins, respectively, as shown in Figure 2.

The first step to build a histogram is to find the data range. Let us denote the minimum value in D as a, and the maximum value as b. Then, we can use the range $[a, b]$ as the range of possible values. We can also extend the range to $[a - \epsilon, b + \epsilon]$ to accommodate for possible variations in the data, where ϵ is a small positive number.

In the second step, we need to determine the number of bins in the histogram. If m bins are used, the range $[a, b]$ is then divided into m non-overlapping sub-ranges

$$\left[a + (i-1)\frac{b-a}{m}, \quad a + i\frac{b-a}{m}\right), \qquad 1 \le i \le m-1\,,$$

and the last sub-range is

$$\left[a + (m-1)\frac{b-a}{m}, \quad b\right]\,.$$

In fact, the assignment of the values $a + i\frac{b-a}{m}$ to either the left or the right sub-range is not important. We choose the right sub-range.

Each sub-range defines a histogram bin, and we use $Bin(i)$ to denote the i-th bin and its associated sub-range. The length of these sub-ranges, $\frac{b-a}{m}$ is the bin *width*. An m-bin histogram is a vector $\boldsymbol{h} = (h_1, h_2, \ldots, h_m)^T$, and h_i is the number of elements in D that falls in the i-th bin, i.e.,

$$h_i = \sum_{j=1}^{n} [\![x_j \in Bin(i)]\!]\,, \tag{26}$$

in which $[\![\cdot]\!]$ is the indicator function. Hence,

$$\sum_{i=1}^{m} h_i = n\,.$$

Sometimes we ℓ_1 normalize the histogram by $h_i \leftarrow \frac{h_i}{n}$ such that $\sum_{i=1}^{m} h_i = 1$ after the normalization. More details on normalization will be introduced in Chapter 9.

In Figure 2, we use stairs instead of bars to draw the histograms, which make the histograms look more similar to the true p.d.f. curve. As the figures show, although the histograms are not smooth and are different from the p.d.f. at almost every single point, the difference between the histograms and the p.d.f. is not large. In other words, the histogram is a good approximation of the p.d.f.

For example, given a value x, we can first find which bin does it belongs to. Denote $id(x)$ as the bin x falls into, we can approximate $p(x)$ as:

$$p_{\text{hist}}(x) \propto h_{id(x)} \,, \tag{27}$$

in which $\propto$ means proportional to. This equation is correct no matter the ℓ_1 normalization is used or not.

4.2 Problems with the histogram approximation

The histogram approximation has quite some problems. The following are a few important ones.

- No continuous estimation. The estimation directly from a histogram is not continuous, leaving discontinuities at the boundary of two bins. Inside each bin, a constant value is representing the entire range, which leads to large errors. Furthermore, if we need to estimate $p(x)$ but x is beyond the range of the histogram, the estimation will be 0, which is not suitable.

- Curse of dimensionality. When there are multiple dimensions, we divide each dimension into bins individually. However, suppose each dimension is divided into m bins, a distribution with d dimensions has m^d bins in total! If $m = 4$ (which is smaller than most typical m values) and $d = 100$ (which is also smaller than typical dimensionality for modern features), the number of bins is $4^{100} \approx 1.6 \times 10^{60}$.

  In other words, we need this huge number of values to describe the 100-dimensional histogram. Since 10^{60} is far exceeding the number of training examples, most of the bins will be empty, and their corresponding estimation is 0. This phenomenon is called the *curse of dimensionality*. As the number of dimensions increase linearly, the complexity of the model (e.g., histogram) increases exponentially, which makes histogram-based estimation impossible because we will never have enough training examples or computing resources to learn these values.

- The need to find a suitable bin width (or equivalently, number of bins.) Figure 2 clearly illustrates this issue. When the number of bins $m = 20$, the histogram in Figure 2b matches the true p.d.f. closely. However, when $m = 10$, the histogram in Figure 2a has clear discrepancy with the p.d.f. The complexity of the model (i.e., histogram) is lower than that of the data (i.e., the p.d.f.). Underfitting leads to inferior approximation.

  In Figure 2c, $m = 40$ leads to an overly complex histogram, which has more peaks and valleys than what the p.d.f. exhibits. It is obvious that the complex model is overfitting peculiar properties of the samples D, but not the p.d.f.

The bin width (or number of bins) in a histogram model is a hyper-parameter. It significantly affects the model's success, but there is not a good theory to guide its choice.

In low-dimensional problems, however, the histogram is a good tool to model and visualize our data. For example, if $d = 1$ or $d = 2$, the curse of dimensionality is not a problem. A histogram with properly chosen bin width can approximate a continuous distribution fairly accurately. One additional benefit is: we do not need to store the dataset D—storing the histogram $\boldsymbol{h}$ is enough.

4.3 Making your examples far-reaching

There is another perspective to examine the histograms. The histogram counts h_i reflect the accumulated contributions of all training examples in D. If we single out one particular example x_i (which falls into a bin with index $id(x_i)$), its contribution to the entire domain ($\mathbb{R}$ in 1-d) is a function $h^{x_i}(x)$:

$$h^{x_i}(x) = \begin{cases} 1 & \text{if } id(x) = id(x_i) \\ 0 & \text{otherwise} \end{cases}, \tag{28}$$

and obviously the histogram estimate $p_{\text{hist}}(x)$ (cf. Equation 27) can be computed as

$$p_{\text{hist}}(x) \propto \sum_{i=1}^{n} h^{x_i}(x)\,. \tag{29}$$

Every training example contributes to the estimate individually and independently. The manner of their contribution, however, is problematic:

- Not symmetric. There is no reason to guess the left of x_i is more important than its right, or vice versa. However, if a bin is defined as the range $[1, 2)$ and $x_i = 1.9$, then only a small range of the right hand side of x_i receives its contribution ($[1.9, 2)$, whose length is 0.1), but on the left a large range is the beneficiary of x_i ($[1, 1.9)$, whose length is 0.9).

- Finite support. As shown in the above example, only samples in the range $[1, 2)$ receive contributions from x_i, a fact that leads to the discontinuous estimate.

- Uniform radiation. In the finite range that x_i affects, the effect is uniform. No matter x is far away from or close to x_i, it receives the same contribution from x_i. This is somehow counterintuitive. We usually agree that x_i has large impact to its near neighbors, but its effect should fade away as the distance grows (and gradually reduces to 0 if the distance grows to infinity.)

In other words, we want to replace $h^{x_i}(x)$ with a continuous, symmetric, and centered (at x_i) function whose support is the entire domain (i.e., the impact of any example is far-reaching) and whose magnitude reduces along with the increase of distance to its center. And of course, the contribution function must be non-negative. In some cases, the infinite support condition can be changed to a limited but sufficiently large support condition.

4.4 Kernel density estimation

Formally, the kernel density estimation (KDE) method satisfies all these expectations. Let K be a *kernel* function that is non-negative ($K(x) \geq 0$ for any $x \in \mathbb{R}$) and integrates to 1 ($\int K(x)\,\mathrm{d}x = 1$). In addition, we also require that K has zero mean, i.e., $\int xK(x) = 0$. Then, the kernel density estimator is

$$p_{\mathrm{KDE}}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{x-x_i}{h}\right) . \tag{30}$$

A few points are worth pointing out about KDE.

- The word *kernel* has a different meaning from the word *kernel* is kernel methods (such as SVM), although some functions are valid kernels in both cases (such as the RBF/Gaussian kernel).
- The parameter $h > 0$ plays a similar role as the bin width in histogram estimation. This parameter is called the *bandwidth* in KDE. The same symbol has different meanings in these two settings (bin counts vs. bandwidth), but the context should make the distinction clear.
- Since $\int K(x)\,\mathrm{d}x = 1$, we have $\int K\left(\frac{x-x_i}{h}\right)\,\mathrm{d}x = h$ for any $h > 0$, $x_i \in \mathbb{R}$, and $\int \frac{1}{h}K\left(\frac{x-x_i}{h}\right)\,\mathrm{d}x = 1$. Because $K(x) \geq 0$, we know $p_{\mathrm{KDE}}(x) \geq 0$ and $\int p_{\mathrm{KDE}}(x)\,\mathrm{d}x = 1$, hence the kernel density estimator is a valid p.d.f.

The Epanechnikov kernel is proved to be the optimal kernel in the sense of least squared error, which is defined as

$$K(x) = \begin{cases} \dfrac{3}{4\sqrt{5}}\left(1-\dfrac{x^2}{5}\right) & \text{if } |x| < \sqrt{5} \\ 0 & \text{otherwise} \end{cases} . \tag{31}$$

This kernel has finite support.

The Gaussian kernel is probably more popular in practice, which has infinite support, as

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

for $-\infty < x < \infty$. When the bandwidth is h, the KDE is

$$p_{\mathrm{KDE}}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}h}\exp\left(-\frac{(x-x_i)^2}{2h^2}\right) . \tag{32}$$

4.5 Bandwidth selection

Bandwidth selection turns out to be much more important than kernel selection. Even though the Gaussian kernel is sub-optimal, when the bandwidth h is chosen carefully, the difference of errors between the Gaussian and the Epanechnikov kernel is small. When a wrong bandwidth is used, however, either underfitting (if h is too large) or overfitting (if h is too small) will lead to poor estimation of the density. Fortunately, for KDE both theoretical and practical guidelines exist for the choice of the bandwidth.

Under rather weak assumptions on the density to be estimated (i.e., p) and the kernel (i.e., K), the theoretically optimal bandwidth is

$$h^\star = \frac{c_1^{-2/5} c_2^{1/5} c_3^{-1/5}}{n^{1/5}}\,, \tag{33}$$

in which $c_1 = \int x^2 K(x)\,\mathrm{d}x$, $c_2 = \int K^2(x)\,\mathrm{d}x$ and $c_3 = \int (p''(x))^2\,\mathrm{d}x$.

Note that c_3 is difficult to be reliably estimated. However, if $p(x)$ is a normal distribution, a practical rule is to use

$$h^\star \approx \left(\frac{4\hat{\sigma}^5}{3n}\right)^{1/5} \approx 1.06\hat{\sigma}n^{-1/5}\,, \tag{34}$$

in which $\hat{\sigma}$ is the standard deviation estimated from the training set.

But, when the data is not similar to a Gaussian (e.g., having two or more modes) Equation 34 may lead to very poor density estimation quality. In that case, the cross validation strategy may be used to estimate the bandwidth.

KDE is continuous, has non-uniform infinite (or enough) support for each training example, is symmetric, and has guided bandwidth selection under some situations. Hence, KDE is a nice method for density estimation in one-dimension. The training examples, however, have to be stored in the KDE model and to compute p_{KDE} requires many computations.

4.6 Multivariate KDE

The extension of KDE to more dimensions, i.e., multivariate KDE, is not trivial. Let $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ be the training set and $\boldsymbol{x}_i \in \mathbb{R}^d$. The bandwidth h now becomes H, a $d \times d$ bandwidth matrix. H is required to be symmetric positive definite, i.e., $H = H^T$ and $H \succ 0$. K is the kernel function, which is centered and symmetric. Hence, we expect $K(\boldsymbol{x} - \boldsymbol{x}_i)$ to be the largest when $\boldsymbol{x} = \boldsymbol{x}_i$, and its value will decrease symmetrically (i.e., at the same speed in all directions) when $\|\boldsymbol{x} - \boldsymbol{x}_i\|$ increases.

If H is not diagonal, then applying the bandwidth matrix H will change the speed of decreasing in different directions. The bandwidth matrix is applied in the following manner

$$|H|^{-1/2} K\left(H^{-1/2}\boldsymbol{x}\right)\,, \tag{35}$$

in which $|\cdot|$ is the determinant of a matrix.[6] This transformation is performing a rotation and a scaling of the dimensions (determined by H) in the d-dimensional space. For example, if we use a multivariate Gaussian kernel, we have

$$p_{\text{KDE}}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{(2\pi)^{d/2}|H|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_i)^T H^{-1}(\boldsymbol{x}-\boldsymbol{x}_i)\right) . \tag{36}$$

In other words, it is a Gaussian mixture model with n components. The i-th component is centered at $\boldsymbol{x}_i$ and all component Gaussians share the same covariance matrix (the bandwidth matrix H).

To find the optimal H, however, is not as easy as in the 1-d case, even though it is theoretically viable. Furthermore, the computation of $p_{\text{KDE}}(\boldsymbol{x})$ is prohibitively expensive when n is large. Hence, in practice, one usually assumes a diagonal bandwidth matrix $H = \text{diag}(d_1, d_2, \ldots, d_n)$.

Diagonal GMMs are also very powerful models, in fact, universal approximator for continuous distributions. Hence, we expect a diagonal H matrix will also lead to accurate (or at least reasonable) approximation of the underlying density $p(\boldsymbol{x})$. The computation of diagonal multivariate KDE is also lighter, e.g., in the Gaussian kernel it becomes

$$p_{\text{KDE}}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{(2\pi)^{d/2}\prod_{j=1}^{d} h_j} \prod_{j=1}^{d} \exp\left(-\frac{(x_j - x_{i,j})^2}{2h_j^2}\right) , \tag{37}$$

in which x_j is the j-th dimension of a new example $\boldsymbol{x}$, and $x_{i,j}$ is the j-th dimension of the i-th training example $\boldsymbol{x}_i$.

Advanced algorithms can highly accelerate the computations in KDE and multivariate KDE, but that is beyond the scope of this book.

5 Making decisions

The estimated densities are used to make decisions, e.g., determining the class label for a test example. In this section, we only consider a simple scenario where point estimation methods are used to estimate $p(\boldsymbol{x}|y = i; \boldsymbol{\theta})$ and $p(y = i)$, in which $1 \le i \le m$ is one label in an m-class classification problem.

Under the 0-1 loss, the optimal strategy is to choose the class with highest posterior probability $p(y|\boldsymbol{x}; \boldsymbol{\theta})$ for a test example $\boldsymbol{x}$, i.e.,

$$y^\star = \underset{1\le i\le m}{\arg\max}\, p(y = i|\boldsymbol{x}; \boldsymbol{\theta}) . \tag{38}$$

We can define m discriminant functions for $1 \le i \le m$, as

$$g_i(\boldsymbol{x}) = p(y = i|\boldsymbol{x}; \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}|y = i; \boldsymbol{\theta})p(y = i)}{p(\boldsymbol{x}; \boldsymbol{\theta})} . \tag{39}$$

[6] A similar transformation is used in Chapter 13 on properties of normal distributions, when we transit from the single variable normal distribution to the multivariate one.

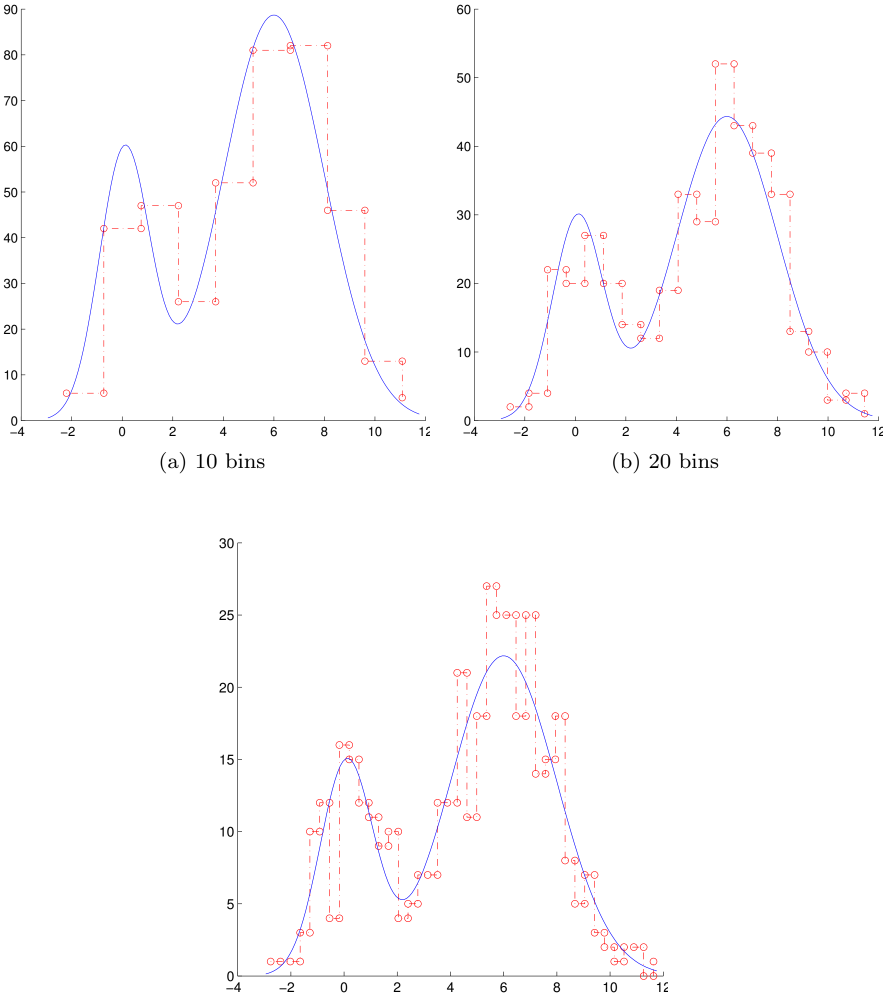
Because $p(\boldsymbol{x};\boldsymbol{\theta})$ has nothing to do with y, we can alternatively define the discriminant function as

$$g_i(\boldsymbol{x}) = p(\boldsymbol{x}|y=i;\boldsymbol{\theta})p(y=i)\,. \tag{40}$$

One further simplification is to take the logarithm, as

$$g_i(\boldsymbol{x}) = \ln\left(p(\boldsymbol{x}|y=i;\boldsymbol{\theta})\right) + \ln(p(y=i))\,, \tag{41}$$

which is useful for simplifying the equations when $p(\boldsymbol{x}|y=i;\boldsymbol{\theta})$ is in the exponential family (e.g., Gaussian). The prior for y is a discrete distribution and estimated as the percentage of examples in different classes, which is easy to handle.



(a) 10 bins

(b) 20 bins

(c) 40 bins

Figure 2: Histograms with different number of bins. The red dash-dotted curves are the histograms calculated from 400 examples. The three figures contain histograms with 10, 20 and 40 bins, respectively. The blue solid curve shows the distribution that generates the 400 data points. The blue curves are scaled to match the magnitude of the red curves in each figure.

Exercises

1. Let $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ be i.i.d. samples from an exponential distribution, whose p.d.f. is

$$p(x) = \lambda \exp(-\lambda x)[\![x \geq 0]\!] = \begin{cases} \lambda \exp(-\lambda x) & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}, \tag{42}$$

in which $\lambda > 0$ is a parameter and $[\![\cdot]\!]$ is the indicator function. Find the maximum likelihood estimate for λ.

2. (Pareto distribution) The Pareto distribution is defined by the following p.d.f.

$$p(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}}[\![x \geq x_m]\!] = \begin{cases} \dfrac{\alpha x_m^\alpha}{x^{\alpha+1}} & x \geq x_m \\ 0 & x < x_m \end{cases}, \tag{43}$$

in which $[\![\cdot]\!]$ is the indicator function. There are two parameters: a scale parameter $x_m > 0$ and a shape parameter $\alpha > 0$. We denote such a Pareto distribution as $\text{Pareto}(x_m, \alpha)$.

(a) Let X be a random variable with p.d.f.

$$p_1(x) = \frac{c_1}{x^{\alpha+1}}[\![x \geq x_m]\!],$$

in which $x_m > 0$, $\alpha > 0$, and we constrain that $c_1 > 0$ does *not* depend on x. Show that X follows $\text{Pareto}(x_m, \alpha)$. You will find this observation useful in later tasks.

(b) Let $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ be i.i.d. samples from $\text{Pareto}(x_m, \alpha)$. Find the maximum likelihood estimation for α and x_m.

(c) Let us consider a uniform distribution in the range $[0, \theta]$ with $p(x) = \frac{1}{\theta}[\![0 \leq x \leq \theta]\!]$. We want to provide a Bayesian estimate for θ. We use a set of i.i.d. examples $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ to estimate θ. Show that uniform and Pareto are conjugate distributions, that is, when the prior for θ is $p(\theta|x_m, k) = \text{Pareto}(x_m, k)$, show that the posterior $p(\theta|\mathcal{D})$ is a Pareto distribution, too. What are parameters for the posterior distribution?

To avoid confusion in the notations, we assume $m > n$. However, we want to emphasize that the notation x_m in a whole is a parameter for the Pareto prior, which does not mean the m-th element in the dataset $\mathcal{D}$.

3. Prove that the Epanechnikov kernel satisfies conditions to be used in KDE, that is, non-negative, zero mean, and the integral is 1.

4. (KDE) In this problem, we use the Matlab function `ksdensity` to obtain first hand experience with kernel density estimation.

(a) Find appropriate function(s) in Matlab to generate 1000 i.i.d. samples from the log-normal distribution. The log-normal distribution is defined by the following p.d.f.

$$p(x) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{(\ln(x)-\mu)^2}{2\sigma^2}\right) & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} . \tag{44}$$

Use $\mu = 2$ and $\sigma = 0.5$ to generate your samples.

(b) Use the `ksdensity` function to perform KDE, draw the true log-normal p.d.f. and the KDE estimation results in one figure. This function automatically chooses a bandwidth. What is the bandwidth value?

(c) In the `ksdensity` function, set the bandwidth to 0.2 and 5 and run KDE, respectively. Draw these two additional curves and compare with previous ones. What causes the differences among these curves (i.e., KDE estimation quality differences)?

(d) If you use 10,000 and 100,000 samples in the `ksdensity` function, what are the automatically chosen bandwidth values? What is the trend for the bandwidth? Explain this trend.

5. (Mean field approximation) In Equation 37, we observe that the multivariate Gaussian kernel (Equation 36) was replaced (or approximated) by a diagonal multivariate Gaussian, which is computationally much more attractive.

   This type of approximation can be generalized. Let $\boldsymbol{X} = (X_1, X_2, \ldots, X_d)$ be a multivariate distribution whose joint p.d.f. is complex. The *mean field approximation* approximates $p_X(\boldsymbol{x})$ using another random vector $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_d)$ whose components are independent, that is, assuming

$$p_X(\boldsymbol{x}) \approx p_Y(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{i=1}^{d} p_{Y_i}(y_i|\boldsymbol{\theta}) ,$$

   in which $\boldsymbol{\theta}$ are the parameters for describing Y. The task of mean field approximation is to find an optimal set of parameters $\boldsymbol{\theta}^\star$ such that $p_X(\boldsymbol{x})$ and $\prod_{i=1}^{d} p_{Y_i}(y_i|\boldsymbol{\theta}^\star)$ are as close to each other as possible.

   This strategy is widely used in *variational inference methods* in Bayesian inference, because $p_Y(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{i=1}^{d} p_{Y_i}(y_i|\boldsymbol{\theta})$ is easy to compute even when the computing of $p_X(\boldsymbol{x})$ is intractable.

   We will not introduce any variational inference details in this introductory book. However, in this problem, we try to empirically answer the following question: is the mean field approximation good enough?

   (a) Use the following Matlab/Octave code to generate a two-dimensional normal density that is non-diagonal. Read and try to understand what these codes are doing.

```
iSigma = inv([2 1; 1 4]);
pts = -5:0.1:5;
l = length(pts);
```

```
GT = zeros(l);
for i=1:l
    for j=1:l
        temp = [pts(i) pts(j)];
        % manually compute the probablity density
        GT(i,j)=exp(-0.5*temp*iSigma*temp'); %#ok<MINV>
    end
end
GT = GT / sum(GT(:)); % make it a discrete distribution
```

Note that the density was computed on a grid of points, and the last line discretizes the density into a discrete joint p.m.f.

(b) Suppose there are two independent normal random variables. They potentially have different standard deviations, but both their mean values equal 0. We can use the product of their p.d.f. to approximate the non-diagonal complex Gaussian density. To do so, we discretize the density of the product on the same grid of points. Write your own code to finish these tasks.

(c) To find the best mean field approximation, we search through possible standard deviations. Try the range 0.05 to 3 (with step size 0.05) as the search range for the two independent normal random variables. One pair of standard deviation candidates should generate a discrete joint p.m.f., denoted as `MF`. We use the following code to compute the distance between it and the distribution `GT`:

```
error = 1 - sum(min(GT(:),MF(:)));
```

Write your own code to finish the search process. What are the optimal values for the two standard deviations? What is the distance at these optimal values? Is this distance small enough such that the mean field approximation is useful?

Note that the purpose of this problem is to intuitively illustrate the usefulness of mean field approximation. In practice there are more advanced methods to find the optimal parameters than grid search.

6. In a binary classification problem, let the two class conditional distributions be $p(\boldsymbol{x}|y=i) = N(\boldsymbol{\mu}_i, \Sigma)$, $i \in \{1,2\}$. That is, the two classes are both Gaussian and sharing the same covariance matrix. Let $\Pr(y=1) = \Pr(y=2) = 0.5$, and the 0-1 loss is used. Then the prediction is given by Equation 38.

   Show that the prediction rule can be rewritten in the following equivalent form:
$$y^\star = \left\{ \begin{array}{ll} 1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} + b > 0 \\ 2 & \text{if } \boldsymbol{w}^T\boldsymbol{x} + b \le 0 \end{array} \right. . \tag{45}$$

   Give the expressions for $\boldsymbol{w}$ and b in terms of $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and Σ.

Index