Notes on Machine Learning

Paulo Eduardo Rauber

2016

1   Introduction

Machine learning is typically divided into three subareas: supervised learning, unsupervised learning, and reinforcement learning.

In broad terms, the goal of supervised learning is to learn a mapping from inputs to outputs based on examples. Consider a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \mathbb{R}$, and each high-dimensional vector $\mathbf{x}_i$ (observation) is associated to a scalar y_i (target), for every $i \in \{1, \ldots, N\}$. Each element of an observation corresponds to a feature. The task of finding a function (classifier) $\hat{f} : \mathbb{R}^D \to \mathbb{R}$ that generalizes from such a training set $\mathcal{D}$ is typical in supervised learning. Generalization is usually evaluated in a separate dataset (test set). If the set of valid targets is finite and associated to categories, the task is called classification. If the set of valid targets is a real-valued interval, the task is called regression.

The goal of unsupervised learning is to find *interesting* structure in a sequence of observations $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathbb{R}^D$. The concept of interesting is intentionally vague. Applications of unsupervised learning include finding clusters (groups of related observations), finding relationships between features, and modeling joint probabilities.

Reinforcement learning is concerned with agents that act in an environment with the goal of maximizing cumulative reward signals. These agents are not told which actions to take, but discover which actions yield most cumulative rewards by interacting with the environment. In this setting, each action may affect all subsequent rewards. Although reinforcement learning will not be covered in this text, it typically employs supervised learning for generalization.

The remainder of this section introduces important concepts in the context of supervised learning.

Probability theory is very useful to model uncertainty in machine learning, and will be reviewed in the next section. In classification, for example, instead of directly finding a function $\hat{f}$ that generalizes from a training set $\mathcal{D}$, it is also possible to model the probability $P(Y = y \mid \boldsymbol{X} = \mathbf{x})$ of target y given observation $\mathbf{x}$. The corresponding classifier $\hat{f}$ could be defined as $\hat{f}(\mathbf{x}) = \arg\max_y P(Y = y \mid \boldsymbol{X} = \mathbf{x})$.

One of the simplest classification techniques is K-nearest neighbors. In this technique, the conditional probability of a target y given an observation $\mathbf{x}$ is defined as

$$P(Y = y \mid \boldsymbol{X} = \mathbf{x}) = \frac{1}{K} \sum_{(\mathbf{x}_i, y_i) \in N(\mathbf{x}, \mathcal{D}, K)} \mathbb{I}(y_i = y),$$

where $\mathbb{I}(e) = 1$ if the logical expression e is true, and 0 otherwise; and $N(\mathbf{x}, \mathcal{D}, K)$ is a subset of $\mathcal{D}$ whose observations are the K closest to $\mathbf{x}$ (ties broken arbitrarily), according to a distance function on $\mathbb{R}^D$ (e.g., Euclidean distance). In other words, this technique defines $P(Y = y \mid \boldsymbol{X} = \mathbf{x})$ as the fraction of the K nearest neighbors of $\mathbf{x}$ that belong to class y.

Although K-nearest neighbors can generalize well given an appropriate distance function and sufficient data, it can also suffer from the *curse* of dimensionality, a difficulty that may arise in high-dimensional spaces. As an example of this curse, consider a hypercube with unit volume. A hypercube with side $l \leq 1$ inside such unit hypercube occupies a fraction $F(l) = l^D$ of the unit volume. Thus, to occupy a fraction r of the volume of the unit hypercube, a hypercube must have side $L(r) = \sqrt[D]{r}$. In the case of $D = 100$, to occupy $r = 1\%$ of the volume of the unit hypercube, the smaller hypercube must have sides larger than 0.95. This example aids the intuition that a technique like K-nearest neighbors may base its decisions on neighbors that are considerably distant in the high dimensional space (depending on the dataset). Other techniques may achieve better generalization by making particular assumptions about the dataset. For the same reason, there is no single *best* learning algorithm for every practical situation.

The misclassification rate $e(\hat{f}, \mathcal{D})$ of classifier $\hat{f}$ on dataset $\mathcal{D}$ is defined as

$$e(\hat{f}, \mathcal{D}) = \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} \mathbb{I}(\hat{f}(\mathbf{x}_i) \neq y_i).$$

Clearly, a 1-nearest neighbor classifier achieves zero misclassification rate on the training set (assuming there are contradictory observation-target pairs). This does not imply that the classifier generalizes well. Generalization

can be evaluated by the misclassification rate on a separate dataset (test set). A classifier is said to underfit when it performs poorly on both training and test sets, and to overfit when it performs poorly on the test set relative to the training set. Overfitting may be caused by a *overly complicated* model, which fails to capture the *important trends* behind the possibly *noisy* training set.

Notice that K-nearest neighbors requires fixing K before the classifier $\hat{f}$ can be fitted to the training set, which makes K a hyperparameter. The general task of choosing hyperparameters that generalize well is called model selection. As already mentioned, generalization cannot be evaluated on the training set. Hyperparameters should not be chosen based on performance on the test set either, because doing so would introduce an *optimistic* bias. In short, it would not be clear how hyperparameters chosen for a particular test set would generalize for other test sets.

A typical solution is to partition the training set into a validation set and an effective training set. For each choice of hyperparameters, the classifier is fitted to the effective training set, and evaluated in the validation set. This procedure is called cross-validation. In K-fold cross-validation, the training set is partitioned into K subsets (folds). Each fold is used as a validation set while the others are used as effective training sets, and the hyperparameters that achieve best (average) results over each validation fold are selected. The selected hyperparameters are used to fit a classifier to the whole training set, which can be finally evaluated on the test set.

2 Preliminaries

2.1 Probability theory

Probability theory is a well developed mathematical framework for reasoning under uncertainty. This section presents the fundamentals of the theory.

A sample space Ω is the set of possible outcomes of an experiment. Elements of Ω are also called sample outcomes. Events are subsets of Ω. An event is said to occur if the outcome of the experiment belongs to the event.

A set of events $\{A_1, ...\}$ is said to be disjoint if $A_i \cap A_j = \emptyset$ for every $i \neq j$. A partition of Ω is a disjoint set of events such that $\bigcup_i A_i = \Omega$.

If P is a function from the set of all subsets of Ω to $\mathbb{R}$ and has the properties:

$$P(A) \geq 0 \text{ for every event } A,$$
$$P(\Omega) = 1,$$
$$\text{If } \{A_1, ...\} \text{ is a disjoint set of events, } P(\bigcup_i A_i) = \sum_i P(A_i),$$

then P is a probability distribution over Ω^1.

For any events A and B, the following properties of a probability distribution P over Ω can be derived from the three properties above:

$$P(\emptyset) = 0,$$
$$A \subseteq B \implies P(A) \leq P(B),$$
$$0 \leq P(A) \leq 1,$$
$$P(A^c) = 1 - P(A),$$
$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Intuitively, $P(A)$ represents the degree of belief that event A will contain the sample outcome of an experiment. For instance, if the sample space is finite and there is no reason to believe that $P(\{u\}) > P(\{w\})$ for any $u, w \in \Omega$, then, for any event A, $P(A) = \frac{|A|}{|\Omega|}$, and P is said to be a uniform distribution.

An example may be useful to illustrate the application of these definitions. Suppose we are interested in computing the probability of obtaining tails exactly once after tossing two unbiased coins. The sample space can be represented as $\Omega = \{(H, H), (H, T), (T, H), (T, T)\}$. We can let $P(\{w\}) = \frac{1}{4}$ for every $w \in \Omega$, since there is no reason to believe that a sample outcome is more likely than any other. The event of obtaining tails exactly once can be represented as $A = \{(T, H), (H, T)\}$. Therefore, $P(A) = P(\{(T, H)\}) + P(\{(H, T)\}) = \frac{1}{2}$, since these two events are disjoint.

---

[1]The function P may also be defined over a set of events that contains Ω and is closed under union and complementation, but the distinction is not crucial in this text.

Let $P(B) > 0$ for an event B. The conditional probability of an event A given B is defined as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Intuitively, $P(A|B)$ can be interpreted as the degree of belief that A will occur given that B is already known to have occurred. The formula for conditional probability also gives the equality $P(A \cap B) = P(B)P(A|B)$.

Let A and B be events on Ω, with $P(B) > 0$, and $Q(A) = P(A|B)$. Then Q is a probability distribution over Ω, as it follows the three properties mentioned in the beginning of this section. Therefore:

$$P(A|B) \geq 0 \text{ for any event } A,$$
$$P(\Omega|B) = 1,$$
If $\{A_1, ...\}$ is a disjoint set of events, $P(\bigcup_i A_i | B) = \sum_i P(A_i|B),$
$$P(\emptyset|B) = 0,$$
$$A \subseteq C \implies P(A|B) \leq P(C|B),$$
$$0 \leq P(A|B) \leq 1,$$
$$P(A^c|B) = 1 - P(A|B),$$
$$P(A \cup C|B) = P(A|B) + P(C|B) - P(A \cap C|B),$$
$$P(A \cap C|B) = P(C|B)P(A|B \cap C).$$

Let $A = \{A_1, ...\}$ be a partition of Ω, then:

$$P((\bigcup_i A_i) \cap B) = P(\Omega \cap B) = P(B).$$

Let $\{A_1, ...\}$ be a partition of Ω. Then, if $P(B) \neq 0$, the definition of conditional probability can be restated as:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}.$$

The equation above is called Bayes' Theorem.

In the special case when $P(A \cap B) = P(A)P(B)$, A and B are said to be independent events.

Introduced the fundamentals, we now present random variables, which are essential to model complicated events. It is often possible to omit the sample space entirely when modeling problems using random variables.

A random variable X is a function $X : \Omega \to \mathbb{R}$. Let P be a probability distribution over Ω. Then, by definition:

$$P(X = x) = P(\{w \in \Omega \mid X(w) = x\}),$$
$$P(a < X < b) = P(\{w \in \Omega \mid a < X(w) < b\}).$$

In words, $P(X = x)$ represents the probability of an event that contains all sample outcomes that are mapped by X to x, while $P(a < X < b)$ represents the probability of an event that contains all sample outcomes that are mapped by X to a real number in the open interval (a, b). If the outcome of the experiment is w and $X(w) = x$, random variable X is said to be in state x or assigned to x.

Consider the experiment mentioned earlier of tossing two unbiased coins. We can let $X(w)$ be the number of tails in a sample outcome $w \in \Omega$. Therefore, $P(X = 0) = \frac{1}{4}$, $P(X = 1) = \frac{1}{2}$ and $P(X = 2) = \frac{1}{4}$.

When working with random variables X and Y, we write $P(X = x \cap Y = y)$ as $P(X = x, Y = y)$ for convenience.

We denote the image of a random variable X by $\text{Val}(X)$, i.e., the set of values assigned to some sample outcome. If $\text{Val}(X)$ is countable, then X is said to be a discrete random variable. If Y is also a random variable, $\text{Val}(X, Y)$ denotes the set of valid assignments to X and Y. This notation also extends to vectors of random variables.

Some properties of a probability distribution P are restated in terms of random variables below.

For a discrete random variable X distributed according to a probability distribution P,

$$\sum_{x \in \text{Val}(X)} P(X = x) = 1.$$

Also, the probability of the union of events can be restated as

$$P((X = x) \cup (Y = y)) = P(X = x) + P(Y = y) - P(X = x, Y = x),$$

and Bayes' theorem can be restated as

$$P(X = x | Y = y) = \frac{P(Y = y | X = x)P(X = x)}{\sum_{x' \in \text{Val}(X)} P(Y = y | X = x')P(X = x')}.$$

Let P be any joint probability distribution over discrete random variables $X_1, \ldots, X_n$. The chain rule of probability, which can be derived from the definition of conditional probability, gives

$$P(X_1 = x_1, \ldots, X_n = x_n) = P(X_n = x_n) \prod_{i=1}^{n-1} P(X_i = x_i | X_{i+1} = x_{i+1}, \ldots X_n = x_n),$$

for every $(x_1, \ldots, x_n) \in \text{Val}(X_1, \ldots, X_n)$.

If P is a probability distribution over Ω and X is a discrete random variable over Ω, we define the probability mass function (pmf) f of X as $f(x) = P(X = x)$, for every x.

If X is a continuous random variable and f is a function such that $f(x) \geq 0$ for all x, $\int_{-\infty}^{\infty} f(x) \, dx = 1$, $P(a < X < b) = \int_a^b f(x) \, dx$ for every $a < b$, then f is called a probability density function (pdf) of X. It is important to note that $f(x)$ does not represent $P(X = x)$ in the continuous case. The remainder of this section presupposes continuous pdfs.

We let $X \sim f$ denote that variable X is distributed according to the probability mass/density function f.

In the discrete case, a function $f_{X,Y}$ such that $f_{X,Y}(x, y) = P(X = x, Y = y)$ for every x and y is called a joint probability mass function of X and Y. A function $f_{X|Y}$ such that $f_{X|Y}(x, y) = P(X = x | Y = y) = P(X = x, Y = y)/P(Y = y)$ for every x and y is called a conditional mass function of X given Y. It is also common to denote $f_{X|Y}(x, y)$ as $f_{X|Y}(x \mid y)$.

In the continuous case, if $f_{X,Y}$ is a joint probability density function of X and Y, then

$$P(X \in [a, b], Y \in [c, d]) = \int_a^b \int_c^d f_{X,Y}(x, y) dy \, dx,$$

for every choice of $a, b, c, d \in \mathbb{R}$.

The conditional density function of X given Y is defined as $f_{X|Y}(x|y) = f_{X,Y}(x, y)/f_Y(y)$ for every x and y such that $f_Y(y) > 0$, where f_Y is the pdf for Y. In this case, $P(X \in [a, b] \mid Y = y)$ is defined as

$$P(X \in [a, b] \mid Y = y) = \int_a^b f_{X|Y}(x|y) \, dx.$$

If $f_{X,Y}$ is a joint pmf for X and Y and f_X is the pmf for X, $f_X(x) = \sum_y f_{X,Y}(x, y)$, for every x. Analogously, if $f_{X,Y}$ is a joint pdf for X and Y and f_X is the pdf for X, $f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) \, dy$, for every x. In both cases, obtaining f_X from $f_{X,Y}$ is called marginalizing over Y.

It is common to omit the name of a random variable when denoting probabilities. For instance, $P(X = x, Y = y) = P(X = x | Y = y)P(Y = y)$ can be written as $P(x, y) = P(x|y)P(y)$. We will often let the context determine whether a function p is a probability density or mass function. Also, when there is no risk of ambiguity, probability functions will omit the subscripts (e.g., $p_{X,Y|Z}(x, y|z)$ will be written as $p(x, y|z)$).

Consider a random vector $\mathbf{X} = (X_1, \ldots, X_D)$, where each X_i is a random variable. The notation $\mathbf{X} \sim p$ indicates that p is a joint probability function for $\mathbf{X}$. We denote the corresponding probability mass/density of $\mathbf{x} \in \text{Val}(\mathbf{X})$ by $p(\mathbf{x})$.

Let X, Y and Z be random variables, and p a probability (mass/density) function. Variable X is independent of Y given Z, written as $X \perp\!\!\!\perp Y \mid Z$, if and only if $p(x, y|z) = p_{X|Z}(x|z)p_{Y|Z}(y|z)$ for all x, y and z such that $p_Z(z) > 0$. Intuitively, this means that knowing the state of Y does not give additional information about the state of X when the state of Z is already known. The definition of unconditional independence (e.g., $X \perp\!\!\!\perp Y$) is analogous. It is easy to show that $X \perp\!\!\!\perp Y \mid Z$ if and only if there exist functions g and h such that $p_{X,Y|Z}(x, y|z) = g(x, z)h(y, z)$ for every x, y and z such that $p_Z(z) > 0$.

Consider a random variable X distributed according to P. An α-quantile is defined as an x such that $P(X \leq x) = \alpha$. A 0.5-quantile is called a median.

The expected value (mean) $\mathbb{E}[X]$ of a discrete random variable $X \sim p$ is defined as $\mathbb{E}[X] = \sum_x xp(x)$. In the case of a continuous random variable $X \sim p$, $\mathbb{E}[X] = \int_{-\infty}^{\infty} xp(x)\ dx$ (assuming the integral exists).

Consider a continuous random variable X, and a function $f : \mathbb{R} \to \mathbb{R}$. The function $f \circ X : \Omega \to \mathbb{R}$, defined as $(f \circ X)(w) = f(X(w))$ is also a random variable. In this case, it may be possible to compute the expectation $\mathbb{E}[f \circ X] = \int_{-\infty}^{\infty} f(x)p(x)\ dx$, which is usually denoted by $\mathbb{E}[f(X)]$. The discrete case is analogous.

In the case of a continuous random vector $\mathbf{X} = (X_1, \ldots, X_D)$ and a function $f : \mathbb{R}^D \to \mathbb{R}$, the expectation is defined using the joint distribution: $\mathbb{E}[f(\mathbf{X})] = \int_{\mathrm{Val}(\mathbf{X})} f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$. The discrete case is analogous.

Expectations over random vectors or random matrices should be considered element-wise. For instance, if $\mathbf{X} = (X_1, \ldots, X_D)$, $\mathbb{E}[\mathbf{X}] = (\mathbb{E}[X_1], \ldots, \mathbb{E}[X_D])$.

The variance of a random variable X is defined as $\mathrm{var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$. The standard deviation of a random variable X is defined as $\mathrm{std}[X] = \sqrt{\mathrm{var}[X]}$. A high variance indicates that a random variable is likely to assume values that deviate highly from the expected value.

The covariance between random variables X and Y is defined as $\mathrm{cov}[X, Y] = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$. It is easy to show that $\mathrm{cov}[X, Y] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$. The (Pearson) correlation coefficient between random variables X and Y is defined as $\mathrm{corr}[X, Y] = \mathrm{cov}[X, Y]/(\mathrm{std}[X]\,\mathrm{std}[Y])$ when $\mathrm{std}[X]\,\mathrm{std}[Y] > 0$, and is always between -1 and 1. A high correlation coefficient (in absolute value) indicates a highly linear relationship between X and Y, and the sign indicates whether the variables are proportional or inversely proportional. In particular, $|\mathrm{corr}[X, Y]| = 1$ implies $Y = aX + b$ for some $a, b \in \mathbb{R}$, where $a \neq 0$. If $X \perp Y$, then $\mathrm{corr}[X, Y] = 0$. However, the converse is not generally true, and the correlation coefficient is not an appropriate measure of dependence between random variables.

The covariance matrix of a random vector $\mathbf{X} = (X_1, \ldots, X_D)$ is defined as $\mathrm{cov}[\mathbf{X}] = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^T]$, where $\mathbf{Z}^T$ denotes the random row matrix corresponding to a random vector $\mathbf{Z}$. In other words, $\mathrm{cov}[\mathbf{X}]_{i,j} = \mathrm{cov}[X_i, X_j]$. The correlation matrix is analogous.

In this text, we write the dot product between vectors $\mathbf{u}$ and $\mathbf{v}$ as $\mathbf{uv}$, since that is usually unambiguous. Operations between matrices and vectors consider the vector as a column matrix, unless stated otherwise.

If X is a random variable and $Y = aX + b$ for $a, b \in \mathbb{R}$, it is easy to show that $\mathbb{E}[Y] = a\mathbb{E}[Y] + b$. More generally, if $\mathbf{X}$ is a random vector, and $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b}$ for a matrix $\mathbf{A}$ and vector $\mathbf{b}$, it is possible to show that $\mathbb{E}[\mathbf{Y}] = \mathbf{A}\mathbb{E}[\mathbf{X}] + \mathbf{b}$, and that $\mathrm{cov}[\mathbf{Y}] = \mathbf{A}\,\mathrm{cov}[\mathbf{X}]\mathbf{A}^T$

Consider a dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$ where each $\mathbf{x}_i$ is obtained by sampling $\mathbf{X}_i \sim p$, for every i. Also, let $\mathbf{X}_{-i} = [\cup_j \mathbf{X}_j] - \mathbf{X}_i$ and suppose $\mathbf{X}_i \perp \mathbf{X}_{-i}$, for every i. In this case, $\mathcal{D}$ is said to be independent, identically distributed (iid) according to p. As a consequence of these assumptions,

$$p(\mathcal{D}) = p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{i=1}^{N} p(\mathbf{x}_i),$$

and $p(\mathcal{D})$ is the so-called likelihood of the dataset $\mathcal{D}$. The definition is the same whether p is a probability mass or density function. It should be noted that p denotes two distinct probability functions on the equation above, which are unambiguous given their arguments. As already mentioned, such abuse of notation will be common.

Consider an iid dataset $\mathcal{D} = x_1, \ldots, x_N$ distributed according to p. If $X \sim p$ and $\mathbb{E}[X]$ exists and is finite, the law of large numbers guarantees that $N^{-1} \sum_{i=1}^{N} x_i \to \mathbb{E}[X]$ as $N \to \infty$. In other words, the mean of a (very large) dataset may be a good approximation to the expected value of X. This is a particular case of Monte Carlo approximation, which can be used to approximate the expected value of any function of random variables.

For instance, if μ is a Monte Carlo approximation of $\mathbb{E}[X]$ considering the dataset $\mathcal{D}$, the variance $\mathrm{var}[X]$ of X can be approximated by $N^{-1} \sum_{i=1}^{N} (x_i - \mu)^2$. It can be shown that this approximation underestimates $\mathrm{var}[X]$ (particularly when N is small), precisely because it also requires an estimate of $\mathbb{E}[X]$.

Consider a discrete random variable $X \sim p_X$ and let $Y = f(X)$. Naturally, $p_Y(y) = \sum_{x|f(x)=y} p_X(x)$.

Now consider a continuous random variable $X \sim p_X$. If $Y = f(X)$ for an invertible function $f : \mathbb{R} \to \mathbb{R}$, $g = f^{-1}$, and g is differentiable, then $p_Y(y) = p_X(g(y))|g'(y)|$.

For the general case, consider a continuous random vector $\mathbf{X} \sim p_X$, and let $\mathbf{Y} = f(\mathbf{X})$ for an invertible function $f : \mathbb{R}^D \to \mathbb{R}^D$, and suppose $g = f^{-1}$ is differentiable. Let $J_{\mathbf{y} \mapsto \mathbf{x}}(\mathbf{y})$ denote a matrix such that $[J_{\mathbf{y} \mapsto \mathbf{x}}(\mathbf{y})]_{i,j} = \frac{\partial x_i}{\partial y_j}$ at point $\mathbf{y}$. We say $J_{\mathbf{y} \mapsto \mathbf{x}}(\mathbf{y})$ the Jacobian matrix of g at $\mathbf{y}$. Under these assumptions, it can be shown that

$$p_Y(\mathbf{y}) = p_X(g(\mathbf{y}))|\det J_{\mathbf{y} \mapsto \mathbf{x}}(\mathbf{y})|.$$

2.2 Notable probability distributions

This section introduces probability distributions that will be useful in the remainder of this text. We will define each probability function p on a subset of its domain whose complement has every element mapped to zero. Notice that the properties presented in the previous section implicitly required continuous pdfs, while some pdfs presented here are continuous only in a particular subset of their domains, requiring additional care.

The binomial (conditional) pmf $\text{Bin}(\cdot \mid n, \theta)$ is defined on $\{0, \ldots, n\}$ as

$$\text{Bin}(x \mid n, \theta) = \binom{n}{x} \theta^x (1 - \theta)^{n-x},$$

where $\theta \in [0, 1]$, and $n \in \mathbb{N}$. If $X \sim \text{Bin}(\cdot \mid n, \theta)$, $P(X = x)$ can be intuitively interpreted as the probability of obtaining exactly x *successes* in n independent trials of an experiment that has probability θ of success in each trial (and probability $1 - \theta$ of failure).

The Bernoulli pmf $\text{Ber}(\cdot \mid \theta)$ is defined on $\{0, 1\}$ as

$$\text{Ber}(x \mid \theta) = \theta^x (1 - \theta)^{1-x},$$

where $\theta \in [0, 1]$. Clearly, $\text{Ber}(x \mid \theta) = \text{Bin}(x \mid 1, \theta)$.

The multinomial (joint) pmf $\text{Mu}(\cdot \mid n, \boldsymbol{\theta})$ is defined on $\{\mathbf{x} \in \mathbb{N}^D \mid \sum_i x_i = n\}$ as

$$\text{Mu}(\mathbf{x} \mid n, \boldsymbol{\theta}) = \frac{n!}{x_1! \cdots x_D!} \prod_{i=1}^{D} \theta_i^{x_i},$$

where $\boldsymbol{\theta} \in [0, 1]^D$, $\sum_i \theta_i = 1$, and $n \in \mathbb{N}$. Intuitively, if $\mathbf{X} \sim \text{Mu}(\cdot \mid n, \boldsymbol{\theta})$, $P(\mathbf{X} = \mathbf{x})$ is the probability of obtaining, jointly for every i, exactly x_i results of type i in n independent trials of an experiment that has probability θ_i of being of type i in each trial.

The categorical pmf $\text{Cat}(\cdot \mid \boldsymbol{\theta})$ is defined on $\{1, \ldots, D\}$ as

$$\text{Cat}(x \mid \boldsymbol{\theta}) = \text{Mu}(e_x \mid 1, \boldsymbol{\theta}) = \theta_x,$$

where $\boldsymbol{\theta} \in [0, 1]^D$, $\sum_i \theta_i = 1$, and e_x is a vector that is zero in every element except the x-th, which is 1.

The Poisson pmf $\text{Poi}(\cdot \mid \lambda)$ is defined on $\mathbb{N}$ as $\text{Poi}(x \mid \lambda) = \frac{\lambda^x}{x!} e^{-\lambda}$, where $\lambda > 0$. If $X \sim \text{Poi}(\cdot \mid \lambda)$, $\text{Poi}(x \mid \lambda)$ is intuitively the probability that x *events* will occur in a fixed interval of *time*, if λ events are expected to occur in this interval, and the occurrence of an event is independent of the time since the previous occurrence.

The empirical pmf $p_{\text{emp}}(\cdot \mid \mathcal{D})$ is defined on the set of elements in the (sequence) dataset $\mathcal{D}$ as

$$p_{\text{emp}}(\mathbf{x} \mid \mathcal{D}) = \frac{1}{N} \sum_{\mathbf{x}_i \in \mathcal{D}} \mathbb{I}[\mathbf{x} = \mathbf{x}_i] = \frac{N_{\mathbf{x}}}{N},$$

where N is the number of elements in $\mathcal{D}$, and $N_{\mathbf{x}}$ is the number of occurrences of $\mathbf{x}$ in $\mathcal{D}$.

The uniform pdf $\text{Unif}(\cdot \mid a, b)$ is defined on $[a, b]$ as

$$\text{Unif}(x \mid a, b) = \frac{1}{b - a},$$

where $a, b \in \mathbb{R}$, and $a < b$. Intuitively, if $X \sim \text{Unif}(\cdot \mid a, b)$, the probability of X being in any two same-length subintervals of $[a, b]$ is the same.

The Gaussian pdf $\mathcal{N}(\cdot \mid \mu, \sigma^2)$ is defined on $\mathbb{R}$ as

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where μ is called mean, and $\sigma^2 > 0$ is called variance. Indeed, if $X \sim \mathcal{N}(\cdot \mid \mu, \sigma^2)$, $\mathbb{E}[X] = \mu$ and $\text{var}[X] = \sigma^2$. The Gaussian pdf is symmetric about its mean, where it also achieves its single maximum. The inverse of the variance $\lambda = \sigma^{-2}$ is also called precision. Intuitively, higher precision indicates higher probability of the variable being close to the mean. The Standard Gaussian pmf is defined as $\mathcal{N}(\cdot \mid 0, 1)$.

The Gaussian distribution is very important in probability, partly due to the central limit theorem. Consider a sequence of iid random variables $X_1, \ldots, X_N$ (sample), with finite expected value $\mu = \mathbb{E}[X_i]$, and finite non-zero

variance $\sigma^2 = \text{var}[X_i]$. Also, let $\bar{X} = \frac{1}{N}\sum_i X_i$ be the so-called sample mean, which is also a random variable. It is possible to show that $\bar{X}$ becomes approximately distributed according to $\mathcal{N}(\cdot \mid \mu, N^{-1}\sigma^2)$ as $N \to \infty$. In other words, for a large sample, the sample mean becomes distributed according to a Gaussian distribution that has the same expected value as any of the (iid) variables in the sample, and a precision that is proportional to N.

The multivariate Gaussian joint pdf $\mathcal{N}(\cdot \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is defined on $\mathbb{R}^D$ as

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})},$$

where $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean vector and $\boldsymbol{\Sigma}$ is the $D \times D$ (positive definite) covariance matrix. Indeed, if $\mathbf{X} \sim \mathcal{N}(\cdot \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, it is possible to show that $\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu}$ and $\text{cov}[\mathbf{X}] = \boldsymbol{\Sigma}$. The matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ is also called precision matrix.

The Gamma function Γ is defined on $\mathbb{R}$ as

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\ dx.$$

It is possible to show that $\Gamma(z) = (z-1)\Gamma(z-1)$, for every $z \in \mathbb{R}$. If $z \in \mathbb{N}$, $\Gamma(z) = (n-1)!$.

The (non-standardized) Student's t pdf $\mathcal{T}(\cdot \mid \mu, \sigma^2, \nu)$ is defined on $\mathbb{R}$ as

$$\mathcal{T}(x \mid \mu, \sigma^2, \nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}\left(1 + \frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}},$$

where μ is the mean, σ^2 is a scale parameter (not variance), and $\nu > 0$ is called degrees of freedom. Indeed, if $X \sim \mathcal{T}(\cdot \mid \mu, \sigma^2, \nu)$ and $\nu > 1$, $\mathbb{E}[X] = \mu$. Also, if $\nu > 2$, $\text{var}[X] = \frac{\nu}{\nu-2}\sigma^2$. The Student's t pdf is symmetric about its mean, and has *heavier tails* than the Gaussian pdf (specially for small ν). In other words, compared to a Gaussian pdf with the same expected value and variance, the Student's t pdf assigns higher density to outliers.

The Laplace pdf $\text{Lap}(\cdot \mid \mu, b)$ is defined on $\mathbb{R}$ as

$$\text{Lap}(x \mid \mu, b) = \frac{1}{2b}e^{-\frac{|x-\mu|}{b}},$$

where $b > 0$. If $X \sim \text{Lap}(\cdot \mid \mu, b)$, $\mathbb{E}[X] = \mu$ and $\text{var}[X] = 2b^2$. The Laplace distribution is symmetric about its mean, and has heavier tails than the Gaussian.

The Gamma pdf $\text{Ga}(\cdot \mid a, b)$ is defined on $\mathbb{R}_{>0}$ as

$$\text{Ga}(t \mid a, b) = \frac{b^a}{\Gamma(a)}t^{a-1}e^{-tb},$$

where $a > 0$ is the shape, and $b > 0$ is the rate. If $X \sim \text{Ga}(\cdot \mid a, b)$, $\mathbb{E}[X] = \frac{a}{b}$, and $\text{var}[X] = \frac{a}{b^2}$. The pdf has a maximum (mode) at $\frac{a-1}{b}$ for $a \geq 1$. The Gamma pdf is very flexible, and has several important pdfs as special cases.

The exponential pdf $\text{Expon}(\cdot \mid \lambda)$ is defined as $\text{Expon}(t \mid \lambda) = \text{Ga}(t \mid 1, \lambda)$. Intuitively, if λ *events* are expected to occur in a unit of *time*, and the occurrence of an event is independent of the time since the previous occurrence, $\text{Expon}(t \mid \lambda)$ is the density for the event occurring at time t.

The chi-squared pdf $\chi^2(\cdot \mid \nu)$ is defined as $\chi^2(x \mid \nu) = \text{Ga}(x \mid \frac{\nu}{2}, \frac{1}{2})$, where $\nu \in \mathbb{N}$ are the degrees of freedom. The importance of this distribution comes from the fact that if $Z_1, \dots, Z_\nu$ are iid according to a standard Gaussian pdf and $S = \sum_i Z_i^2$, then $S \sim \chi^2(\cdot \mid \nu)$.

The beta pdf $\text{Beta}(\cdot \mid a, b)$ is defined on $[0, 1]$ as

$$\text{Beta}(x \mid a, b) = \frac{1}{B(a, b)}x^{a-1}(1-x)^{b-1},$$

where $a, b > 0$, and the beta function B is defined as $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$. If $X \sim \text{Beta}(\cdot \mid a, b)$, $\mathbb{E}[X] = \frac{a}{a+b}$, and $\text{var}[X] = \frac{ab}{(a+b)^2(a+b+1)}$. If $a = b = 1$, the beta pdf is equal to the uniform pdf on $[0, 1]$. If $a = b$ and $0 < a < 1$, the distribution has modes at 0 and 1. This pdf will be discussed in the context of parameter estimation.

The Dirichlet pdf $\text{Dirichlet}(\cdot \mid \boldsymbol{\alpha})$ is defined over the so-called probability simplex $S_D = \{\mathbf{x} \mid 0 < x_i < 0$ for all $i, \sum_{i=1}^D x_i = 1\}$ as

$$\text{Dirichlet}(\mathbf{x} \mid \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{D} x_i^{\alpha_i - 1},$$

where $\alpha_i > 0$ for all i, $\alpha_0 = \sum_{i=1}^{D} \alpha_i$, and the generalized Beta function B is defined as $B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{D} \Gamma(\alpha_i)}{\Gamma(\alpha_0)}$. If $\mathbf{X} \sim \text{Dirichlet}(\cdot \mid \boldsymbol{\alpha})$, $\mathbb{E}[X_i] = \frac{\alpha_i}{\alpha_0}$, and $\text{var}[X_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$, for all i. This pdf will also be discussed in the context of parameter estimation.

The Pareto pdf $\text{Pareto}(\cdot \mid k, m)$ is defined on $[m, \infty)$ as

$$\text{Pareto}(x \mid k, m) = \frac{km^k}{x^{k+1}},$$

where $k, m > 0$. If $X \sim \text{Pareto}(\cdot \mid k, m)$ and $k > 1$, $\mathbb{E}[X] = \frac{k}{k-1} m$. This pdf places high density at m, which decreases fast (particularly when k is large).

2.3 Information theory

The entropy (in bits) $H[X]$ of a discrete random variable $X \sim p$ is defined as

$$H[X] = \mathbb{E}[-\log_2 p(X)] = -\sum_k p(k) \log_2 p(k),$$

where $p(k) > 0$ for every k. If $p(k) = 0$ for some k, $0 \log_a 0$ is conventionally replaced by 0, which is justified by the fact that $\lim_{a \to 0} a \log_2 a = 0$. It is also common to denote $H[X]$ by $H[p]$.

Consider the task of transmitting a sequence of assignments to N iid discrete random variables, and let $X \sim p$ denote one of these variables. Shannon's source coding theorem states that, as $N \to \infty$, $H[X]$ is a lower bound on the average number of bits required to transmit an assignment (after the encoding is established).

The cross-entropy $H[p, q]$ between two pmfs p and q is defined as $H[p, q] = -\sum_k p(k) \log_2 q(k)$, where $0 \log_a 0$ is again replaced by 0. Considering the transmission task mentioned above, the cross-entropy $H[p, q]$ can be interpreted as the average number of bits required to transmit an assignment when the encoding is ideal for q (instead of p).

The Kullback-Leibler divergence $\text{KL}(p||q)$ between two pmfs p and q is defined as

$$\text{KL}(p||q) = \sum_k p(k) \log_2 \frac{p(k)}{q(k)} = H[p, q] - H[p],$$

where $p(k), q(k) > 0$ for every k. If either $p(k)$ or $q(k)$ are 0, the corresponding term is conventionally replaced by 0. Considering again the transmission task, the Kullback-Leibler divergence $\text{KL}(p||q)$ can be interpreted as the increase in the average number of bits required to transmit an assignment when the encoding is ideal for q (instead of p).

We now show that $\text{KL}(p||q) \geq 0$ for any two pmfs p and q, assuming $p(k), q(k) > 0$ for every k. By definition,

$$-\text{KL}(p||q) = -\sum_k p(k) \log_2 \frac{p(k)}{q(k)}$$
$$= \sum_k p(k) \log_2 \frac{q(k)}{p(k)}.$$

Jensen's inequality states that if f is a concave function and $\lambda_1, \ldots, \lambda_N$ are positive numbers such that $\sum_i \lambda_i = 1$, then $f(\sum_i \lambda_i x_i) \geq \sum_i \lambda_i f(x_i)$. Using this result,

$$-\text{KL}(p||q) \leq \log_2 \left[\sum_k p(k) \frac{q(k)}{p(k)} \right] = \log_2 1 = 0.$$

Therefore, $-\text{KL}(p||q) \leq 0$, and $\text{KL}(p||q) \geq 0$.

We now show that a discrete random variable has maximum entropy when it is distributed uniformly. Firstly, note that a pmf q defined as $q(k) = \frac{1}{K}$ for $k \in \{1, \ldots, K\}$ has entropy

$$H[q] = -\sum_{k=1}^{K} \frac{1}{K} \log_2 \frac{1}{K} = -\log_2 \frac{1}{K} = \log_2 K.$$

Let p be a pmf that is non-zero on $D = \{1, \ldots K\}$. By definition:

$$\mathrm{KL}(p||q) = \sum_k p(k) \log_2 \frac{p(k)}{q(k)}$$

$$= \sum_k p(k) \log_2 \frac{p(k)}{\frac{1}{K}}$$

$$= \sum_k p(k) \log_2 p(k) - \sum_k p(k) \log_2 \frac{1}{K}$$

$$= -H[p] + \log_2 K.$$

Because $\mathrm{KL}(p||q) \geq 0$ for any p and q,

$$-H[p] + \log_2 K \geq 0$$
$$H[p] \leq \log_2 K.$$

Thus, an arbitrary pmf p that is non-zero on D has at most the same entropy as the uniform distribution on D.

Consider an iid (discrete) dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, and a joint pmf $q(\cdot \mid \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$, such that $q(\mathbf{x} \mid \boldsymbol{\theta}) > 0$ for any $\mathbf{x}, \boldsymbol{\theta}$. The likelihood $q(\mathcal{D} \mid \boldsymbol{\theta})$ of $\mathcal{D}$ under $q(\cdot \mid \boldsymbol{\theta})$ is defined as

$$q(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^N q(\mathbf{x}_i \mid \boldsymbol{\theta}) = \prod_{\mathbf{x}} q(\mathbf{x} \mid \boldsymbol{\theta})^{N_{\mathbf{x}}},$$

where $N_{\mathbf{x}}$ is the number of occurrences of $\mathbf{x}$ in $\mathcal{D}$. Maximum likelihood estimation consists on finding a $\boldsymbol{\theta}$ such that $q(\mathcal{D} \mid \boldsymbol{\theta})$ is maximum. Because the logarithm is an increasing function (and $q(\mathbf{x} \mid \boldsymbol{\theta}) > 0$ for any $\mathbf{x}, \boldsymbol{\theta}$), this is equivalent to finding a $\boldsymbol{\theta}$ that maximizes the log-likelihood

$$\log_2 q(\mathcal{D} \mid \boldsymbol{\theta}) = \log_2 \prod_{\mathbf{x}} q(\mathbf{x} \mid \boldsymbol{\theta})^{N_{\mathbf{x}}} = \sum_{\mathbf{x}} \log_2 \left[q(\mathbf{x} \mid \boldsymbol{\theta})^{N_{\mathbf{x}}} \right] = \sum_{\mathbf{x}} N_{\mathbf{x}} \log_2 q(\mathbf{x} \mid \boldsymbol{\theta}).$$

Let p_{emp} denote the empirical distribution for $\mathcal{D}$. We now show that maximum likelihood estimation minimizes the Kullback-Leibler divergence between p_{emp} and $q(\cdot \mid \boldsymbol{\theta})$. By definition,

$$\mathrm{KL}(p_{\mathrm{emp}}||q(\cdot \mid \boldsymbol{\theta})) = \sum_{\mathbf{x}} p_{\mathrm{emp}}(\mathbf{x}) \log_2 \frac{p_{\mathrm{emp}}(\mathbf{x})}{q(\mathbf{x} \mid \boldsymbol{\theta})}$$

$$= \sum_{\mathbf{x}} p_{\mathrm{emp}}(\mathbf{x}) \log_2 p_{\mathrm{emp}}(\mathbf{x}) - \sum_{\mathbf{x}} p_{\mathrm{emp}}(\mathbf{x}) \log_2 q(\mathbf{x} \mid \boldsymbol{\theta})$$

$$= \sum_{\mathbf{x}} p_{\mathrm{emp}}(\mathbf{x}) \log_2 p_{\mathrm{emp}}(\mathbf{x}) - \frac{1}{N} \sum_{\mathbf{x}} N_{\mathbf{x}} \log_2 q(\mathbf{x} \mid \boldsymbol{\theta}),$$

where the summation only involves the $\mathbf{x}$ for which $p_{\mathrm{emp}}(\mathbf{x}) > 0$. Because the leftmost summation is a constant wrt $\boldsymbol{\theta}$, minimizing $\mathrm{KL}(p_{\mathrm{emp}}||q(\cdot \mid \boldsymbol{\theta}))$ corresponds to maximizing

$$\frac{1}{N} \sum_{\mathbf{x}} N_{\mathbf{x}} \log_2 q(\mathbf{x} \mid \boldsymbol{\theta})$$

wrt $\boldsymbol{\theta}$, which corresponds to maximum likelihood estimation (since $\frac{1}{N}$ is a positive constant), as we wanted to show.

Consider two discrete random variables X and Y. The conditional entropy $H[X \mid Y]$ of X given Y is defined as

$$H[X \mid Y] = -\sum_y p_Y(y) \sum_x p(x \mid y) \log_2 p(x \mid y) = \sum_y p_Y(y) H[X \mid Y = y],$$

where $0 \log_a 0$ is replaced by 0. It is possible to show that $H[X \mid Y] \leq H[X]$ for any X and Y.

The mutual information $I(X; Y)$ between two discrete random variables X and Y is defined as

$$I(X; Y) = \mathrm{KL}(p_{X,Y}||p_X p_Y) = \sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p_X(x) p_Y(y)},$$

where $p_X(x)$, $p_Y(y)$ and $p(x, y)$ are non-zero for all x, y. If any of them is zero, the corresponding term is conventionally replaced by 0. It is easy to show that

$$I(X; Y) = H[X] - H[X \mid Y] = H[Y] - H[Y \mid X].$$

Intuitively, the mutual information $I(X; Y)$ between X and Y measures the amount of information that Y has about X (dependence between X and Y). Clearly, $I(X; Y) = I(Y; X)$, and $I(X; Y) \geq 0$. It is also possible to show that $I(X; Y) = 0$ if and only if X and Y are independent.

2.4 Iterative minimization

Consider a real-valued function $f : \mathbb{R}^D \to \mathbb{R}$. The task of minimizing $f(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta} \in \mathbb{R}^D$ corresponds to finding a global minimum $\boldsymbol{\theta}^*$ such that $f(\boldsymbol{\theta}^*) = \min_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$.

A continuous function $f : \mathbb{R}^D \to \mathbb{R}$ is convex if, for any $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ in its domain,

$$f(\lambda \boldsymbol{\theta}_1 + (1 - \lambda)\boldsymbol{\theta}_2) \leq \lambda f(\boldsymbol{\theta}_1) + (1 - \lambda)f(\boldsymbol{\theta}_2),$$

for any $0 < \lambda < 1$. Intuitively, the value of the function f on a line segment between $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ is either on or below the line segment that connects $f(\boldsymbol{\theta}_1)$ to $f(\boldsymbol{\theta}_2)$, for any $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. The function f is strictly convex if the inequality above is strict. A function f is concave if $-f$ is convex. A strictly convex function has at most one local minimum, which would also be global minimum.

If a real-valued function f is defined only on a subset $A \subseteq \mathbb{R}^D$, the definition of convexity only applies if A is a convex set. The set A is convex if and only if for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in A$, we have $\lambda \boldsymbol{\theta}_1 + (1 - \lambda)\boldsymbol{\theta}_2 \in A$, for any $0 < \lambda < 1$.

A twice continuously differentiable function $f : \mathbb{R} \to \mathbb{R}$ is convex if and only if its second derivative is always non-negative. A twice continuously differentiable function $f : \mathbb{R}^D \to \mathbb{R}$ is convex if and only if its Hessian matrix $\nabla^2 f(\boldsymbol{\theta})$ at every $\boldsymbol{\theta}$ is positive semidefinite. A matrix $\nabla^2 f(\boldsymbol{\theta})$ is positive semidefinite if and only if $\mathbf{v}^T \nabla^2 f(\boldsymbol{\theta})\mathbf{v} \geq 0$, for any nonzero vector $\mathbf{v}$. The Hessian matrix $\nabla^2 f(\boldsymbol{\theta})$ at $\boldsymbol{\theta}$ of a twice continuously differentiable function f is given by

$$\nabla^2 f(\boldsymbol{\theta})_{i,j} = \frac{\partial^2}{\partial \theta_i \partial \theta_j} f(\boldsymbol{\theta}).$$

Convexity is a highly desirable property, since finding a (global) minimum generally becomes much easier.

Gradient descent is a simple procedure to minimize differentiable functions. This procedure starts at an arbitrary estimate $\boldsymbol{\theta}_0$, which is iteratively updated by the rule $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla f(\boldsymbol{\theta}_t)$, where $\eta_t > 0$ is the so-called learning rate at iteration t. Gradient descent is based on the fact that the vector $-\nabla f(\boldsymbol{\theta}_t)$ points to the direction of maximum local decrease in f. If f is strictly convex and the learning rates are sufficiently small, this procedure is guaranteed to converge to a global minimum (if such minimum exists).

Using a single learning rate for all iterations may be a very poor strategy. If the learning rate is too small, the procedure will converge prohibitively slowly. If the learning rate is too large, the procedure might diverge.

In gradient descent with exact line search, the learning rate η_t for iteration t minimizes

$$\phi_t(\eta) = f(\boldsymbol{\theta}_t - \eta \nabla f(\boldsymbol{\theta}_t))$$

with respect to η. Intuitively, this corresponds to restricting the minimization of f to the direction of the gradient at iteration t.

Suppose ϕ_t is differentiable with respect to η, and let $\phi_t(\eta) = f(\mathbf{z})$, such that $z_j = \theta_{t,j} - \eta \partial f(\boldsymbol{\theta}_t)/\partial \theta_j$. By the chain rule,

$$\phi_t'(\eta) = \sum_{j=1}^{D} \frac{\partial f(\mathbf{z})}{\partial z_j} \frac{\partial z_j}{\partial \eta} = - \sum_{j=1}^{D} \frac{\partial f(\mathbf{z})}{\partial z_j} \frac{\partial f(\boldsymbol{\theta}_t)}{\partial \theta_j} = -\nabla f(\boldsymbol{\theta}_t - \eta \nabla f(\boldsymbol{\theta}_t))\nabla f(\boldsymbol{\theta}_t).$$

Now suppose $\phi_t(\eta^*) = \min_\eta \phi_t(\eta)$, for some global minimum η^*. In that case, $\phi_t'(\eta^*) = 0$. Therefore, $\nabla f(\boldsymbol{\theta}_t - \eta^* \nabla f(\boldsymbol{\theta}_t))$ and $\nabla f(\boldsymbol{\theta}_t)$ must be orthogonal. In other words, the gradient at consecutive estimates obtained by gradient descent with exact line search must be orthogonal. Unfortunately, it is generally too expensive to solve the minimization problem posed by exact line search at each iteration of gradient descent.

A common heuristic employed in minimization by gradient descent is called the momentum (or heavy ball) method. This method updates the current estimate $\boldsymbol{\theta}_t$ by the rule $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla f(\boldsymbol{\theta}_t) + \mu_t(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$. Intuitively, the so-called momentum coefficient $\mu_t \geq 0$ at iteration t controls how much the previous update direction affects the

direction given by the gradient at $\boldsymbol{\theta}_t$. This method allows larger updates when the update direction is consistent over many iterations, an is typically combined with comparatively lower learning rates.

Recall that the linear approximation g_t to a differentiable function f at point $\boldsymbol{\theta}_t$ is given by

$$g_t(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_t) + \nabla f(\boldsymbol{\theta}_t)(\boldsymbol{\theta} - \boldsymbol{\theta}_t),$$

which corresponds to the equation of the plane tangent to f at $\boldsymbol{\theta}_t$ when $D = 2$.

Analogously, the quadratic approximation h_t to a twice differentiable function f at point $\boldsymbol{\theta}_t$ is given by

$$h_t(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_t) + \nabla f(\boldsymbol{\theta}_t)(\boldsymbol{\theta} - \boldsymbol{\theta}_t) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_t)^T \nabla^2 f(\boldsymbol{\theta}_t)(\boldsymbol{\theta} - \boldsymbol{\theta}_t),$$

which corresponds to the equation of a quadratic surface when $D = 2$.

Newton's method is an iterative minimization method based on finding the minimum $\boldsymbol{\theta}_{t+1}$ of the quadratic approximation h_t to f at the current estimate $\boldsymbol{\theta}_t$. Firstly, note that

$$\nabla h_t(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}\left[f(\boldsymbol{\theta}_t) + \nabla f(\boldsymbol{\theta}_t)(\boldsymbol{\theta} - \boldsymbol{\theta}_t) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_t)^T \nabla^2 f(\boldsymbol{\theta}_t)(\boldsymbol{\theta} - \boldsymbol{\theta}_t)\right] = \nabla f(\boldsymbol{\theta}_t) + \nabla^2 f(\boldsymbol{\theta}_t)(\boldsymbol{\theta} - \boldsymbol{\theta}_t),$$

since $\nabla_{\mathbf{a}}\left[\mathbf{a}^T \mathbf{A} \mathbf{a}\right] = (\mathbf{A} + \mathbf{A}^T)\mathbf{a} = 2\mathbf{A}\mathbf{a}$ for vector $\mathbf{a}$ and symmetric matrix A.

Therefore, if $\boldsymbol{\theta}_{t+1}$ is a local minimum of the quadratic approximation h_t to f at the current estimate $\boldsymbol{\theta}_t$, then

$$\nabla^2 f(\boldsymbol{\theta}_t)\boldsymbol{\theta}_{t+1} = -\nabla f(\boldsymbol{\theta}_t) + \nabla^2 f(\boldsymbol{\theta}_t)\boldsymbol{\theta}_t$$

$$\boldsymbol{\theta}_{t+1} = -\left[\nabla^2 f(\boldsymbol{\theta}_t)\right]^{-1}\nabla f(\boldsymbol{\theta}_t) + \left[\nabla^2 f(\boldsymbol{\theta}_t)\right]^{-1}\nabla^2 f(\boldsymbol{\theta}_t)\boldsymbol{\theta}_t = \boldsymbol{\theta}_t - \left[\nabla^2 f(\boldsymbol{\theta}_t)\right]^{-1}\nabla f(\boldsymbol{\theta}_t),$$

where we have assumed that $\nabla^2 f(\boldsymbol{\theta}_t)$ is invertible. Notice that if f is strictly convex, its Hessian matrix is always invertible, and the condition above is sufficient for $\boldsymbol{\theta}_{t+1}$ to be a global minimum of h_t.

More generally, the update rule used by Newton's method is typically given by

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta_t\left[\nabla^2 f(\boldsymbol{\theta}_{t-1})\right]^{-1}\nabla f(\boldsymbol{\theta}_{t-1}),$$

where $0 < \eta_t \le 1$ is the learning rate at iteration t, which is particularly useful when the Hessian is approximated.

In practice, computing the inverse Hessian matrix at each iteration of Newton's method may be too computationally expensive. BFGS (Broyden–Fletcher–Goldfarb–Shanno) and L-BFGS (limited-memory BFGS) are highly successful methods that approximate the inverse Hessian matrix $\left[\nabla^2 f(\boldsymbol{\theta}_t)\right]^{-1}$ at iteration t without employing the exact Hessian. Both methods only require a function f, its gradient function ∇f, and an initial estimate $\boldsymbol{\theta}_0$.

Consider the task of minimizing a differentiable function $f : \mathbb{R}^D \to \mathbb{R}$ given by $f(\boldsymbol{\theta}) = \sum_{i=1}^N f_i(\boldsymbol{\theta})$, where each $f_i : \mathbb{R}^D \to \mathbb{R}$ is also differentiable. Naturally, the gradient $\nabla f(\boldsymbol{\theta})$ of f at $\boldsymbol{\theta}$ is given by

$$\nabla f(\boldsymbol{\theta}) = \sum_{i=1}^N \nabla f_i(\boldsymbol{\theta}).$$

Mini-batch stochastic gradient descent randomly partitions the set $\{1, \ldots, N\}$ into sets of (almost) equal size called batches. A batch S is used to compute a surrogate for $\nabla f(\boldsymbol{\theta})$ given by

$$\sum_{i \in S} \nabla f_i(\boldsymbol{\theta}),$$

which is used in an update analogous to gradient descent. After each batch is considered in turn (one so-called epoch), another random partition into batches is employed. This technique is more computationally efficient than gradient descent when many individual gradients are expected to be similar.

3 Discrete generative models

Consider a sequence of binary random variables $X_1, \ldots, X_N$, each distributed according to $p(\cdot \mid \theta^*)$, for an unknown parameter $\theta^* \in [0, 1]$. Furthermore, suppose $X_i \perp\!\!\!\perp X_{-i} \mid \Theta$, for every i, and let $p(x \mid \theta) = \text{Ber}(x \mid \theta) = \theta^x(1 - \theta)^{1-x}$, for any $x \in \{0, 1\}$ and $\theta \in [0, 1]$. We say that a dataset $\mathcal{D} = x_1, \ldots, x_N$ is iid given Θ according to $p(\cdot \mid \theta^*)$.

The *likelihood* $p(\mathcal{D} \mid \theta)$ of the dataset $\mathcal{D} = x_1, \ldots, x_N$ under θ can be written as

$$p(\mathcal{D} \mid \theta) = p(x_1, \ldots, x_N \mid \theta) = \prod_{i=1}^{N} p(x_i \mid \theta) = \prod_{i=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i} = \theta^{N_1}(1-\theta)^{N_0},$$

where N_k is the number of occurences of k in $\mathcal{D}$.

In this context, $p(\theta)$ is the *prior* density for θ. Suppose $\Theta \sim \mathrm{Beta}(\cdot \mid a, b)$, for constant hyperparameters a and b, which encode *prior beliefs*. By definition,

$$p(\theta) = \mathrm{Beta}(\theta \mid a, b) = \frac{1}{B(a,b)} \theta^{a-1}(1-\theta)^{b-1}.$$

The *posterior* density $p(\theta \mid \mathcal{D})$ for θ given $\mathcal{D}$ is then given by

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})} = \frac{1}{p(\mathcal{D})} \frac{\theta^{N_1}(1-\theta)^{N_0}\theta^{a-1}(1-\theta)^{b-1}}{B(a,b)} = \frac{1}{p(\mathcal{D})} \frac{\theta^{N_1+a-1}(1-\theta)^{N_0+b-1}}{B(a,b)}.$$

The (unconditional) probability $p(\mathcal{D})$ of the dataset $\mathcal{D}$ can be obtained through marginalization:

$$p(\mathcal{D}) = \int_0^1 p(\mathcal{D}, \theta) \, d\theta = \int_0^1 p(\mathcal{D} \mid \theta)p(\theta) \, d\theta = \int_0^1 \frac{\theta^{N_1+a-1}(1-\theta)^{N_0+b-1}}{B(a,b)} \, d\theta = \frac{B(N_1 + a, N_0 + b)}{B(a,b)},$$

where the last equality comes from the definition of the Beta function B. Thus,

$$p(\theta \mid \mathcal{D}) = \frac{\theta^{N_1+a-1}(1-\theta)^{N_0+b-1}}{B(N_1 + a, N_0 + b)} = \mathrm{Beta}(\theta \mid N_1 + a, N_0 + b).$$

Notice that the prior and posterior densities are given by the same distribution under (possibly) distinct hyperparameters. For this reason, the Beta pdf is said to be a conjugate prior for the likelihood function of an iid Bernoulli dataset. Because the hyperparameters a and b are analogous to N_1 and N_0, they are also called pseudo-counts.

For any θ and $\mathcal{D}$, the counts N_0 and N_1 contain all the information about $\mathcal{D}$ that is needed to compute the posterior $p(\theta \mid \mathcal{D})$ given the prior $p(\theta)$. In the general case, if $p(\theta \mid \mathcal{D}) = p(\theta \mid \mathbf{S} = s(\mathcal{D}))$, for every θ and $\mathcal{D}$, then $s(\mathcal{D})$ is a *sufficient statistic* for $\mathcal{D}$.

Suppose the dataset $\mathcal{D}$ is partitioned into two datasets $\mathcal{D}'$ and $\mathcal{D}''$. By definition,

$$\begin{aligned}
p(\theta \mid \mathcal{D}', \mathcal{D}'') &= \frac{p(\mathcal{D}', \mathcal{D}'' \mid \theta)p(\theta)}{p(\mathcal{D}', \mathcal{D}'')} = \frac{p(\mathcal{D}' \mid \mathcal{D}'', \theta)p(\mathcal{D}'' \mid \theta)p(\theta)}{p(\mathcal{D}' \mid \mathcal{D}'')p(\mathcal{D}'')} = \frac{p(\mathcal{D}' \mid \theta)p(\theta \mid \mathcal{D}'')}{p(\mathcal{D}' \mid \mathcal{D}'')} \\
&= \frac{p(\mathcal{D}' \mid \theta)}{p(\mathcal{D}' \mid \mathcal{D}'')} \mathrm{Beta}(\theta \mid N_1'' + a, N_0'' + b) = \frac{\theta^{N_1'}(1-\theta)^{N_0'} \mathrm{Beta}(\theta \mid N_1'' + a, N_0'' + b)}{\int_0^1 p(\mathcal{D}' \mid \theta')p(\theta' \mid \mathcal{D}'') \, d\theta'} \\
&= \mathrm{Beta}(\theta \mid N_1' + N_1'' + a, N_0' + N_0'' + b) \\
&= p(\theta \mid \mathcal{D}),
\end{aligned}$$

where N_k' is the number of occurences of k in $\mathcal{D}'$ (analogously for N_k''). This result states that inference (computing the posterior density) can be performed incrementally (as more data becomes available). Furthermore, the posterior density $p(\theta \mid \mathcal{D}'')$ can be seen as a new prior density after $\mathcal{D}''$ is taken into account (the same is true for $\mathcal{D}'$).

It is possible to show that $\mathrm{Beta}(\cdot \mid N_1 + a, N_0 + b)$ has a mode (global maxima) at $\hat{\theta} = \arg\max_\theta p(\theta \mid \mathcal{D}) = \frac{N_1+a-1}{N+a+b-2}$ when $N_1 + a, N_0 + b > 1$. Therefore, $\hat{\theta}$ is a maximum a posteriori estimate. Similarly, the posterior mean $\mathbb{E}[\Theta \mid \mathcal{D}]$ is given by

$$\mathbb{E}[\Theta \mid \mathcal{D}] = \int_0^1 \theta p(\theta \mid \mathcal{D}) \, d\theta = \frac{N_1 + a}{N + a + b}.$$

The variance $\mathrm{var}[\Theta \mid \mathcal{D}]$ can be approximated by $\frac{\hat{\theta}(1-\hat{\theta})}{N}$ whenever $N \gg a, b$, where $\hat{\theta}$ denotes the maximum likelihood estimate. Notice that less data is required to achieve a low variance when $\hat{\theta}$ is either very close to 0 or 1.

In the case of a $\mathrm{Beta}(\cdot \mid 1, 1)$ prior, which is uniform on $[0, 1]$, the maximum a posteriori estimate $\hat{\theta}_{\mathrm{MAP}} = \frac{N_1}{N}$ is equivalent to the maximum likelihood estimate $\hat{\theta}_{\mathrm{MLE}} = \arg\max_\theta p(\mathcal{D} \mid \theta)$. This is derived easily from the equality

$$\hat{\theta}_{\mathrm{MAP}} = \arg\max_\theta p(\theta \mid \mathcal{D}) = \arg\max_\theta \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})} = \arg\max_\theta p(\mathcal{D} \mid \theta)p(\theta) = \arg\max_\theta \log p(\mathcal{D} \mid \theta) + \log p(\theta).$$

Considering the equality above, suppose $\mathcal{D}^+$ corresponds to the dataset $\mathcal{D}$ extended by a single element $\tilde{x}$. Clearly, $\log p(\mathcal{D}^+ \mid \theta) < \log p(\mathcal{D} \mid \theta)$ for any θ, except possibly for $\theta \in \{0, 1\}$. Thus, the term $\log p(\theta)$ loses importance as the amount of data tends to infinity, and the maximum a posteriori estimate converges to the maximum likelihood estimate.

The task of predicting the assignment to an additional random variable $\tilde{X} \sim p(\cdot \mid \theta^*)$ given $\mathcal{D}$ can be solved by computing the *posterior predictive* distribution for each $\tilde{x} \in \{0, 1\}$, which is given by

$$p(\tilde{x} \mid \mathcal{D}) = \int_0^1 p(\tilde{x}, \theta \mid \mathcal{D}) \, d\theta = \int_0^1 p(\tilde{x} \mid \theta, \mathcal{D}) p(\theta \mid \mathcal{D}) \, d\theta = \int_0^1 p(\tilde{x} \mid \theta) p(\theta \mid \mathcal{D}) \, d\theta.$$

For $\tilde{x} = 1$,

$$p(\tilde{x} \mid \mathcal{D}) = \int_0^1 \theta p(\theta \mid \mathcal{D}) \, d\theta = \mathbb{E}[\Theta \mid \mathcal{D}] = \frac{N_1 + a}{N + a + b}.$$

Thus, it is very easy to make predictions given the sufficient statistics and the prior hyperparameters. It is crucial to notice the distinction between $p(\tilde{x} \mid \mathcal{D})$ and $p(\tilde{x} \mid \hat{\theta}_{\mathrm{MAP}})$. Clearly, the first alternative *never* gives probability 1 to a particular $\tilde{x}$. This is highly desirable, since no finite amount of data would warrant such certainty.

Consider the related task of predicting the assignment to a random variable X that represents the *number of successes* in a sequence $\tilde{X}_1, \ldots, \tilde{X}_M$ of binary random variables, such that $\tilde{X}_i \sim p(\cdot \mid \theta^*)$ and $\tilde{X}_i \perp\!\!\!\perp \tilde{X}_{-i}, \boldsymbol{\mathcal{D}} \mid \Theta$, *after* seeing the dataset $\mathcal{D}$. The corresponding (beta-binomial) posterior predictive distribution is defined on $x \in \{0, \ldots, M\}$ as

$$\begin{aligned}
p(x \mid \mathcal{D}, M) &= \int_0^1 p(x, \theta \mid \mathcal{D}, M) \, d\theta = \int_0^1 p(x \mid \theta, M) p(\theta \mid \mathcal{D}) \, d\theta \\
&= \int_0^1 \mathrm{Bin}(x \mid M, \theta) p(\theta \mid \mathcal{D}) \, d\theta = \int_0^1 \mathrm{Bin}(x \mid M, \theta) \, \mathrm{Beta}(\theta \mid N_1 + a, N_0 + b) \, d\theta \\
&= \binom{M}{x} \frac{B(N_1 + a + x, N_0 + b + M - x)}{B(N_1 + a, N_0 + b)}.
\end{aligned}$$

Consider a sequence of discrete random variables $X_1, \ldots, X_N$ such that $X_i \sim p(\cdot \mid \boldsymbol{\theta}^*)$ and $X_i \perp\!\!\!\perp X_{-i} \mid \Theta$, for every i. In particular, let $p(x \mid \boldsymbol{\theta}) = \mathrm{Cat}(x \mid \boldsymbol{\theta}) = \theta_x$, for every $x \in \{1, \ldots, K\}$ and valid $\boldsymbol{\theta}$.

The likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of the dataset $\mathcal{D} = x_1, \ldots, x_N$ under $\boldsymbol{\theta}$ is defined as

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = p(x_1, \ldots, x_N \mid \boldsymbol{\theta}) = \prod_{i=1}^N p(x_i \mid \boldsymbol{\theta}) = \prod_{i=1}^N \theta_{x_i} = \prod_{k=1}^K \theta_k^{N_k},$$

where N_k is the number of occurrences of k in $\mathcal{D}$. A convenient (conjugate) prior is the Dirichlet joint pdf, which we already defined as

$$\mathrm{Dirichlet}(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k - 1},$$

for every $\boldsymbol{\theta}$ in the probability simplex S_K and hyperparameter vector $\boldsymbol{\alpha}$, whose elements are the pseudo-counts. The posterior is also Dirichlet:

$$\begin{aligned}
p(\boldsymbol{\theta} \mid \mathcal{D}) &= \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{1}{p(\mathcal{D}) B(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{N_k} \prod_{k=1}^K \theta_k^{\alpha_k - 1} = \frac{1}{p(\mathcal{D}) B(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{N_k + \alpha_k - 1} \\
&= \mathrm{Dirichlet}(\boldsymbol{\theta} \mid \boldsymbol{\alpha} + \mathbf{N}),
\end{aligned}$$

where $\mathbf{N} = (N_1, \ldots, N_K)$ is the vector of counts (a sufficient statistic). The last equality follows from the fact that $\int_{S_K} p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta} = 1$.

The posterior predictive for an additional random variable $\tilde{X} \sim p(\cdot \mid \boldsymbol{\theta}^*)$ given $\mathcal{D}$ is given, for $\tilde{x} = j$, by

$$p(\tilde{x} \mid \mathcal{D}) = \int_{S_K} p(\tilde{x}, \boldsymbol{\theta} \mid \mathcal{D}) \, d\boldsymbol{\theta} = \int_{S_K} p(\tilde{x} \mid \boldsymbol{\theta}, \mathcal{D}) p(\boldsymbol{\theta} \mid \mathcal{D}) \, d\boldsymbol{\theta} = \int_{S_K} \theta_j p(\boldsymbol{\theta} \mid \mathcal{D}) \, d\boldsymbol{\theta}$$

$$= \int_{S_K} \theta_j p(\theta_j, \boldsymbol{\theta}_{-j} \mid \mathcal{D}) \, d\boldsymbol{\theta} = \int_{S_K} \theta_j p(\theta_j, \boldsymbol{\theta}_{-j} \mid \mathcal{D}) \, d\boldsymbol{\theta}_{-j} d\theta_j$$

$$= \int_0^1 \theta_j \int_{S_{K,j,\theta_j}} p(\theta_j, \boldsymbol{\theta}_{-j} \mid \mathcal{D}) \, d\boldsymbol{\theta}_{-j} d\theta_j = \int_0^1 \theta_j p(\theta_j \mid \mathcal{D}) \, d\theta_j$$

$$= \mathbb{E}[\Theta_j \mid \mathcal{D}] = \frac{N_j + \alpha_j}{N + \alpha_0},$$

where $S_{K,j,\theta_j} = \{\boldsymbol{\theta}' \in [0,1]^{K-1} \mid \theta_j + \sum_i \theta_i' = 1\}$.

A maximum a posteriori estimate $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D})$ can be found using Lagrangian multipliers. Consider a differentiable function $f : \mathbb{R}^K \to \mathbb{R}$, and the task of finding the local minima of f restricted to the set $S = \{\mathbf{a} \in \mathbb{R}^K \mid \forall i, g_i(\mathbf{a}) = 0\}$, given differentiable real-valued functions $g_1, \ldots, g_c$. The Lagrange multiplier theorem states that, if $\mathbf{a}$ is a local minimum of f restricted to S, and the vectors $\nabla g_1(\mathbf{a}), \ldots, \nabla g_c(\mathbf{a})$ are linearly independent, then $\nabla f(\mathbf{a}) = \sum_{i=1}^c \lambda_i \nabla g_i(\mathbf{a})$, for some $\lambda_i \in \mathbb{R}$. The converse is not generally true.

Consider the task of maximizing

$$\log \text{Dirichlet}(\boldsymbol{\theta} \mid \boldsymbol{\alpha} + \mathbf{N}) = \sum_{k=1}^K \alpha_k \log \theta_k + \sum_{k=1}^K N_k \log \theta_k - \sum_{k=1}^K \log \theta_k - \log B(\boldsymbol{\alpha} + \mathbf{N})$$

with respect to $\boldsymbol{\theta}$, subject to $g(\boldsymbol{\theta}) = 1 - \sum_{k=1}^K \theta_k = 0$. This is equivalent to *minimizing*

$$\ell(\boldsymbol{\theta}) = -\sum_{k=1}^K \alpha_k \log \theta_k - \sum_{k=1}^K N_k \log \theta_k + \sum_{k=1}^K \log \theta_k$$

wrt $\boldsymbol{\theta}$, with the same restriction. Notice that $\nabla g(\boldsymbol{\theta}) = -\mathbf{1}$ is non-zero (linearly independent), for every $\boldsymbol{\theta}$. Therefore, if $\boldsymbol{\theta}$ is a local minimum of ℓ subject to $g(\boldsymbol{\theta}) = 0$,

$$\nabla \ell(\boldsymbol{\theta}) = \lambda \nabla g(\boldsymbol{\theta}),$$

for some $\lambda \in \mathbb{R}$, by the Lagrange multiplier theorem. Because

$$\frac{\partial \ell(\boldsymbol{\theta})}{\partial \theta_j} = \frac{N_j + \alpha_j - 1}{\theta_j}$$

for every $j \in \{1, \ldots, K\}$, a local minimum of ℓ subject to $g(\boldsymbol{\theta}) = 0$ obeys the system of equations

$$\theta_j = \frac{N_j + \alpha_j - 1}{\lambda},$$

for every $j \in \{1, \ldots, K\}$. The requirement $\sum_j \theta_j = 1$ gives $\lambda = N + \alpha_0 - K$. Thus, if a maximum a posteriori estimate $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D})$ is a local maximum of $p(\cdot \mid \mathcal{D})$ restricted to the probability simplex S_K, it follows that

$$\hat{\theta}_j = \frac{N_j + \alpha_j - 1}{N + \alpha_0 - K}.$$

When the prior is Dirichlet($\cdot \mid \mathbf{1}$), the maximum likelihood estimate is given by $\frac{N_j}{N}$, as expected.

Consider the task of creating a classifier given the dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ distributed according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for an unknown $\boldsymbol{\theta}^*$. Consider also that $\mathbf{x} \in \mathbb{R}^D$ and $y \in \{1, \ldots C\}$, for all $(\mathbf{x}, y) \in \mathcal{D}$. Furthermore, let $\mathbf{X}_i, Y_i \perp\!\!\!\perp \mathbf{X}_{-i}, Y_{-i} \mid \Theta$, for every $i \in \{1, \ldots, N\}$. We say that $\mathcal{D}$ is iid given Θ according to $p(\cdot \mid \boldsymbol{\theta}^*)$.

In general, a generative classifier is a probabilistic classifier that uses the fact that

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{x} \mid y, \boldsymbol{\theta}) p(y \mid \boldsymbol{\theta})}{p(\mathbf{x} \mid \boldsymbol{\theta})} \propto_y p(\mathbf{x} \mid y, \boldsymbol{\theta}) p(y \mid \boldsymbol{\theta}),$$

for all $\mathbf{x}, y$ and $\boldsymbol{\theta}$.

A *naive Bayes* classifier is a generative classifier that assumes

$$p(\mathbf{x}, y \mid \boldsymbol{\theta}) = p(y \mid \boldsymbol{\theta}) p(\mathbf{x} \mid y, \boldsymbol{\theta}) = p(y \mid \boldsymbol{\theta}) \prod_{j=1}^{D} p(x_j \mid y, \boldsymbol{\theta}),$$

for all $\mathbf{x}, y$ and $\boldsymbol{\theta}$. In other words, it assumes that each feature is independent of the others given the class and the parameters.

According to this model, the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of the dataset $\mathcal{D}$ under $\boldsymbol{\theta}$ can be written as

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i, y_i \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(y_i \mid \boldsymbol{\theta}) \prod_{j=1}^{D} p(x_{i,j} \mid y_i, \boldsymbol{\theta}).$$

For simplicity, the remainder of this section presupposes binary features, i.e., $\mathbf{x} \in \{0,1\}^D$, for all $(\mathbf{x}, y) \in \mathcal{D}$. Letting $\pi_y = p(y \mid \boldsymbol{\theta})$ and $p(x_j \mid y, \boldsymbol{\theta}) = \mathrm{Ber}(x_j \mid \theta_{j,y}) = \theta_{j,y}^{x_j}(1 - \theta_{j,y})^{1-x_j}$, the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of $\mathcal{D}$ given $\boldsymbol{\theta}$ can be written as

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} \pi_{y_i} \prod_{j=1}^{D} \theta_{j,y_i}^{x_{i,j}}(1 - \theta_{j,y_i})^{1-x_{i,j}} = \left[\prod_{y=1}^{C} \pi_y^{N_y} \right] \left[\prod_{y=1}^{C} \prod_{j=1}^{D} \theta_{j,y}^{N_{j,y}}(1 - \theta_{j,y})^{N_y - N_{j,y}} \right],$$

where N_y is the number of occurrences of class y in $\mathcal{D}$ and $N_{j,y}$ is the number of occurrences of feature j equal to 1 when the class is y.

We will assume that the prior density $p(\boldsymbol{\theta})$ can be written as

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\pi}) \prod_{j=1}^{D} \prod_{y=1}^{C} p(\theta_{j,y}) = \mathrm{Dirichlet}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) \prod_{j=1}^{D} \prod_{y=1}^{C} \mathrm{Beta}(\theta_{j,y} \mid \beta_0, \beta_1),$$

for every $\boldsymbol{\theta}$, where $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are the prior hyperparameters, and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_C)$.

Under these assumptions, it is possible to show that

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} = p(\boldsymbol{\pi} \mid \mathcal{D}) \prod_{j=1}^{D} \prod_{y=1}^{C} p(\theta_{j,y} \mid \mathcal{D}),$$

where

$$p(\boldsymbol{\pi} \mid \mathcal{D}) = \mathrm{Dirichlet}(\boldsymbol{\pi} \mid N_1 + \alpha_1, \ldots, N_C + \alpha_C)$$
$$p(\theta_{j,y} \mid \mathcal{D}) = \mathrm{Beta}(\theta_{j,y} \mid \beta_0 + (N_y - N_{j,y}), \beta_1 + N_{j,y}).$$

Thus, fitting a naive Bayes classifier to binary features simply requires the sufficient statistics N_y and $N_{j,y}$, for every y and j.

Classification can be formulated as follows. Consider an additional pair $\mathbf{X}, Y$ distributed according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for the same unknown $\boldsymbol{\theta}^*$ that generated $\mathcal{D}$. By definition,

$$p(y \mid \mathbf{x}, \mathcal{D}) = \frac{p(y, \mathbf{x} \mid \mathcal{D})}{p(\mathbf{x} \mid \mathcal{D})} = \frac{1}{p(\mathbf{x} \mid \mathcal{D})} \int_{\mathrm{Val}(\boldsymbol{\Theta})} p(y, \mathbf{x}, \boldsymbol{\theta} \mid \mathcal{D}) \, d\boldsymbol{\theta} = \frac{1}{p(\mathbf{x} \mid \mathcal{D})} \int_{\mathrm{Val}(\boldsymbol{\Theta})} p(y, \mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) \, d\boldsymbol{\theta}$$

$$= \frac{1}{p(\mathbf{x} \mid \mathcal{D})} \int_{\mathrm{Val}(\boldsymbol{\Theta})} \left[p(y \mid \boldsymbol{\theta}) \prod_{j=1}^{D} p(x_j \mid y, \boldsymbol{\theta}) \right] \left[p(\boldsymbol{\pi} \mid \mathcal{D}) \prod_{j=1}^{D} \prod_{y'=1}^{C} p(\theta_{j,y'} \mid \mathcal{D}) \right] d\boldsymbol{\theta}$$

$$\propto_y \left[\int_{\mathrm{Val}(\boldsymbol{\Pi})} \pi_y p(\boldsymbol{\pi} \mid \mathcal{D}) \, d\boldsymbol{\pi} \right] \left[\prod_{j=1}^{D} \int_0^1 \theta_{j,y}^{x_j}(1 - \theta_{j,y})^{1-x_j} p(\theta_{j,y} \mid \mathcal{D}) \, d\theta_{j,y} \right]$$

$$\propto_y \frac{N_y + \alpha_y}{N + \alpha_0} \prod_{j=1}^{D} \hat{\theta}_{j,y}^{x_j}(1 - \hat{\theta}_{j,y})^{1-x_j},$$

where

$$\hat{\theta}_{j,y} = \frac{N_{j,y} + \beta_1}{N_y + \beta_1 + \beta_0}.$$

Thus, $p(y \mid \mathbf{x}, \mathcal{D}) = cv_y$, for every y, where v_y is a variable that depends on y, and c is a constant with respect to y. Clearly, $\sum_y p(y \mid \mathbf{x}, \mathcal{D}) = \sum_y cv_y = c\sum_y v_y = 1$. Because $\sum_y v_y$ must be non-zero, $c = \frac{1}{\sum_y v_y}$. Thus, $p(y \mid \mathbf{x}, \mathcal{D}) = \frac{v_y}{\sum_{y'} v_{y'}}$, for every y. This procedure is called renormalization, and often avoids a significant computational challenge. Due to numerical issues, it is also safer to obtain $p(y \mid \mathbf{x}, \mathcal{D})$ through $\log p(y \mid \mathbf{x}, \mathcal{D})$, for every y. Naturally, $\log p(y \mid \mathbf{x}, \mathcal{D}) = \log cv_y = \log v_y + \log c = \log v_y - \log \sum_{y'} e^{\log v_{y'}}$. The term $\log \sum_{y'} e^{\log v_{y'}}$ can be computed more safely using the so-called log-sum-exp trick, which we omit.

Consider once again a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, which is iid given $\boldsymbol{\Theta}$ and distributed according to $p(\cdot \mid \boldsymbol{\theta}^*)$. Suppose the features are binary, and let $\mathbf{X}, Y$ be a pair distributed according to $p(\cdot \mid \boldsymbol{\theta}^*)$. It is possible to define the mutual information $I(X_j; Y \mid \mathcal{D})$ between X_j and Y given the dataset $\mathcal{D}$ as

$$I(X_j; Y \mid \mathcal{D}) = \sum_{x_j} \sum_y p(x_j, y \mid \mathcal{D}) \log \left[\frac{p(x_j, y \mid \mathcal{D})}{p(x_j \mid \mathcal{D}) p(y \mid \mathcal{D})} \right].$$

In the case of a naive Bayes model with binary features, it can be shown that

$$I(X_j; Y \mid \mathcal{D}) = \sum_y \hat{\theta}_{j,y} \frac{N_y + \alpha_y}{N + \alpha_0} \log \frac{\hat{\theta}_{j,y}}{\hat{\theta}_j} + \sum_y (1 - \hat{\theta}_{j,y}) \frac{N_y + \alpha_y}{N + \alpha_0} \log \frac{1 - \hat{\theta}_{j,y}}{1 - \hat{\theta}_j},$$

where $\hat{\theta}_j = \sum_y \hat{\theta}_{j,y} \frac{N_y + \alpha_y}{N + \alpha_0}$. Because the mutual information between X_j and Y can be interpreted as the reduction in uncertainty about Y when X_j is observed, it can be used to discard features that are individually uninformative about the class. However, this method does not take into account whether features are informative when considered in groups. The general task of identifying features that are important for classification is called feature selection.

4 Gaussian models

As already mentioned, a multivariate Gaussian joint pdf $\mathcal{N}(\cdot \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is defined on $\mathbb{R}^D$ as

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})},$$

where $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean vector and $\boldsymbol{\Sigma}$ is the $D \times D$ (positive definite) covariance matrix.

Because $\boldsymbol{\Sigma}$ is positive definite, it can be written as $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where each column j of the $D \times D$ matrix $\boldsymbol{U}$ is an eigenvector $\mathbf{u}_j$ of $\boldsymbol{\Sigma}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix such that $\Lambda_{j,j} = \lambda_j$ is the eigenvalue that corresponds to $\mathbf{u}_j$. Furthermore, $\boldsymbol{U}$ is chosen so that $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$, which means that $(\mathbf{u}_1, \ldots, \mathbf{u}_D)$ is an orthonormal basis for $\mathbb{R}^D$. Because $\boldsymbol{\Sigma}, \boldsymbol{U}$ and $\boldsymbol{\Lambda}$ must be invertible,

$$\boldsymbol{\Sigma}^{-1} = (\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T)^{-1} = \boldsymbol{U}^{-T}(\boldsymbol{U}\boldsymbol{\Lambda})^{-1} = \boldsymbol{U}^{-T}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^{-1} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T,$$

where the second and third equalities follow from a property of the inverse of a product of invertible matrices, and the last equality follows from $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$. Since $\boldsymbol{\Lambda}$ is invertible and diagonal, $\boldsymbol{\Lambda}^{-1} = \text{diag}(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_D})$, and the equation above can be rewritten as

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T = \sum_{i=1}^D \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T,$$

which leads to

$$(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = \sum_{i=1}^D \frac{1}{\lambda_i}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{u}_i \mathbf{u}_i^T (\mathbf{x}-\boldsymbol{\mu}) = \sum_{i=1}^D \frac{y_i^2}{\lambda_i},$$

where $y_i = (\mathbf{x} - \boldsymbol{\mu})\mathbf{u}_i$ is the projection of $\mathbf{x} - \boldsymbol{\mu}$ onto the unit vector $\mathbf{u}_i$.

For $D = 2$, it is important to notice that, given a constant c, the set $\{\mathbf{x} \in \mathbb{R}^2 \mid (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = c\}$ is an ellipse centered on $\boldsymbol{\mu}$. The axis of this ellipse are aligned with $\mathbf{u}_1$ and $\mathbf{u}_2$, and scaled according to $\sqrt{\lambda_1}$ and $\sqrt{\lambda_2}$ (recall that $\lambda_i \geq 0$, since $\boldsymbol{\Sigma}$ is positive-definite). This intuition generalizes for higher dimensions.

Consider a dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, which is iid given $\boldsymbol{\Theta}$ according to a multivariate Gaussian pdf $\mathcal{N}(\cdot \mid \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$, where each assignment to $\boldsymbol{\Theta}$ represents a particular parameterization of the multivariate Gaussian distribution.

The likelihood $p(\mathcal{D} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ of $\mathcal{D}$ under the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is given by

$$p(\mathcal{D} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) \right].$$

$$= \frac{1}{(2\pi)^{\frac{ND}{2}} |\boldsymbol{\Sigma}|^{\frac{N}{2}}} \exp\left[-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) \right].$$

Thus, the log-likelihood $\log p(\mathcal{D} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ of $\mathcal{D}$ under the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is given by

$$\log p(\mathcal{D} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) - \log\left[(2\pi)^{\frac{ND}{2}} |\boldsymbol{\Sigma}|^{\frac{N}{2}}\right]$$

$$= -\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) - \frac{ND}{2}\log 2\pi - \frac{N}{2}\log|\boldsymbol{\Sigma}|.$$

Consider the task of finding the (valid) $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that maximize the log-likelihood $\ell = \log p(\mathcal{D} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ of the dataset $\mathcal{D}$. For this purpose, we will find the partial derivatives of ℓ with respect to μ_j, for every j. By definition,

$$\frac{\partial \ell}{\partial \mu_j} = -\frac{1}{2}\sum_{i=1}^{N} \frac{\partial}{\partial \mu_j}\left[(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right].$$

By letting $\mathbf{y}_i = \mathbf{x}_i - \boldsymbol{\mu}$ and using the chain rule,

$$\frac{\partial \ell}{\partial \mu_j} = -\frac{1}{2}\sum_{i=1}^{N} \frac{\partial y_{i,j}}{\partial \mu_j} \frac{\partial}{\partial y_{i,j}}\left[\mathbf{y}_i^T \boldsymbol{\Sigma}^{-1} \mathbf{y}_i\right] = \frac{1}{2}\sum_{i=1}^{N} \frac{\partial}{\partial y_{i,j}}\left[\mathbf{y}_i^T \boldsymbol{\Sigma}^{-1} \mathbf{y}_i\right].$$

It can be shown that $\nabla_{\mathbf{a}}\left[\mathbf{a}^T \boldsymbol{A} \mathbf{a}\right] = (\boldsymbol{A} + \boldsymbol{A}^T)\mathbf{a}$ for any (valid) matrix $\boldsymbol{A}$ and vector $\mathbf{a}$. Therefore,

$$\frac{\partial \ell}{\partial \mu_j} = \frac{1}{2}\sum_{i=1}^{N} \frac{\partial}{\partial y_{i,j}}\left[\mathbf{y}_i^T \boldsymbol{\Sigma}^{-1} \mathbf{y}_i\right] = \frac{1}{2}\sum_{i=1}^{N}\left[(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-T})\mathbf{y}_i\right]_j$$

Because $\boldsymbol{\Sigma}$ must be positive-definite, $\boldsymbol{\Sigma}^{-1}$ must also be positive-definite, and thus symmetric. Therefore,

$$\frac{\partial \ell}{\partial \mu_j} = \frac{1}{2}\sum_{i=1}^{N}\left[(\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-T})\mathbf{y}_i\right]_j = \frac{1}{2}\sum_{i=1}^{N}\left[2\boldsymbol{\Sigma}^{-1}\mathbf{y}_i\right]_j = \frac{1}{2}\sum_{i=1}^{N} 2\sum_{k=1}^{D}\Sigma_{j,k}^{(-1)} y_{i,k} = \sum_{i=1}^{N}\sum_{k=1}^{D}\Sigma_{j,k}^{(-1)} y_{i,k}.$$

Thus, the gradient $\nabla_{\boldsymbol{\mu}}\ell$ of ℓ wrt $\boldsymbol{\mu}$ is given by

$$\nabla_{\boldsymbol{\mu}}\ell = \sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1}\mathbf{y}_i = \sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-1}\left[\sum_{i=1}^{N}\mathbf{x}_i - \sum_{i=1}^{N}\boldsymbol{\mu}\right] = \boldsymbol{\Sigma}^{-1}\left[-N\boldsymbol{\mu} + \sum_{i=1}^{N}\mathbf{x}_i\right].$$

If $\hat{\boldsymbol{\mu}}$ is a local maximum of ℓ, then $\nabla_{\boldsymbol{\mu}}\ell = \mathbf{0}$ at $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}$. Therefore, a local maximum $\hat{\boldsymbol{\mu}}$ of ℓ can be written as

$$\boldsymbol{\Sigma}^{-1}\left[-N\hat{\boldsymbol{\mu}} + \sum_{i=1}^{N}\mathbf{x}_i\right] = \mathbf{0} \implies -N\hat{\boldsymbol{\mu}} + \sum_{i=1}^{N}\mathbf{x}_i = \mathbf{0} \implies N\hat{\boldsymbol{\mu}} = \sum_{i=1}^{N}\mathbf{x}_i \implies \hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i.$$

The condition above is also sufficient for $\hat{\boldsymbol{\mu}}$ to be a local maximum of ℓ. In simple terms, the maximum likelihood estimate for the mean of a multivariate Gaussian corresponds to the empirical mean.

We now find the partial derivatives of ℓ with respect to $\Sigma_{k,l}^{(-1)}$, for any k and l. From the definition of ℓ,

$$\frac{\partial \ell}{\partial \Sigma_{k,l}^{(-1)}} = -\frac{1}{2}\sum_{i=1}^{N} \frac{\partial}{\partial \Sigma_{k,l}^{(-1)}}\left[(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right] - \frac{N}{2}\frac{\partial}{\partial \Sigma_{k,l}^{(-1)}}\left[\log|\boldsymbol{\Sigma}|\right].$$

Firstly, notice that $\log |\boldsymbol{\Sigma}| = \log \frac{1}{|\boldsymbol{\Sigma}^{-1}|} = -\log |\boldsymbol{\Sigma}^{-1}|$ by a property of the determinant of an invertible matrix $\boldsymbol{\Sigma}$. Also, $\nabla_{\boldsymbol{A}} \log |\boldsymbol{A}| = \boldsymbol{A}^{-T}$, for any invertible matrix $\boldsymbol{A}$. Therefore, because $\boldsymbol{\Sigma}^{-1}$ is invertible and symmetric,

$$\frac{\partial \ell}{\partial \Sigma_{k,l}^{(-1)}} = -\frac{1}{2} \sum_{i=1}^{N} \frac{\partial}{\partial \Sigma_{k,l}^{(-1)}} \left[(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \right] + \frac{N}{2} \Sigma_{k,l}.$$

The next steps involve properties of the trace of a matrix. By definition, $\mathrm{tr}(\boldsymbol{A}) = \sum_i A_{i,i}$ for any matrix $\boldsymbol{A}$. It is particularly important to note that $\mathrm{tr}(\boldsymbol{ABC}) = \mathrm{tr}(\boldsymbol{CAB}) = \mathrm{tr}(\boldsymbol{BCA})$, for any matrices $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$, whenever the multiplications are valid. Also, $tr(a) = a$ for any scalar a represented as a 1×1 matrix. Using the equality $(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) = \mathrm{tr}((\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}))$ and reordering the multiplications,

$$\frac{\partial \ell}{\partial \Sigma_{k,l}^{(-1)}} = -\frac{1}{2} \sum_{i=1}^{N} \frac{\partial}{\partial \Sigma_{k,l}^{(-1)}} \left[\mathrm{tr}((\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}) \right] + \frac{N}{2} \Sigma_{k,l}.$$

Because $\nabla_A \mathrm{tr}(\boldsymbol{B}A) = \boldsymbol{B}^T$ and $(\boldsymbol{A}\boldsymbol{A}^T)^T = \boldsymbol{A}\boldsymbol{A}^T$, for any matrices $\boldsymbol{A}$ and $\boldsymbol{B}$,

$$\frac{\partial \ell}{\partial \Sigma_{k,l}^{(-1)}} = -\frac{1}{2} \sum_{i=1}^{N} \left[(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \right]_{k,l} + \frac{N}{2} \Sigma_{k,l}.$$

Therefore, the gradient $\nabla_{\boldsymbol{\Sigma}^{-1}} \ell$ of ℓ with respect to $\boldsymbol{\Sigma}^{-1}$ can be written as

$$\nabla_{\boldsymbol{\Sigma}^{-1}} \ell = -\frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T + \frac{N}{2} \boldsymbol{\Sigma}.$$

If $\hat{\boldsymbol{\Sigma}}^{-1}$ is a maximum of ℓ, then $\nabla_{\boldsymbol{\Sigma}^{-1}} \ell = \mathbf{0}$ when $\boldsymbol{\Sigma}^{-1} = \hat{\boldsymbol{\Sigma}}^{-1}$ and $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}$. Therefore, a maximum $\hat{\boldsymbol{\Sigma}}^{-1}$ of ℓ has an inverse $\hat{\boldsymbol{\Sigma}}$ that is given by

$$\nabla_{\boldsymbol{\Sigma}^{-1}} \ell = \mathbf{0} \implies -\frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T + \frac{N}{2} \hat{\boldsymbol{\Sigma}} = \mathbf{0} \implies \frac{N}{2} \hat{\boldsymbol{\Sigma}} = \frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$$

$$\implies \hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T.$$

The condition above is also sufficient for $\hat{\boldsymbol{\Sigma}}^{-1}$ to be a maximum of ℓ. In simple terms, the maximum likelihood estimate for the covariance matrix is the empirical covariance.

Consider the task of creating a classifier given the dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, which is iid given $\boldsymbol{\Theta}$ according to $p(\cdot \mid \boldsymbol{\theta}^*)$. Consider also that $\mathbf{x} \in \mathbb{R}^D$ and $y \in \{1, \ldots C\}$, for all $(\mathbf{x}, y) \in \mathcal{D}$. Gaussian discriminant analysis assumes that $p(\mathbf{x} \mid y, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ for every $\mathbf{x}, y$ and $\boldsymbol{\theta}$, where $\boldsymbol{\mu}_y$ and $\boldsymbol{\Sigma}_y$ are represented in $\boldsymbol{\theta}$. In words, it assumes that each class is distributed according to a distinct multivariate Gaussian distribution. For a particular estimate $\hat{\boldsymbol{\theta}}$ of the parameters (maximum likelihood, for instance), classification uses the fact that

$$p(y \mid \mathbf{x}, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{x} \mid y, \hat{\boldsymbol{\theta}}) p(y \mid \hat{\boldsymbol{\theta}})}{p(\mathbf{x} \mid \hat{\boldsymbol{\theta}})} \propto_y p(\mathbf{x} \mid y, \hat{\boldsymbol{\theta}}) p(y \mid \hat{\boldsymbol{\theta}}) = \mathcal{N}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_y, \hat{\boldsymbol{\Sigma}}_y) \hat{\pi}_y,$$

for all $\mathbf{x}$ and y.

In this context, a decision boundary (for a particular $\boldsymbol{\theta}$) is the set of $\mathbf{x}$ such that

$$\max_y p(y \mid \mathbf{x}, \boldsymbol{\theta}) = p(y' \mid \mathbf{x}, \boldsymbol{\theta}) = p(y'' \mid \mathbf{x}, \boldsymbol{\theta}),$$

for some $y' \neq y''$. In other words, the decision boundary is the set of observations for which the classifier is undecided between at least two classes. In a binary classification task, the decision boundary given by Gaussian discriminant analysis is a quadratic surface, which originates the term quadratic discriminant analysis.

Consider the task of finding the maximum likelihood estimate for Gaussian discriminant analysis. By definition,

$$\log p(\mathcal{D} \mid \boldsymbol{\theta}) = \log \prod_{i=1}^{N} p(\mathbf{x}_i \mid y_i, \boldsymbol{\theta}) p(y_i \mid \boldsymbol{\theta}) = \sum_{y=1}^{C} \sum_{i \mid y_i = y} \log \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) + \sum_{y=1}^{C} N_y \log \pi_y.$$

Because the first summation can be maximized (wrt the mean vectors and covariance matrices) independently for each class, it should not be surprising that the maximum likelihood estimates are given by

$$\hat{\boldsymbol{\mu}}_y = \frac{1}{N_y} \sum_{i \mid y_i = y} \mathbf{x}_i$$

$$\hat{\boldsymbol{\Sigma}}_y = \frac{1}{N_y} \sum_{i \mid y_i = y} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_y)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_y)^T$$

$$\hat{\pi}_y = \frac{N_y}{N}.$$

However, if there are insufficient observations in class y, $\hat{\boldsymbol{\Sigma}}_y$ may be singular (non-invertible, and thus invalid as a covariance matrix). Even if $\hat{\boldsymbol{\Sigma}}_y$ is not singular, it may overfit the data. We discuss some techniques that avoid this issue by making additional assumptions about the data.

Linear discriminant analysis additionally assumes that the covariance matrix is the same for all classes. Thus, for a particular estimate $\hat{\boldsymbol{\theta}}$,

$$p(y \mid \mathbf{x}, \hat{\boldsymbol{\theta}}) \propto_y \mathcal{N}(\mathbf{x} \mid \hat{\boldsymbol{\mu}}_y, \hat{\boldsymbol{\Sigma}}) \hat{\pi}_y = \hat{\pi}_y \frac{1}{\sqrt{(2\pi)^D |\hat{\boldsymbol{\Sigma}}|}} e^{-\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_y)^T \hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_y)} \propto_y \hat{\pi}_y e^{-\frac{1}{2}(\mathbf{x} - \hat{\boldsymbol{\mu}}_y)^T \hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_y)},$$

for all $\mathbf{x}$ and y. Using the distributivity of matrix multiplication over addition,

$$p(y \mid \mathbf{x}, \hat{\boldsymbol{\theta}}) \propto_y \hat{\pi}_y \exp\left[\hat{\boldsymbol{\mu}}_y^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{x} - \frac{1}{2}\hat{\boldsymbol{\mu}}_y^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_y - \frac{1}{2}\mathbf{x}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{x}\right]$$

$$= e^{\log \hat{\pi}_y} \exp\left[\hat{\boldsymbol{\mu}}_y^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{x} - \frac{1}{2}\hat{\boldsymbol{\mu}}_y^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_y\right] \exp\left[-\frac{1}{2}\mathbf{x}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{x}\right]$$

$$\propto_y \exp\left[\hat{\boldsymbol{\mu}}_y^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{x} - \frac{1}{2}\hat{\boldsymbol{\mu}}_y^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_y + \log \hat{\pi}_y\right].$$

Letting $\boldsymbol{\beta}_y = \hat{\boldsymbol{\mu}}_y^T \hat{\boldsymbol{\Sigma}}^{-1}$ and $\gamma_y = -\frac{1}{2}\hat{\boldsymbol{\mu}}_y^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_y + \log \hat{\pi}_y$,

$$p(y \mid \mathbf{x}, \hat{\boldsymbol{\theta}}) = \frac{e^{\boldsymbol{\beta}_y \mathbf{x} + \gamma_y}}{\sum_{y'=1}^{C} e^{\boldsymbol{\beta}_{y'} \mathbf{x} + \gamma_{y'}}}.$$

The function $\mathcal{S}$ defined by $\mathcal{S}(\boldsymbol{\eta}) = \left(\frac{e^{\eta_1}}{\sum_{j=1}^{C} e^{\eta_j}}, \ldots, \frac{e^{\eta_C}}{\sum_{j=1}^{C} e^{\eta_j}}\right)$ is also called softmax.

Consider the set S composed of all $\mathbf{x}$ such that $p(y \mid \mathbf{x}, \hat{\boldsymbol{\theta}}) = p(y' \mid \mathbf{x}, \hat{\boldsymbol{\theta}})$, for a particular choice of $y \neq y'$ and $\hat{\boldsymbol{\theta}}$. Clearly, $e^{\boldsymbol{\beta}_y \mathbf{x} + \gamma_y} = e^{\boldsymbol{\beta}_{y'} \mathbf{x} + \gamma_{y'}}$, and $\boldsymbol{\beta}_y \mathbf{x} + \gamma_y = \boldsymbol{\beta}_{y'} \mathbf{x} + \gamma_{y'}$. Rearranging the terms, $(\boldsymbol{\beta}_y - \boldsymbol{\beta}_{y'}) \mathbf{x} = \gamma_{y'} - \gamma_y$. Thus, the set S is an affine hyperplane (a set of vectors whose inner product by a particular vector is a constant), which originates the term linear discriminant analysis.

In diagonal linear discriminant analysis, the covariance matrix is additionally assumed to be diagonal. Implicitly, this corresponds to assuming that each feature is independent of the others given the class and the parameters (as in Naive Bayes models). In this case, appropriate estimates are given by $\hat{\boldsymbol{\mu}}_y = \frac{1}{N_y} \sum_{i \mid y_i = y} \mathbf{x}_i$ and $\hat{\boldsymbol{\Sigma}} = \text{diag}(\frac{1}{\hat{\sigma}_1^2}, \ldots, \frac{1}{\hat{\sigma}_D^2})$, where

$$\hat{\sigma}_j^2 = \frac{\sum_i (x_{i,j} - \hat{\mu}_{y_i, j})^2}{N - C}.$$

It is important to mention that these (unbiased) estimates do not achieve maximum likelihood.

Let $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ for some $\mathbf{x} \in \mathbb{R}^D$ given $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and suppose that $\mathbf{x}$ can be split into two vectors $\mathbf{x}_1 \in \mathbb{R}^{D_1}$ and $\mathbf{x}_2 \in \mathbb{R}^{D_2}$, which we denote by $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$. Analogously, let $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ and $\boldsymbol{\Sigma} = \left(\begin{smallmatrix} \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2} \end{smallmatrix} \right)$, where each $\boldsymbol{\Sigma}_{r,c}$ is a $D_r \times D_c$ matrix. It can be shown that $p(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1 \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{1,1})$ and $p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2 \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{2,2})$. Furthermore, $p(\mathbf{x}_1 \mid \mathbf{x}_2) = \mathcal{N}(\mathbf{x}_1 \mid \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2})$, where $\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{1,2} \boldsymbol{\Sigma}_{2,2}^{-1} (\mathbf{x}_2 - \boldsymbol{\mu}_2)$ and $\boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,2} \boldsymbol{\Sigma}_{2,2}^{-1} \boldsymbol{\Sigma}_{2,1}$. In summary, both conditioning and marginalization result in multivariate Gaussian distributions.

Suppose $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ and $p(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}(\mathbf{y} \mid \boldsymbol{A}\mathbf{x} + \mathbf{b}, \boldsymbol{\Sigma}_y)$, for all $\mathbf{x} \in \mathbb{R}^{D_x}$ and $\mathbf{y} \in \mathbb{R}^{D_y}$, for fixed $\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x, \boldsymbol{\Sigma}_y$, $D_y \times D_x$ matrix $\boldsymbol{A}$ and real vector $\mathbf{b} \in \mathbb{R}^{D_y}$. Such a random process is called a linear Gaussian system. It can be shown that

$$p(\mathbf{x} \mid \mathbf{y}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y}),$$

for all $\mathbf{x}$ and $\mathbf{y}$, where $\boldsymbol{\Sigma}_{x|y}^{-1} = \boldsymbol{\Sigma}_x^{-1} + \boldsymbol{A}^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{A}$, and $\boldsymbol{\mu}_{x|y} = \boldsymbol{\Sigma}_{x|y} \left[\boldsymbol{A}^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{y} - \mathbf{b}) + \boldsymbol{\Sigma}_x^{-1} \boldsymbol{\mu}_x \right]$. Also, $p(\mathbf{y}) = \mathcal{N}(\mathbf{y} \mid \boldsymbol{A}\boldsymbol{\mu}_x + \mathbf{b}, \boldsymbol{\Sigma}_y + \boldsymbol{A}\boldsymbol{\Sigma}_x \boldsymbol{A}^T)$, for every $\mathbf{y}$.

As an example, suppose $p(x) = \mathcal{N}(x \mid \mu_0, \lambda_0^{-1})$ for all $x \in \mathbb{R}$, where μ_0 is the prior mean and $\lambda_0 = 1/\sigma_0^2$ is the prior precision. Furthermore, suppose that $p(y_i \mid x) = \mathcal{N}(y_i \mid x, \lambda_y^{-1})$ for a given precision λ_y^{-1}, for all $i \in \{1, \dots N\}$, $y_i \in \mathbb{R}$, and $x \in \mathbb{R}$. This random process can be modeled as a linear Gaussian system by letting

$$p(\mathbf{y} \mid x) = \mathcal{N}(\mathbf{y} \mid \boldsymbol{A}x + \mathbf{b}, \boldsymbol{\Sigma}_y) = \mathcal{N}(\mathbf{y} \mid \mathbf{1}x, \lambda_y^{-1}\mathbf{I}).$$

Using the properties of a linear Gaussian system,

$$p(x \mid \mathbf{y}) = \mathcal{N}(x \mid \mu_N, \lambda_N^{-1}),$$

for all x and y, where $\lambda_N = \lambda_0 + N\lambda_y$ and

$$\mu_N = \frac{N\lambda_y}{\lambda_0 + N\lambda_y} \left[\frac{1}{N} \sum_{i=1}^N y_i \right] + \frac{\lambda_0}{\lambda_0 + N\lambda_y} \mu_0.$$

In this example, notice how the posterior mean is a convex combination of the prior mean and the empirical mean, weighted according to their precisions. Notice also that the posterior given N observations with precision λ_y is equivalent to a posterior given a single observation with precision $N\lambda_y$. Furthermore, inference can be performed incrementally, i.e., the posterior given each observation can become the prior for the next observations.

As a generalization of the previous example, let $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, for every $\mathbf{x} \in \mathbb{R}^D$, where $\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0$ are fixed. Also, let $p(\mathbf{y}_i \mid \mathbf{x}) = \mathcal{N}(\mathbf{y}_i \mid \mathbf{x}, \boldsymbol{\Sigma}_y)$, for all $i \in \{1, \dots, N\}$, $\mathbf{y}_i \in \mathbb{R}^D$, $\mathbf{x} \in \mathbb{R}^D$, and a fixed $\boldsymbol{\Sigma}_y$. It can be shown that $p(\mathbf{x} \mid \mathbf{y}_1, \dots, \mathbf{y}_N) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$, where $\boldsymbol{\Sigma}_N^{-1} = \boldsymbol{\Sigma}_0^{-1} + N\boldsymbol{\Sigma}_y^{-1}$ and $\boldsymbol{\mu}_N = \boldsymbol{\Sigma}_N \left[\boldsymbol{\Sigma}_y^{-1} \left[\sum_{i=1}^N \mathbf{y}_i \right] + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 \right]$.

Consider the task of computing the posterior $p(\boldsymbol{\mu} \mid \mathcal{D}, \boldsymbol{\Sigma})$ for some $\boldsymbol{\mu}$ given a fixed $\boldsymbol{\Sigma}$ and the dataset $\mathcal{D} = \mathbf{x}_1, \dots, \mathbf{x}_N$, which is iid given $\boldsymbol{\Theta}$ according to $\mathcal{N}(\cdot \mid \boldsymbol{\mu}^*, \boldsymbol{\Sigma})$, for an unknown $\boldsymbol{\mu}^*$. Using an appropriate (conjugate) Gaussian prior defined by $p(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu} \mid \mathbf{m}_0, \boldsymbol{V}_0)$ for all $\boldsymbol{\mu}$, it is possible to show that the desired posterior is given by $p(\boldsymbol{\mu} \mid \mathcal{D}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu} \mid \mathbf{m}_N, \boldsymbol{V}_N)$, where $\boldsymbol{V}_N^{-1} = \boldsymbol{V}_0^{-1} + N\boldsymbol{\Sigma}^{-1}$ and $\mathbf{m}_N = \boldsymbol{V}_N \left[\boldsymbol{\Sigma}^{-1} \left[\sum_{i=1}^N \mathbf{x}_i \right] + \boldsymbol{V}_0^{-1} \mathbf{m}_0 \right]$. Notice that the uncertainty about the parameter $\boldsymbol{\mu}$ is being modeled in a manner analogous to the previous example.

5 Bayesian statistics

Several models presented so far represent uncertainty about unknown parameters using (prior) probability distributions. This allows computing (posterior) probability distributions that represent the uncertainty about the unknown parameters after some data is observed. Models with these characteristics are studied by Bayesian statistics.

More concretely, in Bayesian models, the prior density $p(\boldsymbol{\theta})$ of the parameters $\boldsymbol{\theta}$ can be used to compute the posterior density $p(\boldsymbol{\theta} \mid \mathcal{D})$ of the parameters $\boldsymbol{\theta}$ given the dataset $\mathcal{D}$ using

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})},$$

where $p(\mathcal{D} \mid \boldsymbol{\theta})$ is the likelihood of dataset $\mathcal{D}$ under the parameters $\boldsymbol{\theta}$, and $p(\mathcal{D}) = \int_{\text{Val}(\boldsymbol{\Theta})} p(\mathcal{D} \mid \boldsymbol{\theta}')p(\boldsymbol{\theta}') \, d\boldsymbol{\theta}'$ is the marginal likelihood of $\mathcal{D}$.

A point estimate $\hat{\boldsymbol{\theta}}$ is an element in the domain of a posterior distribution that can be used (as an attempt) to summarize it. Common point estimates are the posterior mode (maximum a posteriori estimate), posterior mean, and posterior median. The obvious drawback of summarizing a posterior distribution by a point estimate is that the information about uncertainty is lost.

Although maximum a posteriori estimates are often useful for being easy to compute, they have several issues. In particular, maximum a posteriori estimates are not invariant to reparameterization. Concretely, consider a continuous random variable $X \sim p_X$. If $Y = f(X)$ for an invertible function $f : \mathbb{R} \to \mathbb{R}$, $g = f^{-1}$, and g is differentiable, then $p_Y(y) = p_X(g(y))|g'(y)|$. Now consider the maximum a posteriori estimates $\hat{x} = \arg\max_{x'} p_X(x')$ and $\hat{y} = \arg\max_{y'} p_Y(y')$. It is not generally true that $\hat{x} = g(\hat{y})$. Consequently, a maximum a posteriori estimate $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}'} p(\boldsymbol{\theta}' \mid \mathcal{D})$ is dependent on how the parameters are represented.

A $100(1-\alpha)\%$ credible interval for a random variable Θ distributed according to a (posterior) probability density function $p(\cdot \mid \mathcal{D})$ is an interval (l, u) such that

$$P(l < \Theta < u \mid \mathcal{D}) = \int_l^u p(\theta \mid \mathcal{D}) \, d\theta = 1 - \alpha.$$

Intuitively, if α is small and (l, u) is a $100(1 - \alpha)\%$ credible interval for Θ, there is a high probability that Θ is in the interval (l, u) according to the posterior distribution. This is an alternative way to summarize a posterior distribution, which preserves more information about uncertainty than point estimates.

The $100(1 - \alpha)\%$ *central* credible interval for a random variable Θ distributed according to a (posterior) probability density function $p(\cdot \mid \mathcal{D})$ is an interval (l, u) such that

$$\int_{-\infty}^l p(\theta \mid \mathcal{D}) \, d\theta = \int_u^{+\infty} p(\theta \mid \mathcal{D}) \, d\theta = \frac{\alpha}{2}.$$

For instance, the 95% central credible interval for a random variable $\Theta \sim \mathcal{N}(\cdot \mid \mu, \sigma^2)$ is approximately $(\mu - 1.96\sigma, \mu + 1.96\sigma)$. In the general case, central credible intervals can be found by Monte Carlo approximations of quantiles. Credible intervals should not be confused with confidence intervals.

Consider two random variables Θ_1 and Θ_2, which represent the unknown rates of success of two independent processes. Comparing such rates is a common task in statistics. Suppose the dataset $\mathcal{D}$ contains exactly y_i successes in N_i independent trials of process i, for $i = 1$ and $i = 2$. Using a Bayesian approach, we could use $\text{Beta}(\cdot \mid a, b)$ as a prior distribution for both Θ_1 and Θ_2. Therefore, the posterior probability that $\Theta_1 > \Theta_2$ would be given by

$$\begin{aligned}
P(\Theta_1 > \Theta_2 \mid \mathcal{D}) &= \int_R p(\theta_1, \theta_2 \mid \mathcal{D}) \, d\theta_1 d\theta_2 \\
&= \int_R p(\theta_1 \mid \mathcal{D}) p(\theta_2 \mid \mathcal{D}) \, d\theta_1 d\theta_2 \\
&= \int_R \text{Beta}(\theta_1 \mid a + y_1, N_1 - y_1 + b) \, \text{Beta}(\theta_2 \mid a + y_2, N_2 - y_2 + b) \, d\theta_1 d\theta_2,
\end{aligned}$$

where the second equality follows from the assumption of independence, and $R = \{(\theta_1, \theta_2) \in [0, 1]^2 \mid \theta_1 > \theta_2\}$. Such an integral can be approximated by Monte Carlo methods.

Usually, there are many ways to model a problem in statistical inference. For instance, different models can make different independence assumptions between random variables, which can lead to different numbers of free parameters. In Bayesian statistics, uncertainty about a model m is represented by a prior probability $p(m)$. Thus, it becomes possible to compute the posterior probability $p(m \mid \mathcal{D})$ of m given a dataset $\mathcal{D}$ by

$$p(m \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid m) p(m)}{\sum_{m'} p(m', \mathcal{D})}.$$

In this context, the marginal likelihood $p(\mathcal{D} \mid m)$ of the dataset $\mathcal{D}$ under the model m is given by

$$p(\mathcal{D} \mid m) = \int_{\text{Val}(\boldsymbol{\Theta})} p(\mathcal{D}, \boldsymbol{\theta} \mid m) \, d\boldsymbol{\theta} = \int_{\text{Val}(\boldsymbol{\Theta})} p(\mathcal{D} \mid \boldsymbol{\theta}, m) p(\boldsymbol{\theta} \mid m) \, d\boldsymbol{\theta},$$

where $\text{Val}(\boldsymbol{\Theta})$ is the set of admissible parameterizations of the model m. Notice that although a complex model may have particular parameterizations that have higher likelihood than a simpler model, its marginal likelihood is not necessarily higher. Intuitively, a complex model usually also has many parameterizations with low likelihood, which bring the marginal likelihood down. This phenomenon is called conservation of probability mass, and is highly related to the concepts of overfitting and Occam's razor.

Consider a prior density function p for a random vector of parameters $\boldsymbol{\Theta}$ defined by $p(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{Z_0}$ for every $\boldsymbol{\theta}$, where $q(\boldsymbol{\theta})$ is an unnormalized positive density (i.e., q is a positive real function of $\boldsymbol{\theta}$), and $Z_0 > 0$. Furthermore,

suppose the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of any dataset $\mathcal{D}$ under any $\boldsymbol{\theta}$ is defined by $p(\mathcal{D} \mid \boldsymbol{\theta}) = \frac{q(\mathcal{D}|\boldsymbol{\theta})}{Z_l}$, where $q(\mathcal{D} \mid \boldsymbol{\theta})$ is an unnormalized positive density, and $Z_l > 0$. Finally, suppose $p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{q(\mathcal{D}|\boldsymbol{\theta})q(\boldsymbol{\theta})}{Z_N}$, where $Z_N > 0$. By definition,

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}$$

$$\frac{q(\mathcal{D} \mid \boldsymbol{\theta})q(\boldsymbol{\theta})}{Z_N} = \frac{\frac{q(\mathcal{D}|\boldsymbol{\theta})}{Z_l} \frac{q(\boldsymbol{\theta})}{Z_0}}{p(\mathcal{D})}$$

$$p(\mathcal{D}) = \frac{Z_N}{Z_l Z_0},$$

for every $\boldsymbol{\theta}$ and $\mathcal{D}$. This result can be used to compute the marginal likelihood of many models that employ conjugate priors.

For instance, consider the Beta-Bernoulli model, already discussed in a previous section. The prior density is given by $p(\theta) = \frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}$, the likelihood by $p(\mathcal{D} \mid \theta) = \theta^{N_1}(1-\theta)^{N_0}$, and the posterior density by $p(\theta \mid \mathcal{D}) = \frac{\theta^{N_1+a-1}(1-\theta)^{N_0+b-1}}{B(N_1+a,N_0+b)}$, for all $\theta \in (0,1)$, dataset $\mathcal{D}$, and valid hyperparameters a and b. Let $q(\theta) = \theta^{a-1}(1-\theta)^{b-1}$, $Z_0 = B(a,b)$, $q(\mathcal{D} \mid \theta) = \theta^{N_1}(1-\theta)^{N_0}$, and $Z_l = 1$. In that case, $Z_N = B(N_1 + a, N_0 + b)$ to satisfy $p(\theta \mid \mathcal{D}) = \frac{q(\mathcal{D}|\theta)q(\theta)}{Z_N}$. Using the result mentioned in the previous paragraph, $p(\mathcal{D}) = \frac{B(N_1+a,N_0+b)}{B(a,b)}$, which we already knew from a previous section.

When the prior over models is uniform, finding $\arg\max_{m'} p(m' \mid \mathcal{D})$ corresponds to finding the model with maximum marginal likelihood $\arg\max_{m'} p(\mathcal{D} \mid m')$. In cases where computing the marginal likelihood $p(\mathcal{D} \mid m)$ for each m is intractable, it is possible to maximize alternative heuristic scores. One example is the BIC score for model m given the dataset $\mathcal{D}$, defined as $\log p(\mathcal{D} \mid m, \hat{\boldsymbol{\theta}}) - \frac{\text{dof}(m)}{2}\log N$, where $\text{dof}(m)$ is the number of degrees of freedom in model m (related to the number of free parameters in m), $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of the parameters for model m under the dataset $\mathcal{D}$, and N is the number of observations in $\mathcal{D}$.

The Bayes factor between models m_0 and m_1 given the dataset $\mathcal{D}$ is defined as the ratio $\frac{p(\mathcal{D}|m_0)}{p(\mathcal{D}|m_1)}$, and can be used to decide between models when the prior over models is uniform. Consider, as an example, a dataset $\mathcal{D} = x_1, \ldots, x_N$, which is iid given Θ according to a Bernoulli distribution $\text{Ber}(\cdot \mid \theta^*)$, for an unknown θ^*. A model m_0 that states with complete certainty that $\theta = \frac{1}{2}$ has marginal likelihood $p(\mathcal{D} \mid m_0) = \frac{1}{2^N}$. A model m_1 such that $p(\theta \mid m_1) = \text{Beta}(\theta \mid a, b)$ has marginal likelihood $p(\mathcal{D} \mid m_1) = \frac{B(N_1+a,N_0+b)}{B(a,b)}$, as previously noted. For instance, letting $a = b = 1$, $N_0 = 2$ and $N_1 = 5$, the Bayes factor is in favor of m_0, whereas if $N_0 = 20$ and $N_1 = 50$, the Bayes factor is overwhelmingly in favor of m_1.

Consider a mixture prior defined by $p(\theta) = \sum_k p(k)p(\theta \mid k)$, for every parameter θ. By definition,

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}$$

$$= \frac{p(\mathcal{D} \mid \theta)\sum_k p(k)p(\theta \mid k)}{p(\mathcal{D})}$$

$$= \sum_k \frac{p(k)p(\mathcal{D} \mid \theta)p(\theta \mid k)}{p(\mathcal{D})}.$$

By the definition of conditional probability, $\frac{p(k)}{p(\mathcal{D})} = \frac{p(k|\mathcal{D})}{p(\mathcal{D}|k)}$, whenever $p(\mathcal{D}), p(\mathcal{D} \mid k) > 0$. Thus,

$$p(\theta \mid \mathcal{D}) = \sum_k \frac{p(k)p(\mathcal{D} \mid \theta)p(\theta \mid k)}{p(\mathcal{D})}$$

$$= \sum_k \frac{p(k \mid \mathcal{D})p(\mathcal{D} \mid \theta)p(\theta \mid k)}{p(\mathcal{D} \mid k)}$$

$$= \sum_k \frac{p(k \mid \mathcal{D})p(\mathcal{D} \mid \theta, k)p(\theta \mid k)}{p(\mathcal{D} \mid k)}$$

$$= \sum_k p(k \mid \mathcal{D})p(\theta \mid \mathcal{D}, k),$$

by simply assuming that the likelihood of any dataset $\mathcal{D}$ is independent of K given Θ, where

$$p(k \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid k)p(k)}{\sum_{k'} p(\mathcal{D} \mid k')p(k')}.$$

In simple terms, this result states that the posterior of a mixture of priors is a particular mixture of corresponding posteriors. If computing the posterior density $p(\theta \mid \mathcal{D}, k)$ and the marginal likelihood $p(\mathcal{D} \mid k)$ are easy tasks given the prior density $p(\theta \mid k)$, for all θ, k and $\mathcal{D}$, then computing the posterior is also easy. By mixing conjugate priors, it is possible to perform tractable inference using very complex priors.

Instead of choosing hyperparameters $\boldsymbol{\eta}$ that define a prior density for every $\boldsymbol{\theta}$, it is also possible to model the uncertainty about hyperparameters (models) by a probability density $p(\boldsymbol{\eta})$, for every $\boldsymbol{\eta}$. This approach is typically called hierarchical Bayes. The resulting prior density for each $\boldsymbol{\theta}$ becomes $p(\boldsymbol{\theta}) = \int_{\text{Val}(\boldsymbol{H})} p(\boldsymbol{\theta} \mid \boldsymbol{\eta}) p(\boldsymbol{\eta}) \, d\boldsymbol{\eta}$, where $\boldsymbol{H}$ is the random variable that represents the hyperparameters. Suppose the likelihood of $\mathcal{D}$ is independent of $\boldsymbol{H}$ given $\boldsymbol{\Theta}$. In that case, the posterior density for $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ given $\mathcal{D}$ becomes

$$p(\boldsymbol{\eta}, \boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\eta}, \boldsymbol{\theta}) p(\boldsymbol{\eta}, \boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{\eta}) p(\boldsymbol{\eta})}{p(\mathcal{D})}.$$

Therefore,

$$p(\boldsymbol{\eta} \mid \mathcal{D}) = \int_{\text{Val}(\boldsymbol{\Theta})} \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{\eta}) p(\boldsymbol{\eta})}{p(\mathcal{D})} \, d\boldsymbol{\theta} = \frac{p(\boldsymbol{\eta})}{p(\mathcal{D})} \int_{\text{Val}(\boldsymbol{\Theta})} p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{\eta}) \, d\boldsymbol{\theta},$$

In this context, the hyperparameters $\arg\max_{\boldsymbol{\eta}'} p(\boldsymbol{\eta}' \mid \mathcal{D})$ are called a type-II maximum a posteriori estimate. The hyperparameters $\arg\max_{\boldsymbol{\eta}'} p(\mathcal{D} \mid \boldsymbol{\eta}')$ are called a type-II maximum likelihood estimate, and the general approach for obtaining this estimate is called empirical Bayes. Using the data to estimate hyperparameters is useful when priors for parameters that are assumed to be similar share the same hyperparameters.

Bayesian decision theory formalizes the task of making decisions under uncertainty. Concretely, let $\mathbf{y} \in \mathcal{Y}$ represent the state of the *environment*, and $\mathbf{x} \in \mathcal{X}$ represent the corresponding state observed by the decision-making *agent*. Also, let $\mathcal{A}$ be the set of *actions* available for the agent, and $\ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$ be a *loss function*. The posterior expected loss $\rho(a \mid \mathbf{x})$ of taking action a given the observation $\mathbf{x}$ is defined by

$$\rho(a \mid \mathbf{x}) = \mathbb{E}_{P(\mathbf{Y} \mid \mathbf{x})}[\ell(\mathbf{Y}, a)] = \int_{\text{Val}(\mathbf{Y})} \ell(\mathbf{y}, a) p(\mathbf{y} \mid \mathbf{x}) \, d\mathbf{y},$$

where $p(\mathbf{y} \mid \mathbf{x})$ is the density associated to state $\mathbf{y}$ given the observation $\mathbf{x}$.

The goal in a decision task is to find an optimal decision policy $\delta : \mathcal{X} \to \mathcal{A}$ such that, for all $\mathbf{x} \in \mathcal{X}$,

$$\delta(\mathbf{x}) = \arg\min_{a \in \mathcal{A}} \rho(a \mid \mathbf{x}).$$

Intuitively, if $\ell(\mathbf{y}, a)$ is the penalty for taking action a in the (typically unknown) state $\mathbf{y}$, the action $\delta(\mathbf{x})$ minimizes the expected penalty given the observation $\mathbf{x}$.

A classification task can be formulated as a decision task by letting $\mathcal{X} = \mathbb{R}^d$ be the set of observations and $\mathcal{Y} = \mathcal{A} = \{1, \ldots, C\}$ be the set of classes. In this case, the optimal decision policy δ is a classifier given by

$$\delta(\mathbf{x}) = \arg\min_{a \in \mathcal{Y}} \rho(a \mid \mathbf{x}) = \arg\min_{a} \sum_{y} \ell(y, a) p(y \mid \mathbf{x}),$$

In a classification task, the typical choice of loss function ℓ is the 0-1 loss function, given by

$$\ell(y, a) = \begin{cases} 0 & \text{if } y = a \\ 1 & \text{if } y \neq a. \end{cases}$$

By definition,

$$\rho(a \mid \mathbf{x}) = \mathbb{E}_{P(Y \mid \mathbf{x})}[\ell(Y, a)] = \sum_{y} \ell(y, a) p(y \mid \mathbf{x}) = \sum_{y : y \neq a} p(y \mid \mathbf{x}) = P(Y \neq a \mid \mathbf{x}) = 1 - P(Y = a \mid \mathbf{x}).$$

Thus,

$$\delta(\mathbf{x}) = \arg\min_{a \in \mathcal{Y}} \rho(a \mid \mathbf{x}) = \arg\min_{a \in \mathcal{Y}} [1 - P(Y = a \mid \mathbf{x})] = \arg\max_{a} P(Y = a \mid \mathbf{x}) = \arg\max_{y} p(y \mid \mathbf{x}).$$

In other words, the maximum a posteriori estimate $\arg\max_y p(y \mid \mathbf{x})$ is the optimal classification for observation $\mathbf{x}$ under a 0-1 loss function.

As an alternative, consider a classification task where one of the available actions r is to *reject* an observation. We let $\mathcal{A} = \mathcal{Y} \cup \{r\}$ represent such extended action set. An appropriate loss function ℓ is given by

$$\ell(y, a) = \begin{cases} 0 & \text{if } a \neq r, a = y \\ \lambda_s & \text{if } a \neq r, a \neq y \\ \lambda_r & \text{if } a = r, \end{cases}$$

for $\lambda_s, \lambda_r > 0$. In this case, it is easy to show that

$$\rho(a \mid \mathbf{x}) = \begin{cases} \lambda_r & \text{if } a = r \\ \lambda_s[1 - P(Y = a \mid \mathbf{x})] & \text{if } a \neq r. \end{cases}$$

As a consequence, the corresponding optimal decision policy (classifier) δ is given by

$$\delta(\mathbf{x}) = \begin{cases} r & \text{if } \max_y p(y \mid \mathbf{x}) < 1 - \frac{\lambda_r}{\lambda_s} \\ \arg\max_y p(y \mid \mathbf{x}) & \text{otherwise.} \end{cases}$$

Consider a *binary* classification task where the loss function ℓ is defined by

$$\ell(y, a) = \begin{cases} 0 & \text{if } a = y \\ \ell_{FN} & \text{if } a \neq y, a = 0 \\ \ell_{FP} & \text{if } a \neq y, a = 1, \end{cases}$$

where $\ell_{FN} > 0$ represents the loss for a false negative, and $\ell_{FP} > 0$ represents the loss for a false positive. In this case, it is easy to show that the optimal decision policy δ is given by $\delta(\mathbf{x}) = 1$ iff $\frac{\rho(1|\mathbf{x})}{\rho(0|\mathbf{x})} > \frac{\ell_{FP}}{\ell_{FN}}$, and 0 otherwise.

Consider an *arbitrary* decision policy (classifier) δ for a binary classification task such that

$$\delta(\mathbf{x}) = \begin{cases} 1, & \text{if } f(\mathbf{x}) > \tau \\ 0, & \text{if } f(\mathbf{x}) \leq \tau, \end{cases}$$

for some function f that represents the confidence that $\mathbf{x}$ belongs to class 1, and a threshold $\tau \in \mathbb{R}$. Consider also a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, for every i. For any threshold τ, it is possible to compute the true positive rate (TPR, ratio of positive observations classified as positive) and false positive rate (FPR, ratio of negative observations classified as positive) of δ on $\mathcal{D}$. Clearly, at most $N+1$ thresholds will result in different pairs of TPR and FPR. The TPR and FPR never increase as the threshold τ increases. A *perfect* classifier δ for $\mathcal{D}$ achieves FPR zero and TPR one for some threshold. A line chart depicting FPR $\times$ TPR can be created by connecting each attainable pair of FPR and TPR, in order of corresponding thresholds. Such chart depicts the so-called receiver operating characteristic (ROC) curve of δ on $\mathcal{D}$. The area under the ROC curve (AUC) can be used to summarize the efficacy of δ on $\mathcal{D}$. The ROC curve or AUC can be used to compare classifiers independently of how false negatives/positives are weighted.

A regression task can also be formulated as a decision task by letting $\mathcal{X} = \mathbb{R}^d$ be the set of observations and $\mathcal{Y} = \mathcal{A} = \mathbb{R}$ be the set of targets. A typical choice of loss function is the squared error loss ℓ defined by $\ell(y, a) = (y - a)^2$. In this case, by definition,

$$\rho(a \mid \mathbf{x}) = \int_{-\infty}^{+\infty} \ell(y, a)p(y \mid \mathbf{x}) \, dy = \int_{-\infty}^{+\infty} (y - a)^2 p(y \mid \mathbf{x}) \, dy = \int_{-\infty}^{+\infty} (y^2 - 2ay + a^2)p(y \mid \mathbf{x}) \, dy$$

$$= \int_{-\infty}^{+\infty} y^2 p(y \mid \mathbf{x}) \, dy - 2a \int_{-\infty}^{+\infty} yp(y \mid \mathbf{x}) \, dy + a^2 \int_{-\infty}^{+\infty} p(y \mid \mathbf{x}) \, dy$$

$$= \mathbb{E}[Y^2 \mid \mathbf{x}] - 2a\mathbb{E}[Y \mid \mathbf{x}] + a^2.$$

Notice that $\rho(a \mid \mathbf{x})$ is differentiable with respect to a. Therefore, if a is a local minima of $\rho(\cdot \mid \mathbf{x})$, then

$$\frac{\partial}{\partial a}\rho(a \mid \mathbf{x}) = -2\mathbb{E}[Y \mid \mathbf{x}] + 2a = 0.$$

Rearranging terms, if a is a local minima of $\rho(\cdot \mid \mathbf{x})$, then $a = \mathbb{E}[Y \mid \mathbf{x}]$. The second derivative test confirms that such a is always local minima. Therefore,

$$\delta(\mathbf{x}) = \arg\min_a \rho(a \mid \mathbf{x}) = \mathbb{E}[Y \mid \mathbf{x}] = \int_{-\infty}^{+\infty} yp(y \mid \mathbf{x}) \, dy.$$

In other words, the posterior mean estimate is the optimal assignment for observation $\mathbf{x}$ under a squared error loss function.

If the absolute loss function ℓ defined by $\ell(y, a) = |y - a|$ is used instead of the squared error loss function, the estimate corresponding to the optimal assignment $\delta(\mathbf{x})$ is the posterior median given $\mathbf{x}$. We omit the proof, which can be obtained by applying the Leibniz integral rule to $\frac{\partial}{\partial a}\rho(a \mid \mathbf{x})$.

Consider a supervised learning task (either regression or classification), and a corresponding decision policy (classifier) $\delta : \mathcal{X} \to \mathcal{Y}$. The generalization error $L(\boldsymbol{\theta}, \delta)$ of δ on an environment parameterized by $\boldsymbol{\theta}$ is defined by

$$L(\boldsymbol{\theta}, \delta) = \mathbb{E}[\ell(Y, \delta(\mathbf{X}))] = \int_{\text{Val}(Y)} \int_{\text{Val}(\mathbf{X})} \ell(y, \delta(\mathbf{x}))p(\mathbf{x} \mid y, \boldsymbol{\theta})p(y \mid \boldsymbol{\theta}) \, d\mathbf{x}dy,$$

where $p(\mathbf{x}, y \mid \boldsymbol{\theta})$ is the density associated to the observation-target pair $(\mathbf{x}, y)$.

Intuitively, the generalization error $L(\boldsymbol{\theta}, \delta)$ averages every penalty $\ell(y, \delta(\mathbf{x}))$, weighted by the probability of each observation-target pair $(\mathbf{x}, y)$ occurring in the environment (given $\boldsymbol{\theta}$).

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots (\mathbf{x}_N, y_N)$, which is iid given $\boldsymbol{\Theta}$ and distributed according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for an unknown $\boldsymbol{\theta}^*$. The goal in supervised learning is to find a decision policy δ^* that minimizes the posterior expected loss $\rho(\delta \mid \mathcal{D})$ of δ given $\mathcal{D}$ defined by

$$\rho(\delta \mid \mathcal{D}) = \int_{\text{Val}(\boldsymbol{\Theta})} L(\boldsymbol{\theta}, \delta)p(\boldsymbol{\theta} \mid \mathcal{D}) \, d\boldsymbol{\theta}.$$

6 Frequentist statistics

Frequentist statistical methods avoid representing uncertainty about parameters using probabilities. Instead, they focus on the sampling distribution of estimators.

Suppose that each assignment $\mathcal{D}$ to a random variable $\boldsymbol{\mathcal{D}}$ represents an iid dataset with N elements, distributed according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for an unknown $\boldsymbol{\theta}^*$.

An estimator δ is a function that maps each dataset $\mathcal{D}$ to a parameter estimate $\hat{\boldsymbol{\theta}}$ (e.g., maximum likelihood). The probability distribution for the random variable $\delta(\boldsymbol{\mathcal{D}})$ is called the sampling distribution of the estimator δ. An estimator δ is consistent if its estimates converge in probability to the true unknown parameters when the size of the dataset tends to infinity.

The bootstrap is a method to approximate the sampling distribution of an estimator δ. Given an observed dataset $\mathcal{D}$, parametric bootstrap relies on sampling S datasets $\mathcal{D}_1, \ldots, \mathcal{D}_S$ of size N from the distribution $p(\cdot \mid \hat{\boldsymbol{\theta}})$, where $\hat{\boldsymbol{\theta}} = \delta(\mathcal{D})$. Non-parametric bootstrap relies on sampling N elements (with replacement) from the elements in the observed dataset $\mathcal{D}$ to create each dataset $\mathcal{D}_i$, for $i \in \{1, \ldots, S\}$. In both cases, the sampling distribution of the estimator δ can be approximated, for instance, by the empirical distribution given by $p(\boldsymbol{\theta}) = \frac{1}{S}\sum_s \mathbb{I}[\delta(\mathcal{D}_s) = \boldsymbol{\theta}]$. The probability density $p(\boldsymbol{\theta})$ should not be confused with the approximate density associated to a particular parameterization $\boldsymbol{\theta}$. Instead, $p(\boldsymbol{\theta})$ is the approximate density associated to making a particular estimate $\boldsymbol{\theta}$ when considering all possible datasets distributed according to the distribution governed by the (unknown and fixed) parameter $\boldsymbol{\theta}^*$.

In frequentist decision theory, the expected risk $R(\boldsymbol{\theta}, \delta)$ of the estimator δ on an environment parameterized by $\boldsymbol{\theta}$ is defined by

$$R(\boldsymbol{\theta}, \delta) = \mathbb{E}_{P(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta})}[L(\boldsymbol{\theta}, \delta(\boldsymbol{\mathcal{D}}))] = \int_{\text{Val}(\boldsymbol{\mathcal{D}})} L(\boldsymbol{\theta}, \delta(\mathcal{D}))p(\mathcal{D} \mid \boldsymbol{\theta}) \, d\mathcal{D},$$

where $L(\boldsymbol{\theta}, \delta(\mathcal{D}))$ is the loss associated to estimating $\delta(\mathcal{D})$ when the environment is parameterized by $\boldsymbol{\theta}$. Intuitively, the expected risk $R(\boldsymbol{\theta}, \delta)$ corresponds to the average loss of the estimates over all possible datasets, weighted by the probability of each dataset given the true (but typically unknown) $\boldsymbol{\theta}$.

It is important to compare the expected risk to the posterior expected loss defined in the previous section. Concretely, let $\mathcal{Y} = \text{Val}(\boldsymbol{\Theta})$ be the set of states of the environment, $\mathcal{X} = \text{Val}(\boldsymbol{\mathcal{D}})$ be the set of observations, and $\mathcal{A} = \mathcal{Y}$ be the set of actions. The posterior expected loss $\rho(\delta(\mathcal{D}) \mid \mathcal{D})$ of estimate (action) $\delta(\mathcal{D})$ given the dataset $\mathcal{D}$ is given by

$$\rho(\delta(\mathcal{D}) \mid \mathcal{D}) = \mathbb{E}_{P(\boldsymbol{\Theta}|\mathcal{D})}[L(\boldsymbol{\Theta}, \delta(\mathcal{D}))] = \int_{\text{Val}(\boldsymbol{\Theta})} L(\boldsymbol{\theta}, \delta(\mathcal{D}))p(\boldsymbol{\theta} \mid \mathcal{D}) \, d\boldsymbol{\theta}.$$

Intuitively, the posterior expected loss corresponds to the average loss of the estimates over all parameterizations, weighted by the probability of each parameterization given the observed data.

The expected risk and the posterior expected loss are connected through the Bayes risk. The Bayes risk of estimator δ for loss function L and prior density $p(\boldsymbol{\theta})$ for each state $\boldsymbol{\theta}$ of the environment is given by

$$
\begin{aligned}
\mathbb{E}_{P(\boldsymbol{\Theta})}[R(\boldsymbol{\Theta}, \delta)] &= \int_{\mathrm{Val}(\boldsymbol{\Theta})} R(\boldsymbol{\theta}, \delta) p(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \\
&= \int_{\mathrm{Val}(\boldsymbol{\Theta})} \left[\int_{\mathrm{Val}(\boldsymbol{\mathcal{D}})} L(\boldsymbol{\theta}, \delta(\mathcal{D})) p(\mathcal{D} \mid \boldsymbol{\theta}) \, d\mathcal{D} \right] p(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \\
&= \int_{\mathrm{Val}(\boldsymbol{\Theta})} \int_{\mathrm{Val}(\boldsymbol{\mathcal{D}})} L(\boldsymbol{\theta}, \delta(\mathcal{D})) p(\mathcal{D}, \boldsymbol{\theta}) \, d\mathcal{D} \, d\boldsymbol{\theta} \\
&= \int_{\mathrm{Val}(\boldsymbol{\mathcal{D}})} \int_{\mathrm{Val}(\boldsymbol{\Theta})} L(\boldsymbol{\theta}, \delta(\mathcal{D})) p(\boldsymbol{\theta}, \mathcal{D}) \, d\boldsymbol{\theta} d\mathcal{D}. \\
&= \int_{\mathrm{Val}(\boldsymbol{\mathcal{D}})} \left[\int_{\mathrm{Val}(\boldsymbol{\Theta})} L(\boldsymbol{\theta}, \delta(\mathcal{D})) p(\boldsymbol{\theta} \mid \mathcal{D}) \, d\boldsymbol{\theta} \right] p(\mathcal{D}) \, d\mathcal{D} \\
&= \int_{\mathrm{Val}(\boldsymbol{\mathcal{D}})} \rho(\delta(\mathcal{D}) \mid \mathcal{D}) p(\mathcal{D}) \, d\mathcal{D} \\
&= \mathbb{E}_{P(\boldsymbol{\mathcal{D}})}[\rho(\delta(\boldsymbol{\mathcal{D}}) \mid \boldsymbol{\mathcal{D}})],
\end{aligned}
$$

under the assumptions that enable Fubini's theorem. Therefore, an estimator that minimizes the posterior expected loss for each dataset individually will also minimize the Bayes risk. An estimator that minimizes the Bayes risk is called a Bayes estimator.

The maximum risk of an estimator δ is defined as $\max_{\boldsymbol{\theta}} R(\boldsymbol{\theta}, \delta)$. The estimator with minimum maximum risk is called the minimax estimator, and intuitively corresponds to a very *defensive* estimator. Minimizing the maximum risk does not require a prior over states of the environment.

Consider two estimators δ_1 and δ_2, and suppose $R(\boldsymbol{\theta}, \delta_1) < R(\boldsymbol{\theta}, \delta_2)$ for all $\boldsymbol{\theta}$. In this case, the estimator δ_1 is said to strictly dominate the estimator δ_2. Intuitively, the estimator δ_1 is always a better choice according to the expected risk. An estimator is admissible when it is not strictly dominated by any other estimator.

The bias of an estimator δ is defined as $\mathrm{bias}(\delta) = \mathbb{E}_{P(\boldsymbol{\mathcal{D}} \mid \boldsymbol{\theta}^*)}[\delta(\boldsymbol{\mathcal{D}}) - \boldsymbol{\theta}^*]$, where $\boldsymbol{\theta}^*$ is the true parameter that generates the data. An estimator is unbiased if its bias is zero.

As an example, consider a dataset $\mathcal{D} = x_1, \ldots, x_N$, which is iid according to a Gaussian distribution $\mathcal{N}(\cdot \mid \mu, \sigma^2)$. We have already shown that the MLE for the mean is given by $\hat{x} = \frac{1}{N} \sum_i x_i$, and the MLE for the variance is $\hat{\sigma}^2 = \frac{1}{N} \sum_i (x_i - \hat{x})^2$. Therefore, the bias of the maximum likelihood estimator for the mean (seen individually) is given by

$$
\begin{aligned}
\mathbb{E}_{P(\boldsymbol{\mathcal{D}})}[\hat{X} - \mu] &= \mathbb{E}_{P(\boldsymbol{\mathcal{D}})}\left[\frac{1}{N} \sum_i X_i \right] - \mathbb{E}_{P(\boldsymbol{\mathcal{D}})}[\mu] \\
&= \frac{1}{N} \mathbb{E}_{P(\boldsymbol{\mathcal{D}})}\left[\sum_i X_i \right] - \mu \\
&= \frac{1}{N} \sum_i \mathbb{E}_{P(X_i)}[X_i] - \mu \\
&= \frac{N\mu}{N} - \mu = 0.
\end{aligned}
$$

The bias of the maximum likelihood estimator for the variance is given by

$$
\begin{aligned}
\mathbb{E}_{P(\boldsymbol{\mathcal{D}})}\left[\frac{1}{N} \sum_i (X_i - \hat{X})^2 - \sigma^2 \right] &= \frac{1}{N} \mathbb{E}_{P(\boldsymbol{\mathcal{D}})}\left[\sum_i (X_i - \hat{X})^2 \right] - \sigma^2 \\
&= \frac{1}{N} \sum_i \mathbb{E}_{P(\boldsymbol{\mathcal{D}})}\left[X_i^2 \right] - 2\mathbb{E}_{P(\boldsymbol{\mathcal{D}})}\left[X_i \hat{X} \right] + \mathbb{E}_{P(\boldsymbol{\mathcal{D}})}\left[\hat{X}^2 \right] - \sigma^2 \\
&= \sigma^2 \left(\frac{N-1}{N} - 1 \right),
\end{aligned}
$$

although we omit important steps for brevity. Intuitively, the ML estimator typically underestimates the variance because it employs the empirical mean $\hat{x}$ instead of the true (but unknown) mean μ. An unbiased estimator for the variance is given by $\frac{N}{N-1}\hat{\sigma}^2 = \frac{1}{N-1} \sum_i (x_i - \hat{x})^2$.

An unbiased estimator does not necessarily minimize the expected risk. For instance, consider a loss function L given by $L(\theta^*, \theta) = (\theta^* - \theta)^2$. The corresponding expected risk of estimator δ is the mean squared error given by

$$R(\theta^*, \delta) = \mathbb{E}_{P(\mathcal{D}|\theta^*)}[L(\theta^*, \delta(\mathcal{D}))] = \int_{\text{Val}(\mathcal{D})} (\theta^* - \delta(\mathcal{D}))^2 p(\mathcal{D} \mid \theta^*) \, d\mathcal{D}.$$

Let $\bar{\theta} = \mathbb{E}_{P(\mathcal{D}|\theta^*)}[\delta(\mathcal{D})]$ denote the expected estimate. It can be easily shown that

$$R(\theta^*, \delta) = \mathbb{E}_{P(\mathcal{D}|\theta^*)}[(\delta(\mathcal{D}) - \bar{\theta})^2] + \mathbb{E}_{P(\mathcal{D}|\theta^*)}[\delta(\mathcal{D}) - \theta^*]^2.$$

In words, the expected risk under a squared error loss is given by the variance of the estimator plus the squared bias of the estimator. Therefore, a biased estimator might be beneficial, as long as its variance is low. This bias-variance trade-off is highly related to the concept of overfitting.

Consider a supervised learning task (either regression or classification), and a corresponding decision policy (estimator) δ. The risk $R(\boldsymbol{\theta}, \delta)$ of the decision policy δ on an environment parameterized by $\boldsymbol{\theta}$ can be redefined as

$$R(\boldsymbol{\theta}, \delta) = \int_{\text{Val}(Y)} \int_{\text{Val}(\boldsymbol{X})} L(y, \delta(\mathbf{x})) p(\mathbf{x}, y \mid \boldsymbol{\theta}) \, d\mathbf{x} dy,$$

where $L(y, \delta(\mathbf{x}))$ is the loss associated to predicting $\delta(\mathbf{x})$ when the target is y, and $p(\mathbf{x}, y \mid \boldsymbol{\theta})$ is the density associated to $(\mathbf{x}, y)$ given $\boldsymbol{\theta}$.

Consider a supervised dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$ for an unknown $\boldsymbol{\theta}^*$. The empirical risk $R_e(\mathcal{D}, \delta)$ is defined as

$$R_e(\mathcal{D}, \delta) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \delta(\mathbf{x}_i)),$$

which is a Monte Carlo approximation to the risk $R(\boldsymbol{\theta}^*, \delta)$. The task of finding $\delta^* = \arg\min_\delta R_e(\mathcal{D}, \delta)$ for a given dataset $\mathcal{D}$ is called empirical risk minimization, and is crucial in supervised learning.

Regularized risk minimization is the task of finding $\delta^* = \arg\min_\delta R_e(\mathcal{D}, \delta) + \lambda C(\delta)$, where $C(\delta) \in \mathbb{R}$ measures the complexity of the decision policy δ, and $\lambda \in \mathbb{R}$ is a hyperparameter that penalizes complexity. Penalizing complex decision policies is a way to avoid overfitting.

In structural risk minimization, the penalty hyperparameter for regularized risk minimization is chosen as

$$\lambda^* = \arg\min_\lambda \hat{R}(\arg\min_\delta \left[R_e(\mathcal{D}, \delta) + \lambda C(\delta) \right]),$$

where $\hat{R}(\delta)$ is a *risk estimate* for the decision policy δ. In words, the penalty hyperparameter is chosen in a way that leads to choosing a decision policy with low risk estimate. A typical choice of risk estimate is the average k-fold cross validation empirical risk of δ.

7 Linear regression

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$ for an unknown $\boldsymbol{\theta}^*$. Suppose also that $\mathbf{x}_i \in \mathbb{R}^D$, and $y_i \in \mathbb{R}$, for all i. Consider the task of regression, which consists on predicting targets from observations based on generalization from $\mathcal{D}$.

Linear regression is a regression technique that assumes that the density $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ associated to target y given the observation $\mathbf{x}$ and parameter $\boldsymbol{\theta}$ can be written as

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \mathbf{w}\mathbf{x}, \sigma^2),$$

for every $\mathbf{x}$, for some weight vector $\mathbf{w}$ and variance σ^2, which are represented in $\boldsymbol{\theta}$. Intuitively, the uncertainty about the target is given by a Gaussian distribution with mean $f(\mathbf{x}) = \mathbf{w}\mathbf{x}$ and variance σ^2. Notice that the predictor $f : \mathbb{R}^D \to \mathbb{R}$ is a linear functional, and $f(\mathbf{x})$ is the optimal prediction given $\mathbf{x}$ under a squared error loss function according to Bayesian decision theory.

Although the assumptions made by linear regression appear very restrictive, an input observation $\mathbf{x}$ can be transformed by a pre-defined basis function $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^{D'}$ before the technique is applied. In that case, we say the density $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ is given by

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \mathbf{w}\boldsymbol{\phi}(\mathbf{x}), \sigma^2).$$

If we let $f(\mathbf{x}) = \mathbf{w}\phi(\mathbf{x})$, the corresponding predictor $f : \mathbb{R}^D \to \mathbb{R}$ is no longer (necessarily) a linear functional in the original space.

For instance, consider a dataset composed of single dimensional observations ($D = 1$), and the polynomial basis function ϕ given by

$$\phi(x) = \left(1, x^1, \ldots, x^{D'}\right),$$

for some $D' \geq 1$. In this case, the corresponding predictor $f : \mathbb{R} \to \mathbb{R}$ is given, for some $\mathbf{w}$, by

$$f(x) = \mathbf{w}\phi(x) = \sum_{j=0}^{D'} w_j x^j,$$

which is (potentially) a polynomial of degree D' on x. Therefore, linear regression is able to fit a polynomial of arbitrary degree to a dataset composed of single dimensional observations. This example generalizes to polynomials involving multiple variables ($D > 1$).

In what follows, we assume that the observations were already transformed by a basis function. It is advisable to at least prefix each original observation $\mathbf{x}$ by the constant 1, leading to a transformed observation $\mathbf{x}' = (1, \mathbf{x})$, so that the first element w_0 of the weight vector $\mathbf{w}$ becomes the intercept the corresponding predictor f given by $f(\mathbf{x}) = w_0 + \mathbf{w}_{1:D}\mathbf{x}$.

The (conditional) likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of the dataset $\mathcal{D}$ under $\boldsymbol{\theta}$ according to a linear regression model is given by

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = p(\mathbf{y} \mid \mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\theta}) = \prod_{i=1}^{N} p(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = \prod_{i=1}^{N} \mathcal{N}(y_i \mid \mathbf{w}\mathbf{x}_i, \sigma^2),$$

for a choice of $\mathbf{w}$ and σ^2, and where $\mathbf{y} = (y_1, \ldots, y_N)$. Notice that this likelihood is not based on the joint density over observations and targets, which is irrelevant for our purposes. Instead, $p(\mathcal{D} \mid \boldsymbol{\theta})$ is the density associated to the particular assignment $\mathbf{y}$ when the observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are seen as constants (according to $\mathcal{D}$), and $Y_i \perp\!\!\!\perp \mathbf{X}_{-i} \mid \mathbf{X}_i, \boldsymbol{\Theta}$.

The log-likelihood $\ell(\boldsymbol{\theta}) = \log p(\mathcal{D} \mid \boldsymbol{\theta})$ of the dataset $\mathcal{D}$ under $\boldsymbol{\theta}$ is given by

$$
\begin{aligned}
\log \prod_{i=1}^{N} \mathcal{N}(y_i \mid \mathbf{x}_i\mathbf{w}, \sigma^2) &= \log \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \mathbf{w}\mathbf{x}_i)^2}{2\sigma^2}} = \sum_{i=1}^{N} \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \mathbf{w}\mathbf{x}_i)^2}{2\sigma^2}} \right] \\
&= \sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} + \sum_{i=1}^{N} \log e^{-\frac{(y_i - \mathbf{w}\mathbf{x}_i)^2}{2\sigma^2}} \\
&= N \log \frac{1}{\sqrt{2\pi\sigma^2}} + \sum_{i=1}^{N} -\frac{(y_i - \mathbf{w}\mathbf{x}_i)^2}{2\sigma^2} \\
&= -\frac{N}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \mathbf{w}\mathbf{x}_i)^2.
\end{aligned}
$$

Consider the task of finding a maximum likelihood estimate $\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathcal{D} \mid \mathbf{w}, \sigma^2)$ given a dataset $\mathcal{D}$ and a constant variance σ^2. Finding such estimate would be sufficient to define a candidate predictor f. Clearly, the maximum likelihood estimate also minimizes half the residual sum of squares $\mathrm{RSS}(\cdot)$ given by

$$\frac{\mathrm{RSS}(\mathbf{w})}{2} = \frac{1}{2} \sum_{i=1}^{N} (y_i - \mathbf{w}\mathbf{x}_i)^2 = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}),$$

where $\mathbf{X}$ is the so-called *design matrix*, where each row i corresponds to observation $\mathbf{x}_i$, and $\mathbf{y} = (y_1, \ldots, y_N)$. Using elementary properties of matrices,

$$
\begin{aligned}
\frac{\mathrm{RSS}(\mathbf{w})}{2} &= \frac{1}{2}(\mathbf{y}^T - (\mathbf{X}\mathbf{w})^T)(\mathbf{y} - \mathbf{X}\mathbf{w}) = \frac{1}{2}\left[\mathbf{y}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) - (\mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})\right] \\
&= \frac{1}{2}\left[\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} - (\mathbf{X}\mathbf{w})^T\mathbf{y} + (\mathbf{X}\mathbf{w})^T\mathbf{X}\mathbf{w}\right] \\
&= \frac{1}{2}\left[\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}\right] \\
&= \frac{1}{2}\left[\mathbf{y}^T\mathbf{y} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}\right].
\end{aligned}
$$

Using the facts that $\nabla_{\mathbf{a}}[\mathbf{a}^T \mathbf{A} \mathbf{a}] = (\mathbf{A} + \mathbf{A}^T)\mathbf{a}$, and $\nabla_{\mathbf{a}}[\mathbf{a}\mathbf{b}] = \mathbf{b}$, the gradient of $\frac{\text{RSS}(\mathbf{w})}{2}$ with respect to $\mathbf{w}$ is given by

$$\nabla_{\mathbf{w}}\left[\frac{\text{RSS}(\mathbf{w})}{2}\right] = \frac{1}{2}\nabla_{\mathbf{w}}[\mathbf{y}^T\mathbf{y}] - \nabla_{\mathbf{w}}[\mathbf{w}^T\mathbf{X}^T\mathbf{y}] + \frac{1}{2}\nabla_{\mathbf{w}}[\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}]$$

$$= -\mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\mathbf{w},$$

since $\mathbf{X}^T\mathbf{X}$ is always symmetric. Finally, if $\hat{\mathbf{w}}$ is a local minimum of $\frac{\text{RSS}(\cdot)}{2}$, then

$$\mathbf{X}^T\mathbf{X}\hat{\mathbf{w}} = \mathbf{X}^T\mathbf{y},$$

by equating the gradient found above with zero. Assume that $\mathbf{X}^T\mathbf{X}$ is invertible, which is the case whenever the columns in $\mathbf{X}$ are linearly independent. In that case, if a local minimum $\hat{\mathbf{w}}$ exists, then it must be given by

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

The equation above is a sufficient condition for $\hat{\mathbf{w}}$ to be a maximum likelihood estimate, although we omit the proof. There are more efficient methods to obtain this estimate, although the details are out of the scope of this text.

Given the maximum likelihood estimate $\hat{\mathbf{w}}$, the vector $\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}}$ has an interesting geometrical interpretation. Clearly $\hat{\mathbf{y}}$ belongs to the span S of the columns of $\mathbf{X}$. In fact, $\hat{\mathbf{y}}$ is the vector in S that is closest (in Euclidean distance) to the desired target $\mathbf{y}$. Notice that this is related to the assumption that the columns in $\mathbf{X}$ are linearly independent, which is necessary to obtain the maximum likelihood estimate $\hat{\mathbf{w}}$.

The negative log-likelihood of a linear regression model is convex with respect to $\mathbf{w}$. This is a highly desirable property, since finding a (global) minimum generally becomes much easier.

However, a maximum likelihood estimate $\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathcal{D} \mid \mathbf{w}, \sigma^2)$ may have very large coefficients (in absolute value), making the corresponding predictor f very sensitive to small changes in its inputs.

Instead, consider the task of maximizing the posterior density $p(\mathbf{w} \mid \mathcal{D})$ with respect to $\mathbf{w}$. Firstly, suppose the prior density $p(\mathbf{w})$ associated to the weight vector $\mathbf{w}$ is given by

$$p(\mathbf{w}) = \prod_{j=1}^{D} \mathcal{N}(w_j \mid 0, \tau^2).$$

In words, this represents the prior belief that each coefficient is distributed according to a Gaussian distribution with mean zero and variance τ^2.

Naturally, the maximum a posteriori estimate $\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathcal{D})$ also maximizes

$$\log p(\mathbf{w} \mid \mathcal{D}) = \log p(\mathcal{D} \mid \mathbf{w}) + \log p(\mathbf{w}) - \log p(\mathcal{D}).$$

Since the last term is a constant with respect to $\mathbf{w}$, the desired estimate can be obtained by maximizing $\log p(\mathcal{D} \mid \mathbf{w}) + \log p(\mathbf{w})$ with respect to $\mathbf{w}$.

Firstly, note that

$$\log p(\mathbf{w}) = \log \prod_{j=1}^{D} \mathcal{N}(w_j \mid 0, \tau^2) = -\sum_{j=1}^{D} \frac{w_j^2}{2\tau^2} - \sum_{j=1}^{D} \log \sqrt{2\pi\tau^2} = -\frac{1}{2\tau^2}\sum_{j=1}^{D} w_j^2 - \frac{D}{2}\log 2\pi\tau^2.$$

Therefore,

$$\log p(\mathcal{D} \mid \mathbf{w}) + \log p(\mathbf{w}) = -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mathbf{w}\mathbf{x}_i)^2 - \frac{1}{2\tau^2}\sum_{j=1}^{D} w_j^2 - \frac{D}{2}\log 2\pi\tau^2.$$

By eliminating the constants with respect to $\mathbf{w}$ and letting $\sigma^2 = \lambda\tau^2$, the maximum a posteriori estimate $\hat{\mathbf{w}}$ minimizes the cost function J given by

$$J(\mathbf{w}) = \sum_{i=1}^{N}(y_i - \mathbf{w}\mathbf{x}_i)^2 + \lambda \sum_{j=1}^{D} w_j^2.$$

The hyperparameter λ relates the prior beliefs about the variance of the targets given the observations to the variance of the coefficients around zero, and intuitively penalizes weight vectors with large norms. Penalizing large coefficients is a specific instance of a general approach called regularization.

Let $\boldsymbol{I}$ denote the $D \times D$ identity matrix. If we assume that $\lambda \boldsymbol{I} + \mathbf{X}^T\mathbf{X}$ is invertible, then a local minimum $\hat{\mathbf{w}}$ of J must be given by

$$\hat{\mathbf{w}} = (\lambda \boldsymbol{I} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},$$

using a similar argument to the one that led to the maximum likelihood estimate. The equation above is also a sufficient condition for $\hat{\mathbf{w}}$ to be a maximum a posteriori estimate, although we omit the proof.

Note that if a coefficient w_0 in a weight vector is used as an intercept term (due to a choice of basis function ϕ), then it should not be penalized by regularization, since it does not make the corresponding predictor f more sensitive to small changes in its inputs.

Consider the task of computing the posterior density $p(\mathbf{w} \mid \mathcal{D}, \sigma^2)$ given by

$$p(\mathbf{w} \mid \mathcal{D}, \sigma^2) = \frac{p(\mathcal{D} \mid \mathbf{w}, \sigma^2)p(\mathbf{w} \mid \sigma^2)}{p(\mathcal{D} \mid \sigma^2)} \propto_{\mathbf{w}} p(\mathcal{D} \mid \mathbf{w}, \sigma^2)p(\mathbf{w} \mid \sigma^2).$$

Firstly, notice that the likelihood $p(\mathcal{D} \mid \mathbf{w}, \sigma^2)$ can be written as

$$p(\mathcal{D} \mid \mathbf{w}, \sigma^2) = \prod_{i=1}^{N} \mathcal{N}(y_i \mid \mathbf{w}\mathbf{x}_i, \sigma^2) = \mathcal{N}(\mathbf{y} \mid \mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}),$$

where $\mathbf{y} = (y_1, \ldots, y_N)$, $\mathbf{X}$ is the design matrix corresponding to $\mathcal{D}$, and $\mathbf{I}$ is the $N \times N$ identity matrix.

Furthermore, suppose the prior density for $\mathbf{w}$ is given by $p(\mathbf{w} \mid \sigma^2) = \mathcal{N}(\mathbf{w} \mid \mathbf{w}_0, \mathbf{V}_0)$, for some hyperparameters $\mathbf{w}_0$ and $\mathbf{V}_0$.

In that case, the posterior density is given by $p(\mathbf{w} \mid \mathcal{D}, \sigma^2) \propto_{\mathbf{w}} \mathcal{N}(\mathbf{y} \mid \mathbf{X}\mathbf{w}, \sigma^2\mathbf{I})\mathcal{N}(\mathbf{w} \mid \mathbf{w}_0, \mathbf{V}_0)$. By noting that the random variables $\mathbf{W}$ and $\mathbf{Y}$ are related by a linear Gaussian system (conditionally on σ^2), the desired posterior density is given by

$$p(\mathbf{w} \mid \mathcal{D}, \sigma^2) = \mathcal{N}(\mathbf{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where

$$\boldsymbol{\Sigma}^{-1} = \mathbf{V}_0^{-1} + \mathbf{X}^T[\sigma^2\mathbf{I}]^{-1}\mathbf{X} = \mathbf{V}_0^{-1} + \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}[\mathbf{X}^T[\sigma^2\mathbf{I}]^{-1}\mathbf{y} + \mathbf{V}_0^{-1}\mathbf{w}_0] = \frac{1}{\sigma^2}\boldsymbol{\Sigma}\mathbf{X}^T\mathbf{y} + \boldsymbol{\Sigma}\mathbf{V}_0^{-1}\mathbf{w}_0.$$

Consider again the assumptions $\mathbf{V}_0 = \tau^2\mathbf{I}$ and $\mathbf{w}_0 = \mathbf{0}$, which were used to obtain a maximum a posteriori estimate, and let $\sigma^2 = \lambda\tau^2$. The corresponding posterior parameters $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ are given by

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\tau^2}\mathbf{I} + \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X} = \frac{1}{\tau^2}\left[\mathbf{I} + \frac{1}{\lambda}\mathbf{X}^T\mathbf{X}\right]$$

$$\boldsymbol{\mu} = \frac{1}{\lambda\tau^2}\left[\frac{1}{\tau^2}\left[\mathbf{I} + \frac{1}{\lambda}\mathbf{X}^T\mathbf{X}\right]\right]^{-1}\mathbf{X}^T\mathbf{y} = \frac{1}{\lambda}\left[\mathbf{I} + \frac{1}{\lambda}\mathbf{X}^T\mathbf{X}\right]^{-1}\mathbf{X}^T\mathbf{y} = \left[\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X}\right]^{-1}\mathbf{X}^T\mathbf{y},$$

by twice using the fact that $(c\mathbf{A})^{-1} = c^{-1}\mathbf{A}^{-1}$, for any $c \neq 0$ and invertible matrix $\mathbf{A}$. Therefore, under the same assumptions, the posterior mean matches the maximum a posteriori estimate. This follows from the fact that the multivariate Gaussian mean and mode are the same.

Finally, consider the task of computing the posterior predictive density $p(y \mid \mathbf{x}, \mathcal{D}, \sigma^2)$ associated to target y given a new observation $\mathbf{x}$, and the observed dataset $\mathcal{D}$. By marginalizing the parameters,

$$p(y \mid \mathbf{x}, \mathcal{D}, \sigma^2) = \int_{\text{Val}(\mathbf{W})} p(y, \mathbf{w} \mid \mathbf{x}, \mathcal{D}, \sigma^2)d\mathbf{w} = \int_{\text{Val}(\mathbf{W})} p(y \mid \mathbf{x}, \mathbf{w}, \mathcal{D}, \sigma^2)p(\mathbf{w} \mid \mathbf{x}, \mathcal{D}, \sigma^2)d\mathbf{w}$$

$$= \int_{\text{Val}(\mathbf{W})} p(y \mid \mathbf{x}, \mathbf{w}, \sigma^2)p(\mathbf{w} \mid \mathcal{D}, \sigma^2)d\mathbf{w} = \int_{\text{Val}(\mathbf{W})} \mathcal{N}(y \mid \mathbf{w}\mathbf{x}, \sigma^2)\mathcal{N}(\mathbf{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})d\mathbf{w},$$

where $\boldsymbol{\mu}$ is the posterior mean and $\boldsymbol{\Sigma}$ is the posterior covariance matrix, for a particular choice of prior hyperparameters $\mathbf{w}_0, \mathbf{V}_0$ and σ^2. Intuitively, the posterior predictive density of target y given $\mathbf{x}$ and $\mathcal{D}$ corresponds

to the average probability of y given $\mathbf{x}$ under each parameterization, weighted by the posterior density of each parameterization given $\mathcal{D}$.

Because the random variables $\mathbf{W}$ and Y are related (conditionally on $\mathcal{D}$ and σ^2) by a linear Gaussian system, the marginalization of their joint density function over $\mathbf{W}$ results in a posterior predictive density given by

$$p(y \mid \mathbf{x}, \mathcal{D}, \sigma^2) = \mathcal{N}(y \mid \boldsymbol{\mu}\mathbf{x}, \sigma^2 + \mathbf{x}^T \boldsymbol{\Sigma} \mathbf{x}).$$

Notice that the posterior predictive mean matches the prediction made by the maximum a posteriori estimate. However, the posterior predictive provides a measure of uncertainty that takes into consideration the underlying variance and the covariance of the parameters after the data is observed.

8 Logistic regression

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$ for some unknown $\boldsymbol{\theta}^*$. Also, suppose $\mathbf{x}_i \in \mathbb{R}^D$, and $y_i \in \{0, 1\}$, for all i. Consider the task of binary classification, which corresponds to predicting binary targets from observations based on generalization from $\mathcal{D}$.

Logistic regression is a technique that assumes that the probability $p(y \mid \mathbf{x}, \mathbf{w})$ of class $y \in \{0, 1\}$ given observation $\mathbf{x}$ and weight vector $\mathbf{w}$ is given by

$$p(y \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}\mathbf{x})^y (1 - \sigma(\mathbf{w}\mathbf{x}))^{1-y},$$

where σ is the sigmoid function given by

$$\sigma(t) = \frac{1}{1 + e^{-t}},$$

for all $t \in \mathbb{R}$. Intuitively, a logistic regression model assigns more probability to class 1 whenever $\mathbf{w}\mathbf{x} > 0$, and each element w_j of $\mathbf{w}$ indicates how much feature j contributes (or detracts) to the detection of class 1. Because the probability of the distinct classes given $\mathbf{x}$ is the same if and only if $\mathbf{w}\mathbf{x} = 0$, the decision boundary of this classification technique is the hyperplane $S = \{\mathbf{x} \in \mathbb{R}^D \mid \mathbf{w}\mathbf{x} = 0\}$.

Although the assumptions made by logistic regression appear very restrictive, an input observation $\mathbf{x}$ can be transformed by a pre-defined basis function $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^{D'}$ before the technique is applied. In that case, we say the probability $p(y \mid \mathbf{x}, \mathbf{w})$ is given by

$$p(y \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}\boldsymbol{\phi}(\mathbf{x}))^y (1 - \sigma(\mathbf{w}\boldsymbol{\phi}(\mathbf{x})))^{1-y}.$$

Therefore, the decision boundary is no longer (necessarily) a hyperplane on the original space.

In what follows, we assume that the observations were already transformed by a basis function. It is advisable to at least prefix each original observation $\mathbf{x}$ by the constant 1, leading to a transformed observation $\mathbf{x}' = (1, \mathbf{x})$. In this case, the decision boundary on the original space becomes an affine hyperplane $S = \{\mathbf{x} \in \mathbb{R}^D \mid w_0 + \mathbf{w}_{1:D}\mathbf{x} = 0\}$.

The (conditional) likelihood $p(\mathcal{D} \mid \mathbf{w})$ of the dataset $\mathcal{D}$ is given by

$$p(\mathcal{D} \mid \mathbf{w}) = \prod_{i=1}^N p(y_i \mid \mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^N \sigma(\mathbf{w}\mathbf{x}_i)^{y_i} (1 - \sigma(\mathbf{w}\mathbf{x}_i))^{1-y_i},$$

for a choice of $\mathbf{w}$. Notice that this likelihood is not based on the joint density over observations and targets, which is irrelevant for our purposes. Instead, $p(\mathcal{D} \mid \mathbf{w})$ is the probability associated to the particular assignment $\mathbf{y} = (y_1, \ldots, y_N)$ when the observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are seen as constants (according to $\mathcal{D}$), and $Y_i \perp\!\!\!\perp \mathbf{X}_{-i}, Y_{-i} \mid \mathbf{X}_i, \mathbf{W}$.

The log-likelihood $\ell(\mathbf{w}) = \log p(\mathcal{D} \mid \mathbf{w})$ is given by

$$\begin{aligned}
\ell(\mathbf{w}) &= \log \prod_{i=1}^N \sigma(\mathbf{w}\mathbf{x}_i)^{y_i} (1 - \sigma(\mathbf{w}\mathbf{x}_i))^{1-y_i} \\
&= \sum_{i=1}^N \log\left[\sigma(\mathbf{w}\mathbf{x}_i)^{y_i}\right] + \log\left[(1 - \sigma(\mathbf{w}\mathbf{x}_i))^{1-y_i}\right] \\
&= \sum_{i=1}^N y_i \log \sigma(\mathbf{w}\mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\mathbf{w}\mathbf{x}_i)).
\end{aligned}$$

It can be shown that there is no analytical expression for the maximum (log-)likelihood estimate $\hat{\mathbf{w}}$ for a logistic regression model.

For any k, the partial derivative $\partial \ell(\mathbf{w})/\partial w_k$ of ℓ at $\mathbf{w}$ with respect to the variable w_k is given by

$$
\begin{aligned}
\frac{\partial \ell(\mathbf{w})}{\partial w_k} &= \sum_{i=1}^{N} y_i \frac{\partial}{\partial w_k}\big[\log \sigma(\mathbf{w}\mathbf{x}_i)\big] + (1-y_i)\frac{\partial}{\partial w_k}\big[\log(1-\sigma(\mathbf{w}\mathbf{x}_i))\big]\\
&= \sum_{i=1}^{N} y_i \frac{1}{\sigma(\mathbf{w}\mathbf{x}_i)}\frac{\partial}{\partial w_k}\big[\sigma(\mathbf{w}\mathbf{x}_i)\big] + (1-y_i)\frac{1}{1-\sigma(\mathbf{w}\mathbf{x}_i)}\frac{\partial}{\partial w_k}\big[1-\sigma(\mathbf{w}\mathbf{x}_i)\big]\\
&= \sum_{i=1}^{N} y_i \frac{1}{\sigma(\mathbf{w}\mathbf{x}_i)}\sigma(\mathbf{w}\mathbf{x}_i)(1-\sigma(\mathbf{w}\mathbf{x}_i))\frac{\partial}{\partial w_k}\big[\mathbf{w}\mathbf{x}_i\big] + (1-y_i)\frac{1}{1-\sigma(\mathbf{w}\mathbf{x}_i)}\frac{\partial}{\partial w_k}\big[1-\sigma(\mathbf{w}\mathbf{x}_i)\big],\\
&= \sum_{i=1}^{N} y_i(1-\sigma(\mathbf{w}\mathbf{x}_i))x_{i,k} - (1-y_i)\sigma(\mathbf{w}\mathbf{x}_i)x_{i,k}\\
&= \sum_{i=1}^{N} (y_i - \sigma(\mathbf{w}\mathbf{x}_i))x_{i,k}.
\end{aligned}
$$

Thus, the gradient of ℓ with respect to $\mathbf{w}$ is given by

$$
\nabla_{\mathbf{w}}\ell(\mathbf{w}) = \mathbf{X}^T(\mathbf{y} - \sigma(\mathbf{X}\mathbf{w})),
$$

where σ is applied element-wise, $\mathbf{X}$ is the design matrix (where each observation corresponds to a row), and $\mathbf{y} = (y_1, \ldots, y_N)$.

For any j and k, the second order partial derivative $\partial^2 \ell(\mathbf{w})/\partial w_j \partial w_k$ at $\mathbf{w}$ is given by

$$
\begin{aligned}
\frac{\partial^2 \ell(\mathbf{w})}{\partial w_j \partial w_k} &= \sum_{i=1}^{N} \frac{\partial}{\partial w_j}\big[(y_i - \sigma(\mathbf{w}\mathbf{x}_i))x_{i,k}\big] = \sum_{i=1}^{N} \frac{\partial}{\partial w_j}\big[-\sigma(\mathbf{w}\mathbf{x}_i)x_{i,k}\big] = -\sum_{i=1}^{N} x_{i,k}\frac{\partial}{\partial w_j}\big[\sigma(\mathbf{w}\mathbf{x}_i)\big]\\
&= -\sum_{i=1}^{N} x_{i,k}x_{i,j}\sigma'(\mathbf{w}\mathbf{x}_i) = -\sum_{i=1}^{N} x_{i,k}x_{i,j}\sigma(\mathbf{w}\mathbf{x}_i)(1-\sigma(\mathbf{w}\mathbf{x}_i)).
\end{aligned}
$$

Recall that the Hessian matrix $\nabla^2 \ell(\mathbf{w})$ of ℓ at $\mathbf{w}$ is given by $\nabla^2 \ell(\mathbf{w})_{j,k} = \partial^2 \ell(\mathbf{w})/\partial w_j \partial w_k$. Using the result above, it can easily be shown that

$$
\nabla^2 \ell(\mathbf{w}) = -\sum_{i=1}^{N} \sigma(\mathbf{w}\mathbf{x}_i)(1-\sigma(\mathbf{w}\mathbf{x}_i))\mathbf{x}_i\mathbf{x}_i^T = -\mathbf{X}^T\mathbf{S}\mathbf{X},
$$

where $\mathbf{X}$ is the design matrix (where each observation corresponds to a row), and $\mathbf{S}$ is given by

$$
\mathbf{S} = \mathrm{diag}(\sigma(\mathbf{w}\mathbf{x}_1)(1-\sigma(\mathbf{w}\mathbf{x}_1)), \ldots, \sigma(\mathbf{w}\mathbf{x}_N)(1-\sigma(\mathbf{w}\mathbf{x}_N))).
$$

Because $\nabla^2 \ell(\mathbf{w})$ is negative definite for any $\mathbf{w}$, the log-likelihood ℓ is concave, and thus has at most one local maximum, which would necessarily be a global maximum.

Many iterative optimization methods may be employed to maximize such log-likelihood ℓ. We present some minimization methods in Sec. 2.4, which may be applied to minimize the negative log-likelihood $-\ell$.

However, the resulting estimate may have extremely large coefficients, particularly when the data is linearly separable. Alternatively, consider the task of finding a maximum a posteriori estimate. As before, the maximum a posteriori estimate $\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathcal{D})$ also maximizes $\log p(\mathcal{D} \mid \mathbf{w}) + \log p(\mathbf{w})$.

As in the previous section, suppose the prior density $p(\mathbf{w})$ associated to the weight vector $\mathbf{w}$ is given by

$$
p(\mathbf{w}) = \prod_{j=1}^{D} \mathcal{N}(w_j \mid 0, \tau^2).
$$

We have already seen that

$$
\log p(\mathbf{w}) = -\frac{1}{2\tau^2}\sum_{j=1}^{D} w_j^2 - \frac{D}{2}\log 2\pi\tau^2.
$$

Therefore,

$$\log p(\mathcal{D} \mid \mathbf{w}) + \log p(\mathbf{w}) = \ell(\mathbf{w}) - \frac{1}{2\tau^2} \sum_{j=1}^{D} w_j^2 - \frac{D}{2} \log 2\pi\tau^2.$$

Eliminating the constants with respect to $\mathbf{w}$ leads to the task of maximizing the regularized log-likelihood ℓ_λ given by

$$\ell_\lambda(\mathbf{w}) = \ell(\mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||^2,$$

where $\lambda = 1/\tau^2$. It is easy to show that

$$\nabla \ell_\lambda(\mathbf{w}) = \nabla \ell(\mathbf{w}) - \lambda \mathbf{w}$$
$$\nabla^2 \ell_\lambda(\mathbf{w}) = \nabla^2 \ell(\mathbf{w}) - \lambda \mathbf{I},$$

where $\mathbf{I}$ is the $D \times D$ identity matrix.

Note that if each observation was prefixed by the constant 1, the corresponding weight should not be penalized by λ, as already explained in the previous section.

Logistic regression can be generalized to multi-class problems. In that case, the technique assumes that the probability $p(y \mid \mathbf{x}, \mathbf{w})$ of class $y \in \{1, \ldots, C\}$ given observation $\mathbf{x} \in \mathbb{R}^D$ and parameter vector $\mathbf{w}$ is given by

$$p(y \mid \mathbf{x}, \mathbf{w}) = \frac{\exp(\mathbf{w}_y \mathbf{x})}{\sum_{y'} \exp(\mathbf{w}_{y'} \mathbf{x})},$$

where $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_C) \in \mathbb{R}^{CD}$ contains a parameter vector $\mathbf{w}_y$ corresponding to each class y.

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$ for some unknown $\boldsymbol{\theta}^*$, such that $\mathbf{x}_i \in \mathbb{R}^D$, and $\mathbf{y}_i \in \{0,1\}^C$. Suppose also that each observation belongs to a single class, and $y_{i,c} = 1$ if and only if $\mathbf{x}_i$ belongs to class c. This is a one-of-C encoding, which will simplify our presentation.

The (conditional) likelihood $p(\mathcal{D} \mid \mathbf{w})$ of $\mathcal{D}$ under $\mathbf{w}$ is given by

$$p(\mathcal{D} \mid \mathbf{w}) = \prod_{i=1}^{N} p(\mathbf{y}_i \mid \mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{N} \prod_{j=1}^{C} z_{i,j}^{y_{i,j}},$$

where

$$z_{i,j} = \frac{\exp(\mathbf{w}_j \mathbf{x}_i)}{\sum_{j'} \exp(\mathbf{w}_{j'} \mathbf{x}_i)}.$$

Therefore, the log-likelihood $\ell = \log p(\mathcal{D} \mid \mathbf{w})$ is given by

$$\ell = \log p(\mathcal{D} \mid \mathbf{w}) = \log \prod_{i=1}^{N} \prod_{j=1}^{C} z_{i,j}^{y_{i,j}} = \sum_{i=1}^{N} \sum_{j=1}^{C} y_{i,j} \log z_{i,j}.$$

Once again, the methods presented in Sec. 2.4 may be employed to minimize the negative log-likelihood $-\ell$ given the gradient and Hessian of $-\ell$ with respect to $\mathbf{w}$.

The partial derivative $\partial \ell / \partial w_{k,l}$ of ℓ with respect to parameter $w_{k,l}$ is given by

$$\frac{\partial \ell}{\partial w_{k,l}} = \frac{\partial}{\partial w_{k,l}} \left[\sum_{i=1}^{N} \sum_{j=1}^{C} y_{i,j} \log z_{i,j} \right] = \sum_{i=1}^{N} \sum_{j=1}^{C} \frac{y_{i,j}}{z_{i,j}} \frac{\partial z_{i,j}}{\partial w_{k,l}},$$

where the second equality follows from the chain rule.

Let $a_{i,j} = \mathbf{w}_j \mathbf{x}_i$, and notice that $w_{k,l}$ only affects $z_{i,j}$ through $a_{i,k}$. By the chain rule,

$$\frac{\partial \ell}{\partial w_{k,l}} = \sum_{i=1}^{N} \sum_{j=1}^{C} \frac{y_{i,j}}{z_{i,j}} \frac{\partial z_{i,j}}{\partial a_{i,k}} \frac{\partial a_{i,k}}{\partial w_{k,l}} = \sum_{i=1}^{N} \sum_{j=1}^{C} \frac{y_{i,j} x_{i,l}}{z_{i,j}} \frac{\partial z_{i,j}}{\partial a_{i,k}}.$$

By the definition of $z_{i,j}$,

$$\frac{\partial z_{i,j}}{\partial a_{i,k}} = \frac{\partial}{\partial a_{i,k}} \left[\frac{\exp(a_{i,j})}{\sum_{j'} \exp(a_{i,j'})} \right].$$

Firstly, suppose $j = k$. By the quotient rule,

$$\frac{\partial z_{i,j}}{\partial a_{i,j}} = \frac{\partial}{\partial a_{i,j}} \left[\frac{\exp(a_{i,j})}{\sum_{j'} \exp(a_{i,j'})} \right] = \frac{\left[\sum_{j'} \exp(a_{i,j'})\right] \exp(a_{i,j}) - \exp(a_{i,j}) \exp(a_{i,j})}{\left[\sum_{j'} \exp(a_{i,j'})\right]^2} = z_{i,j} - z_{i,j}^2.$$

Alternatively, suppose $j \neq k$. In that case,

$$\frac{\partial z_{i,j}}{\partial a_{i,k}} = \frac{\partial}{\partial a_{i,k}} \left[\frac{\exp(a_{i,j})}{\sum_{j'} \exp(a_{i,j'})} \right] = -\frac{\exp(a_{i,j}) \exp(a_{i,k})}{\left[\sum_{j'} \exp(a_{i,j'})\right]^2} = -z_{i,j} z_{i,k}.$$

Therefore, for any j and k,

$$\frac{\partial z_{i,j}}{\partial a_{i,k}} = z_{i,j}(\mathbb{I}\,[j = k] - z_{i,k}),$$

where $\mathbb{I}$ is the indicator function.

Therefore, $\partial \ell / \partial w_{k,l}$ is given by

$$\frac{\partial \ell}{\partial w_{k,l}} = \sum_{i=1}^{N} \sum_{j=1}^{C} \frac{y_{i,j} x_{i,l}}{z_{i,j}} \frac{\partial z_{i,j}}{\partial a_{i,k}} = \sum_{i=1}^{N} x_{i,l} \sum_{j=1}^{C} y_{i,j}(\mathbb{I}\,[j = k] - z_{i,k})$$

$$= \sum_{i=1}^{N} x_{i,l} \left[\sum_{j=1}^{C} y_{i,j} \mathbb{I}\,[j = k] - \sum_{j=1}^{C} y_{i,j} z_{i,k} \right] = \sum_{i=1}^{N} x_{i,l}(y_{i,k} - z_{i,k}),$$

where we used the fact that $\sum_j y_{i,j} = 1$, for any i.

Using this result, the gradient $\nabla_{\mathbf{w}_k} \ell$ of ℓ with respect to the weight vector $\mathbf{w}_k$ associated to class k is given by

$$\nabla_{\mathbf{w}_k} \ell = \sum_{i=1}^{N} (y_{i,k} - z_{i,k}) \mathbf{x}_i.$$

Using a similar argument, the second order partial derivative $\partial^2 \ell / \partial w_{m,n} \partial w_{k,l}$ is given by

$$\frac{\partial^2 \ell}{\partial w_{m,n} \partial w_{k,l}} = \frac{\partial}{\partial w_{m,n}} \left[\sum_{i=1}^{N} x_{i,l} y_{i,k} - x_{i,l} z_{i,k} \right] = -\sum_{i=1}^{N} x_{i,l} \frac{\partial z_{i,k}}{\partial w_{m,n}}$$

$$= -\sum_{i=1}^{N} x_{i,l} \frac{\partial z_{i,k}}{\partial a_{i,m}} \frac{\partial a_{i,m}}{\partial w_{m,n}} = -\sum_{i=1}^{N} x_{i,l} x_{i,n} z_{i,k}(\mathbb{I}[k = m] - z_{i,m}).$$

It is straightforward to show that the corresponding (negative definite) Hessian matrix $\nabla_{\mathbf{w}}^2 \ell$ is a $CD \times CD$ matrix composed of $D \times D$ blocks, such that block j, k is given by

$$-\sum_{i=1}^{N} z_{i,k}(\mathbb{I}[k = j] - z_{i,j}) \mathbf{x}_i \mathbf{x}_i^T.$$

Maximum a posteriori estimation is analogous to the binary case.

Consider the task of computing the posterior density $p(\mathbf{w} \mid \mathcal{D}) \propto_{\mathbf{w}} p(\mathcal{D} \mid \mathbf{w}) p(\mathbf{w})$ associated to weight vector $\mathbf{w}$ given $\mathcal{D}$ according to a binary logistic regression model. In contrast to linear regression, there is no convenient conjugate prior.

Instead of computing the exact posterior, we will compute its Laplace approximation. Firstly, recall that the desired posterior is given by

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w}) p(\mathbf{w})}{p(\mathcal{D})} = \frac{p(\mathcal{D}, \mathbf{w})}{p(\mathcal{D})}.$$

By the assumptions made by binary logistic regression, we know that $p(\mathcal{D}, \mathbf{w}) > 0$, for any $\mathcal{D}$ and $\mathbf{w}$. Therefore, the desired posterior may be rewritten as

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{e^{-E(\mathbf{w})}}{p(\mathcal{D})},$$

where E is the so-called energy function given by $E(\mathbf{w}) = -\log p(\mathcal{D}, \mathbf{w}) = -\log p(\mathcal{D} \mid \mathbf{w}) - \log p(\mathbf{w})$. Recall that we already know the first and second order partial derivatives required to minimize $E(\mathbf{w})$ with respect to $\mathbf{w}$ considering a multivariate Gaussian prior given by $p(\mathbf{w}) = \prod_{j=1}^{D} \mathcal{N}(w_j \mid 0, \tau^2)$.

Let $\mathbf{w}^*$ denote a maximum a posteriori (minimum energy) estimate such that $E(\mathbf{w}^*) = \min_{\mathbf{w}} E(\mathbf{w}) = \max_{\mathbf{w}} p(\mathbf{w} \mid \mathcal{D})$, and consider the quadratic approximation $\hat{E}$ to E at point $\mathbf{w}^*$, which is given by

$$\hat{E}(\mathbf{w}) = E(\mathbf{w}^*) + (\mathbf{w} - \mathbf{w}^*)\nabla E(\mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \nabla^2 E(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*).$$

Because $\mathbf{w}^*$ is a local minimum of the twice differentiable function E (by construction), we know that $\nabla E(\mathbf{w}^*) = \mathbf{0}$. Therefore,

$$\hat{E}(\mathbf{w}) = E(\mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \nabla^2 E(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*).$$

The corresponding approximate posterior is given by

$$p(\mathbf{w} \mid \mathcal{D}) \propto_{\mathbf{w}} e^{-E(\mathbf{w})} \approx e^{-\hat{E}(\mathbf{w})}.$$

The appropriate normalization constant Z, also called Laplace approximation to the marginal likelihood, is given by

$$Z = \int_{\text{Val}(\mathbf{W})} e^{-\hat{E}(\mathbf{w})} d\mathbf{w} = \int_{\text{Val}(\mathbf{W})} e^{-E(\mathbf{w}^*)} e^{-\frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^T \nabla^2 E(\mathbf{w}^*)(\mathbf{w}-\mathbf{w}^*)} d\mathbf{w}$$

$$= e^{-E(\mathbf{w}^*)} \int_{\text{Val}(\mathbf{W})} e^{-\frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^T \nabla^2 E(\mathbf{w}^*)(\mathbf{w}-\mathbf{w}^*)} d\mathbf{w} = e^{-E(\mathbf{w}^*)} \sqrt{(2\pi)^D |\nabla^2 E(\mathbf{w}^*)|^{-1}},$$

by assuming that $\nabla^2 E(\mathbf{w}^*)$ is positive definite, and recognizing the last integral as the normalization constant of a multivariate Gaussian distribution.

Finally, the desired approximate posterior is given by

$$p(\mathbf{w} \mid \mathcal{D}) \approx \frac{e^{-\hat{E}(\mathbf{w})}}{Z} = \frac{e^{-E(\mathbf{w}^*)} e^{-\frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^T \nabla^2 E(\mathbf{w}^*)(\mathbf{w}-\mathbf{w}^*)}}{e^{-E(\mathbf{w}^*)} \sqrt{(2\pi)^D |\nabla^2 E(\mathbf{w}^*)|^{-1}}} = \mathcal{N}(\mathbf{w} \mid \mathbf{w}^*, [\nabla^2 E(\mathbf{w}^*)]^{-1}).$$

In other words, the Laplace approximation to the posterior distribution is a multivariate Gaussian distribution whose mean is the maximum a posteriori estimate and whose covariance matrix is the inverse of the Hessian of the energy function at that estimate.

The posterior predictive probability $p(y \mid \mathbf{x}, \mathcal{D})$ of class y given observation $\mathbf{x}$ and dataset $\mathcal{D}$ is given by

$$p(y \mid \mathbf{x}, \mathcal{D}) = \int_{\text{Val}(\mathbf{W})} p(y, \mathbf{w} \mid \mathbf{x}, \mathcal{D}) d\mathbf{w} = \int_{\text{Val}(\mathbf{W})} p(y \mid \mathbf{w}, \mathbf{x}, \mathcal{D}) p(\mathbf{w} \mid \mathbf{x}, \mathcal{D}) d\mathbf{w} = \int_{\text{Val}(\mathbf{W})} p(y \mid \mathbf{w}, \mathbf{x}) p(\mathbf{w} \mid \mathcal{D}) d\mathbf{w},$$

by using the facts that $Y \perp\!\!\!\perp \mathcal{D} \mid \mathbf{W}, \mathbf{X}$, and $\mathbf{W} \perp\!\!\!\perp \mathbf{X} \mid \mathcal{D}$. In other words, the posterior predictive probability is an expected value with respect to the posterior distribution.

We may combine the Laplace approximation with a Monte Carlo approximation to obtain an approximate posterior predictive probability. Concretely, we may approximate $p(y \mid \mathbf{x}, \mathcal{D})$ by

$$p(y \mid \mathbf{x}, \mathcal{D}) \approx \int_{\text{Val}(\mathbf{W})} p(y \mid \mathbf{w}, \mathbf{x}) \mathcal{N}(\mathbf{w} \mid \mathbf{w}^*, [\nabla^2 E(\mathbf{w}^*)]^{-1}) d\mathbf{w} \approx \frac{1}{S} \sum_{s=1}^{S} \sigma(\mathbf{w}_s \mathbf{x})^y (1 - \sigma(\mathbf{w}_s \mathbf{x}))^{1-y},$$

where $\mathbf{w}_1, \ldots, \mathbf{w}_S$ is an iid dataset distributed according to $\mathcal{N}(\cdot \mid \mathbf{w}^*, [\nabla^2 E(\mathbf{w}^*)]^{-1})$, which may be obtained by standard methods.

9 Exponential family

A probability density or mass function $p(\cdot \mid \boldsymbol{\theta})$ belongs to the exponential family if it can be written as

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \frac{h(\mathbf{x})\exp(\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\phi}(\mathbf{x}))}{Z(\boldsymbol{\theta})} = h(\mathbf{x})\exp\left(\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\phi}(\mathbf{x}) - A(\boldsymbol{\theta})\right),$$

for every assignment $\mathbf{x} \in \mathrm{Val}(\mathbf{X})$, for some choice of functions $h : \mathrm{Val}(\mathbf{X}) \to \mathbb{R}_{>0}$, $\boldsymbol{\eta} : \mathrm{Val}(\boldsymbol{\Theta}) \to \mathbb{R}^D$, and $\boldsymbol{\phi} : \mathrm{Val}(\mathbf{X}) \to \mathbb{R}^D$. If $\eta(\boldsymbol{\theta}) = \boldsymbol{\theta}$ for all $\boldsymbol{\theta}$, the distribution is said to be in canonical form.

Furthermore, the log-partition function A is given by $A(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta})$, and the partition function Z must be given by either

$$Z(\boldsymbol{\theta}) = \int_{\mathrm{Val}(\mathbf{X})} h(\mathbf{x})\exp(\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\phi}(\mathbf{x}))\ d\mathbf{x}$$

or

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x} \in \mathrm{Val}(\mathbf{X})} h(\mathbf{x})\exp(\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\phi}(\mathbf{x})),$$

depending on whether $p(\cdot \mid \boldsymbol{\theta})$ is a density or mass function, respectively.

As an example, consider the Bernoulli pmf $\mathrm{Ber}(\cdot \mid \theta)$, which is defined on $\{0,1\}$ as $\mathrm{Ber}(x \mid \theta) = \theta^x(1-\theta)^{1-x}$. Assuming $\theta \in (0,1)$,

$$\mathrm{Ber}(x \mid \theta) = \exp\left(\log\left(\theta^x(1-\theta)^{1-x}\right)\right) = \exp(x \log\theta + (1-x)\log(1-\theta)).$$

By letting $h(x) = 1$, $\boldsymbol{\phi}(x) = (x, 1-x)$, $\boldsymbol{\eta}(\theta) = (\log\theta, \log(1-\theta))$, and $Z(\theta) = 1$, we see that the Bernoulli pmf $\mathrm{Ber}(\cdot \mid \theta)$ is in the exponential family, for any $\theta \in (0,1)$.

Alternatively, by letting $h(x) = 1$, $\phi(x) = x$, $\eta(\theta) = \log\frac{\theta}{1-\theta}$, and $Z(\theta) = \frac{1}{1-\theta}$, it is easy to inspect that

$$\mathrm{Ber}(x \mid \theta) = (1-\theta)\exp\left(x\log\left(\frac{\theta}{1-\theta}\right)\right),$$

for $x \in \{0,1\}$ and $\theta \in (0,1)$. In this case, the Bernoulli parameter θ is a sigmoid function of $\eta(\theta)$, since

$$\eta(\theta) = \log\frac{\theta}{1-\theta} \implies e^{\eta(\theta)} = \frac{\theta}{1-\theta} \implies \theta = \frac{e^{\eta(\theta)}}{1+e^{\eta(\theta)}} = \frac{1}{1+e^{-\eta(\theta)}}.$$

As another example, consider the Gaussian pdf $\mathcal{N}(\cdot \mid \mu, \sigma^2)$. By letting $\boldsymbol{\theta} = (\mu, \sigma^2)$, $h(x) = 1$, $\boldsymbol{\phi}(x) = (x, x^2)$, $\boldsymbol{\eta}(\boldsymbol{\theta}) = (\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2})$, and $Z(\boldsymbol{\theta}) = \sqrt{2\pi\sigma^2}\exp\left(\frac{\mu^2}{2\sigma^2}\right)$,

$$\frac{\exp(\frac{x\mu}{\sigma^2} - \frac{x^2}{2\sigma^2})}{\sqrt{2\pi\sigma^2}\exp\left(\frac{\mu^2}{2\sigma^2}\right)} = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{x\mu}{\sigma^2} - \frac{x^2}{2\sigma^2} - \frac{\mu^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \mathcal{N}(x \mid \mu, \sigma^2).$$

Therefore, the Gaussian pdf $\mathcal{N}(\cdot \mid \mu, \sigma^2)$ is in the exponential family for any μ and $\sigma^2 > 0$.

The exponential family also includes the gamma, beta, Dirichlet, categorical, and Poisson distributions.

Consider a univariate probability density function $p(\cdot \mid \theta)$ in canonical form given by

$$p(x \mid \theta) = \frac{h(x)\exp\left(\theta\phi(x)\right)}{Z(\theta)},$$

and suppose the partition function Z is differentiable. By the chain rule,

$$\frac{d}{d\theta}A(\theta) = \frac{d}{d\theta}\log Z(\theta) = \frac{d}{d\theta}\log\left[\int_{\mathrm{Val}(X)} h(x)\exp\left(\theta\phi(x)\right)\ dx\right] = \frac{1}{Z(\theta)}\frac{d}{d\theta}\int_{\mathrm{Val}(X)} h(x)\exp\left(\theta\phi(x)\right)\ dx.$$

Furthermore, suppose the function f given by $f(x,\theta) = h(x)\exp\left(\theta\phi(x)\right)$ is differentiable with respect to θ, and that both f and $\partial f/\partial\theta$ are continuous in $\mathrm{Val}(X) \times \mathrm{Val}(\Theta)$. In that case, the Leibniz integral rule states that

$$\frac{d}{d\theta}\int_{\mathrm{Val}(X)} h(x)\exp\left(\theta\phi(x)\right)\ dx = \int_{\mathrm{Val}(X)} \frac{d}{d\theta}\left[h(x)\exp\left(\theta\phi(x)\right)\right]\ dx.$$

By the chain rule and the definition of $p(x \mid \theta)$,

$$\frac{d}{d\theta} A(\theta) = \frac{1}{Z(\theta)} \int_{\text{Val}(X)} h(x) \exp\left(\theta\phi(x)\right) \phi(x) \, dx = \int_{\text{Val}(X)} p(x \mid \theta)\phi(x) \, dx = \mathbb{E}_{p(X|\theta)}[\phi(X)].$$

Using the assumptions outlined above, the second derivative of A with respect to θ is given by

$$\frac{d^2}{d\theta^2} A(\theta) = \frac{d}{d\theta} \int_{\text{Val}(X)} h(x) \exp\left(\theta\phi(x) - A(\theta)\right) \phi(x) \, dx.$$

Supposing that the function f given by $f(x, \theta) = h(x) \exp\left(\theta\phi(x) - A(\theta)\right) \phi(x)$ is differentiable with respect to θ, and that both f and $\partial f / \partial \theta$ are continuous in $\text{Val}(X) \times \text{Val}(\Theta)$, the Leibniz integral rule states that

$$\begin{aligned}
\frac{d^2}{d\theta^2} A(\theta) &= \int_{\text{Val}(X)} \frac{d}{d\theta} \left[h(x) \exp\left(\theta\phi(x) - A(\theta)\right) \phi(x) \right] \, dx \\
&= \int_{\text{Val}(X)} h(x)\phi(x) \exp\left(\theta\phi(x) - A(\theta)\right) \left[\phi(x) - \frac{d}{d\theta} A(\theta) \right] \, dx \\
&= \int_{\text{Val}(X)} h(x)\phi(x) \exp\left(\theta\phi(x) - A(\theta)\right) \left[\phi(x) - \mathbb{E}_{p(X|\theta)}[\phi(X)] \right] \, dx \\
&= \left[\int_{\text{Val}(X)} \phi(x)^2 p(x \mid \theta) \, dx \right] - \mathbb{E}_{p(X|\theta)}[\phi(X)] \int_{\text{Val}(X)} h(x)\phi(x) \exp\left(\theta\phi(x) - A(\theta)\right) \, dx \\
&= \mathbb{E}_{p(X|\theta)}[\phi(X)^2] - \mathbb{E}_{p(X|\theta)}[\phi(X)]^2 = \text{var}_{p(X|\theta)}[\phi(X)].
\end{aligned}$$

The analogous result for a well-behaved multivariate probability density function $p(\cdot \mid \boldsymbol{\theta})$ in canonical form states that the gradient $\nabla A(\boldsymbol{\theta})$ of A at $\boldsymbol{\theta}$ is given by $\nabla A(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{X}|\boldsymbol{\theta})}[\boldsymbol{\phi}(\mathbf{X})]$, and that the Hessian matrix $\nabla^2 A(\boldsymbol{\theta})$ of A at $\boldsymbol{\theta}$ is given by $\nabla^2 A(\boldsymbol{\theta}) = \text{cov}_{p(\mathbf{X}|\boldsymbol{\theta})}[\boldsymbol{\phi}(\mathbf{X})]$. Because a covariance matrix is always positive semidefinite, the log-partition function A of such well-behaved multivariate probability density function is convex.

Consider an iid dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$ distributed according to $p(\cdot \mid \boldsymbol{\theta})$, which is in the exponential family. The corresponding likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ is given by

$$\begin{aligned}
p(\mathcal{D} \mid \boldsymbol{\theta}) &= \prod_{i=1}^{N} p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} \frac{h(\mathbf{x}_i) \exp(\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\phi}(\mathbf{x}_i))}{Z(\boldsymbol{\theta})} = \frac{1}{Z(\boldsymbol{\theta})^N} \left[\prod_{i=1}^{N} h(\mathbf{x}_i) \right] \left[\prod_{i=1}^{N} \exp(\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\phi}(\mathbf{x}_i)) \right] \\
&= \frac{1}{Z(\boldsymbol{\theta})^N} \left[\prod_{i=1}^{N} h(\mathbf{x}_i) \right] \exp\left(\sum_{i=1}^{N} \boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\phi}(\mathbf{x}_i) \right) = \frac{1}{Z(\boldsymbol{\theta})^N} \left[\prod_{i=1}^{N} h(\mathbf{x}_i) \right] \exp\left(\boldsymbol{\eta}(\boldsymbol{\theta}) \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i) \right).
\end{aligned}$$

Notice that if the function h is constant and the number of observations N is known, the so-called sufficient statistics $\boldsymbol{\phi}(\mathcal{D}) = \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i)$ of the dataset $\mathcal{D}$ are sufficient to compute the likelihood under any parameter $\boldsymbol{\theta}$. This is a crucial property of distributions in the exponential family.

Now suppose that $p(\cdot \mid \boldsymbol{\theta})$ can be written in canonical form, and that $h(\mathbf{x}) = 1$, for all $\mathbf{x}$. In that case, the log-likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ is given by

$$\log p(\mathcal{D} \mid \boldsymbol{\theta}) = \log \left[\frac{1}{Z(\boldsymbol{\theta})^N} \exp\left(\boldsymbol{\theta} \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i) \right) \right] = \boldsymbol{\theta} \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i) - N \log Z(\boldsymbol{\theta}) = \boldsymbol{\theta}\boldsymbol{\phi}(\mathcal{D}) - NA(\boldsymbol{\theta}),$$

where $\boldsymbol{\phi}(\mathcal{D}) = \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i)$ are the sufficient statistics.

As already mentioned, the log-partition function A of a well-behaved probability density (or mass) function in canonical form is convex. Since $\boldsymbol{\theta}\boldsymbol{\phi}(\mathcal{D})$ is a linear function of $\boldsymbol{\theta}$, the log-likelihood given by $\log p(\mathcal{D} \mid \boldsymbol{\theta}) = \boldsymbol{\theta}\boldsymbol{\phi}(\mathcal{D}) - NA(\boldsymbol{\theta})$ is concave, and therefore has at most one global maximum.

Furthermore, the gradient $\nabla_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta})$ at $\boldsymbol{\theta}$ of such well-behaved probability density (or mass) function in canonical form is given by

$$\nabla_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \left[\boldsymbol{\theta}\boldsymbol{\phi}(\mathcal{D}) - NA(\boldsymbol{\theta}) \right] = \boldsymbol{\phi}(\mathcal{D}) - N\mathbb{E}_{p(\mathbf{X}|\boldsymbol{\theta})}[\boldsymbol{\phi}(\mathbf{X})].$$

Therefore, if the parameter $\boldsymbol{\theta}$ is an (interior) local maximum, which would necessarily be the maximum likelihood estimate, then

$$\mathbb{E}_{p(\mathbf{X}|\boldsymbol{\theta})}[\boldsymbol{\phi}(\mathbf{X})] = \frac{1}{N} \boldsymbol{\phi}(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i).$$

In other words, if the parameter $\boldsymbol{\theta}$ is an (interior) local maximum, the expected value of the sufficient statistics given $\boldsymbol{\theta}$ must match the empirical mean of the sufficient statistics.

A generalized linear model is a regression/classification model that assumes that the probability density/mass function associated to the target variable Y given an observation $\mathbf{x}$ and a parameter vector $\mathbf{w}$ is in the exponential family. Furthermore, the conditional expected value of Y must be given by $\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}, \mathbf{W} = \mathbf{w}] = f(\mathbf{wx})$, where f is a so-called activation function, whose inverse f^{-1} is called a link function.

Linear regression is a generalized linear model whose activation function f is the identity, since the probability density associated to target $y \in \mathbb{R}$ given observation $\mathbf{x} \in \mathbb{R}^D$ and parameter vector $\mathbf{w}$ is given by

$$p(y \mid \mathbf{x}, \mathbf{w}) = N(y \mid \mathbf{wx}, \sigma^2),$$

assuming a known variance σ^2.

Logistic regression is a generalized linear model whose activation function f is sigmoid, since the probability associated to target $y \in \{0, 1\}$ given observation $\mathbf{x} \in \mathbb{R}^D$ and parameter vector $\mathbf{w}$ is given by

$$p(y \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{wx})^y (1 - \sigma(\mathbf{wx}))^{1-y} = \mathrm{Ber}(y \mid \sigma(\mathbf{wx})).$$

10 Bayesian networks

This section briefly presents Bayesian networks, which are probabilistic graphical models. For more details, see the corresponding notes.

Consider the task of representing a joint probability mass function p for a random vector $\mathbf{X} = (X_1, \ldots, X_D)$. By the chain rule of probability, for any assignment $\mathbf{x} \in \mathrm{Val}(\mathbf{X})$,

$$p(\mathbf{x}) = p(x_1, \ldots, x_D) = p(x_1) \prod_{j=2}^{D} p(x_j \mid x_1, \ldots, x_{j-1}).$$

Suppose each discrete random variable in $\mathbf{X}$ has exactly K valid assignments. In that case, the conditional probability mass function for X_j given $X_1, \ldots, X_{j-1}$ may be represented by a table with $(K-1)K^{j-1}$ elements. Therefore, the total number of free parameters needed to represent the full joint p is

$$\sum_{j=1}^{D} (K-1)K^{j-1} = (K-1)\frac{1 - K^D}{1 - K} = K^D - 1.$$

Because the number of free parameters grows exponentially with the number of variables D, learning requires a large amount of data, unless independence assumptions are made.

A Bayesian network (G, p) represents independence statements about a joint probability mass/density function p for a random vector $\mathbf{X} = (X_1, \ldots, X_D)$ using a directed acyclic graph $G = (V, E)$, such that $V = \{X_1, \ldots, X_D\}$, and $E \subseteq V^2$. The joint probability function p is said to factorize over G, and is given by

$$p(\mathbf{x}) = p(x_1, \ldots, x_D) = \prod_{j=1}^{D} p(x_j \mid \mathrm{pa}_{X_j}),$$

for every $\mathbf{x} \in \mathrm{Val}(\mathbf{X})$, where pa_{X_j} is the assignment to the parents Pa_{X_j} of X_j in G.

In the case of a Bayesian network over discrete random variables with exactly K valid assignments, the conditional probability mass function for X_j given Pa_{X_j} may be represented by a table with $(K-1)K^{|\mathrm{Pa}_{X_j}|}$ elements. Therefore, independence assumptions may lead to a drastic decrease in the number of free parameters.

In a Bayesian network, every random variable is conditionally independent of its non-descendants given its parent variables. This is called the local Markov property. Every variable is also independent of every other given its Markov blanket: its children, parents and other parents of its children.

Different directed acyclic graphs may represent the same independence statements between random variables in a Bayesian network. Two directed acyclic graphs are Markov equivalent if they represent the same set of conditional independence statements. Let an immorality in a directed acyclic graph $G = (V, E)$ be defined as a set of distinct vertices $\{U, V, W\}$ such that $(U, V) \in E$, $(W, V) \in E$, $(U, W) \notin E$ and $(W, U) \notin E$. Two directed acyclic graphs are Markov equivalent if and only if they they have the same set of immoralities and the same underlying graph. The underlying (undirected) graph of a directed graph G is the graph obtained by replacing all edges of G with undirected edges.

The following definitions are useful to establish if a random variable X is independent of Y given Z in a Bayesian network.

Let $G = (V, E)$ be a directed acyclic graph, G' its underlying graph and P' a path on G'. A set of three distinct nodes $\{U, V, W\}$ in P' is a v-structure if $(U, V) \in E$ and $(W, V) \in E$ and these two edges are in P'. In this structure, V is called a collider. In simple terms, in a v-structure, a collider V has two distinct neighbors U and W in P' with edges ending in V in the corresponding directed acyclic graph.

An undirected path P is said to be active given the set of random variables $\mathcal{Z}$ if all non-colliders in P are not in $\mathcal{Z}$, and all colliders in P are in $\mathcal{Z}$ or have a descendant in $\mathcal{Z}$.

In a Bayesian network (G, p), if all undirected paths between random variables X and Y are inactive given the set of random variables $\mathcal{Z}$, then $X \perp\!\!\!\perp Y | \mathcal{Z}$. In such case, it is said that X and Y are d-separated by $\mathcal{Z}$, denoted d-sep$_G(X; Y | \mathcal{Z})$. The notation extends to sets of variables naturally. Random variables that are not d-separated are d-connected. It is important to note that d-connected variables may still be independent. However, variables that are d-separated by $\mathcal{Z}$ are always independent given $\mathcal{Z}$.

We let $\mathcal{I}(p)$ denote the set of independence statements of the form $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$ that hold according to the joint probability mass/density function p. If G is a graph in a Bayesian network, then $\mathcal{I}(G)$ denotes the set of statements of the form $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$ such that d-sep$_G(\mathcal{X}; \mathcal{Y} | \mathcal{Z})$. A graph is an I-map for a set of independence statements I if $\mathcal{I}(G) \subseteq I$. Therefore, if $B = (G, p)$ is a Bayesian network, G is an I-map for $\mathcal{I}(p)$. If $\mathcal{I}(G) = \mathcal{I}(p)$, then G is a perfect map for p.

Causal relationships may be useful for building models intuitively. In this case, a causal variable often has an edge ending at an effect variable. In causal models, causal reasoning is defined as computing the posterior probability of effects given causes. The posterior probability of causes given effects is called evidential reasoning. Intercausal reasoning combines causal and evidential reasoning. *Explaining away* is the term given to the intercausal reasoning pattern in that a cause explains an effect and, by consequence, lowers the probability of another cause. However, it must be clear that independencies in Bayesian networks are not (in general) causal. A model in which the edges are causally reversed may be equally valid to represent the same set of independencies.

We now present some examples of probabilistic models that are conveniently represented as Bayesian networks.

Consider an iid dataset $\mathcal{D} = x_1, \ldots, x_N$ distributed according to $p(\cdot \mid \theta)$. The corresponding joint density $p(x_1, \ldots, x_N, \theta)$ is given by

$$p(x_1, \ldots, x_N, \theta) = p(\theta)p(x_1, \ldots, x_N \mid \theta) = p(\theta) \prod_{i=1}^{N} p(x_i \mid \theta),$$

which may be represented by a Bayesian network (G, p) where $G = (V, E)$, $V = \{\Theta\} \cup \{X_1, \ldots, X_N\}$, and $E = \{(\Theta, X_i) \in V^2 \mid 1 \leq i \leq N\}$.

A naive Bayes classifier assumes that the joint density $p(\mathbf{x}, y)$ of observation $\mathbf{x}$ and class y is given by

$$p(\mathbf{x}, y) = p(y)p(\mathbf{x} \mid y) = p(y) \prod_{j=1}^{D} p(x_j \mid y), \tag{1}$$

which may be represented by a Bayesian network (G, p) where $G = (V, E)$, $V = \{Y\} \cup \{X_1, \ldots, X_D\}$, and $E = \{(Y, X_j) \mid 1 \leq j \leq D\}$.

A Markov chain is a probabilistic model defined over a sequence of random variables $X_1, \ldots, X_T$. Each random variable X_t typically represents the state of a process at time step t, and the model encodes the assumption that the *future* is independent of the *past* given the *present*. The density $p(x_1, \ldots, x_T)$ is given by

$$p(x_1, \ldots, x_T) = p(x_1) \prod_{t=1}^{T-1} p(x_{t+1} \mid x_t),$$

which may be represented by a Bayesian network (G, p) where $G = (V, E)$, $V = \{X_1, \ldots, X_T\}$, and $E = \{(X_t, X_{t+1}) \mid 1 \leq t \leq T - 1\}$.

A hidden Markov model is a probabilistic model defined over two sequences of random variables: $Z_1, \ldots, Z_T$, and $X_1, \ldots, X_T$. Each random variable Z_t typically represents the *underlying* state of a process at time step t, while X_t represents the *observed* state at time step t. The density $p(z_1, \ldots, z_T, x_1, \ldots, x_T)$ is given by

$$p(z_1, \ldots, z_T, x_1, \ldots, x_T) = p(z_1)p(x_1 \mid z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1})p(x_t \mid z_t),$$

which may be represented by a Bayesian network (G, p) where $G = (V, E)$, $V = \{X_1, \ldots, X_T\} \cup \{Z_1, \ldots, Z_T\}$, and $E = \{(Z_{t-1}, Z_t) \mid 2 \leq t \leq T\} \cup \{(Z_t, X_t) \mid 1 \leq t \leq T\}$.

Consider a Bayesian network (G, p) for a discrete random vector $\mathbf{X}$, and suppose $\mathbf{X}$ is partitioned into a query vector $\mathbf{X}_q$, a visible vector $\mathbf{X}_v$, and a nuisance vector $\mathbf{X}_n$. The task of computing the probability $p(\mathbf{x}_q \mid \mathbf{x}_v) = \sum_{\mathbf{x}_n} p(\mathbf{x}_q, \mathbf{x}_n \mid \mathbf{x}_v)$ for some assignment $\mathbf{x}_q, \mathbf{x}_v \in \mathrm{Val}(\mathbf{X}_q, \mathbf{X}_v)$ is called inference. The continuous random vector case is analogous. Inference may be performed by either exact or approximate algorithms, which are outside the scope of this text. For more details, see the corresponding notes.

Consider an iid dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$ distributed according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for an unknown $\boldsymbol{\theta}^*$. Furthermore, suppose $p(\cdot \mid \boldsymbol{\theta}^*)$ factorizes over a known graph G.

Suppose $p(\cdot \mid \boldsymbol{\theta})$ also factorizes over G. In that case, the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of $\mathcal{D}$ under $\boldsymbol{\theta}$ is given by

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} \prod_{j=1}^{D} p(x_{i,j} \mid \mathrm{pa}_{X_{i,j}}, \boldsymbol{\theta}).$$

This definition leads to the usual tasks of maximum likelihood and maximum a posteriori estimation. For more details on learning in Bayesian networks, see the corresponding notes.

11 Expectation maximization

Expectation maximization is a method that may be employed to estimate parameters associated to so-called hidden random variables, whose assignments are never directly observed.

There are two main reasons for estimating such parameters. Firstly, the task naturally appears in practical problems. Secondly, positing the existence of hidden random variables allows defining complex joint probability density functions for observed random variables using few parameters. For instance, it may be impossible to represent all independence statements on given set of (observed) variables by a Bayesian network without adding hidden variables.

Concretely, consider a dataset $\mathcal{D} = (\mathbf{x}_1, \mathbf{z}_1), \ldots, (\mathbf{x}_N, \mathbf{z}_N)$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for some unknown $\boldsymbol{\theta}^*$. Furthermore, suppose that each $\mathbf{x}_i$ is an assignment to a continuous random vector $\mathbf{X}_i$, and that each $\mathbf{z}_i$ is an assignment to a discrete random vector $\mathbf{Z}_i$. By our definition of iid, for every parameter $\boldsymbol{\theta} \in \mathrm{Val}(\boldsymbol{\Theta})$,

$$p(\mathbf{x}_1, \mathbf{z}_1, \ldots, \mathbf{x}_N, \mathbf{z}_N \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta}).$$

If $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$ are hidden random vectors, then the assignments $\mathbf{z}_1, \ldots, \mathbf{z}_N$ are unknown, and the (observed) likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of the partially observed dataset $\mathcal{D}$ given the parameter $\boldsymbol{\theta}$ is defined as

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\theta}).$$

Consider the task of finding a maximum likelihood or maximum a posteriori estimate given a partially observed dataset $\mathcal{D}$. By marginalization,

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\theta}) = \sum_{\mathbf{z}_1} \cdots \sum_{\mathbf{z}_N} \prod_{i=1}^{N} p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta}).$$

Assume that $p(\mathbf{x}_i \mid \boldsymbol{\theta}) > 0$, for every i and $\boldsymbol{\theta}$. In that case, the corresponding log-likelihood $\ell(\boldsymbol{\theta})$ is given by

$$\ell(\boldsymbol{\theta}) = \log p(\mathcal{D} \mid \boldsymbol{\theta}) = \log \prod_{i=1}^{N} p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left[\sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta}) \right].$$

The log-likelihood $\ell(\boldsymbol{\theta})$ could be maximized with respect to $\boldsymbol{\theta}$ by typical gradient-based methods (Sec. 2.4). However, many models of practical interest require enforcing parameter constraints that require non-trivial adaptations.

Expectation maximization is an iterative maximization procedure that starts at a (possibly arbitrary) parameter estimate $\boldsymbol{\theta}^{(0)}$, and obtains each parameter estimate $\boldsymbol{\theta}^{(t+1)}$ by maximizing a function $Q(\cdot, \boldsymbol{\theta}^{(t)})$. The procedure relies on three simple requirements, which we present next.

Firstly, the procedure requires that $Q(\boldsymbol{\theta}', \boldsymbol{\theta}') = \ell(\boldsymbol{\theta}')$, for every $\boldsymbol{\theta}'$.

Secondly, the procedure requires $Q(\cdot, \boldsymbol{\theta}')$ to be a lower bound on ℓ, for every $\boldsymbol{\theta}'$. In that case, $\ell(\boldsymbol{\theta}) \geq Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$, for every $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$.

Thirdly, the procedure requires that $Q(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)})$, for every $\boldsymbol{\theta}^{(t)}$, where $\boldsymbol{\theta}^{(t+1)}$ is the subsequent estimate obtained by the procedure. Notice that this is obviously true if $\boldsymbol{\theta}^{(t+1)}$ is a global maximum of $Q(\cdot, \boldsymbol{\theta}^{(t)})$. Expectation maximization is particularly useful when $Q(\cdot, \boldsymbol{\theta}^{(t)})$ is easy to maximize, for every $\boldsymbol{\theta}^{(t)}$.

Combining these three requirements,

$$\ell(\boldsymbol{\theta}^{(t+1)}) \geq Q(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) = \ell(\boldsymbol{\theta}^{(t)}),$$

for every $\boldsymbol{\theta}^{(t)}$, where $\boldsymbol{\theta}^{(t+1)}$ is the subsequent estimate obtained by the procedure. Consequently, the sequence $\ell(\boldsymbol{\theta}^{(0)}), \ldots, \ell(\boldsymbol{\theta}^{(T)})$ is non-decreasing. Notice that this fact does not guarantee convergence to either local or global maxima, although the procedure is guaranteed to converge to critical points for many models of practical interest.

A convenient auxiliary function $Q : \text{Val}(\boldsymbol{\Theta})^2 \to \mathbb{R}$ that satisfies these requirements, and therefore allows maximizing the log-likelihood given a partially observed dataset, is given by

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \sum_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \frac{p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta})}{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')} \\
&= \sum_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta}) - \sum_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \\
&= \sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{Z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')}[\log p(\mathbf{x}_i, \mathbf{Z}_i \mid \boldsymbol{\theta})] + \sum_{i=1}^{N} H[p(\mathbf{Z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')],
\end{aligned}
$$

assuming that $p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') > 0$, for every i, and assignment $\boldsymbol{\theta}'$ to $\boldsymbol{\Theta}$. Without further assumptions, notice that

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left[\sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta}) \right] = \sum_{i=1}^{N} \log \left[\sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \frac{p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta})}{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')} \right],$$

for every $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \text{Val}(\boldsymbol{\Theta})$. Furthermore, notice that

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \sum_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \frac{p(\mathbf{x}_i, \mathbf{z}_i \mid \boldsymbol{\theta})}{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')} = \sum_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \frac{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}) p(\mathbf{x}_i \mid \boldsymbol{\theta})}{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')} \\
&= \sum_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \frac{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta})}{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')} + \sum_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log p(\mathbf{x}_i \mid \boldsymbol{\theta}) \\
&= -\sum_{i=1}^{N} \sum_{\mathbf{z}_i} p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \frac{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')}{p(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\theta})} + \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \boldsymbol{\theta}) \\
&= \ell(\boldsymbol{\theta}) - \sum_{i=1}^{N} \text{KL}(p(\mathbf{Z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \| p(\mathbf{Z}_i \mid \mathbf{x}_i, \boldsymbol{\theta})),
\end{aligned}
$$

for every $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. Recall that the Kullback-Leibler divergence between two pmfs is always non-negative. Therefore, $\ell(\boldsymbol{\theta}) \geq Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$, for any $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. The Kullback-Leibler divergence between two pmfs is also zero if and only if they are equal. Therefore, $Q(\boldsymbol{\theta}', \boldsymbol{\theta}') = \ell(\boldsymbol{\theta}')$, for every $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. The first two requirements are satisfied.

Therefore, if we guarantee that $Q(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)})$ during expectation maximization, then it is guaranteed that the sequence $\ell(\boldsymbol{\theta}^{(0)}), \ldots, \ell(\boldsymbol{\theta}^{(T)})$ is non-decreasing.

As a final note, recall that the auxiliary function Q is given by

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{Z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')}[\log p(\mathbf{x}_i, \mathbf{Z}_i \mid \boldsymbol{\theta})] + \sum_{i=1}^{N} H[p(\mathbf{Z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')] = \sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{Z}_i \mid \mathbf{x}_i, \boldsymbol{\theta}')}[\log p(\mathbf{x}_i, \mathbf{Z}_i \mid \boldsymbol{\theta})] + C,$$

where C is a constant with respect to $\boldsymbol{\theta}$. Thus, for practical purposes, the term C may be omitted.

A Gaussian mixture model assumes that the probability density $p(\mathbf{x} \mid \boldsymbol{\theta})$ associated to observation $\mathbf{x}$ given the parameter vector $\boldsymbol{\theta}$ is given by

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{z} p(z \mid \boldsymbol{\theta}) p(\mathbf{x} \mid z, \boldsymbol{\theta}) = \sum_{z} \pi_z \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z),$$

where $\text{Val}(Z) = \{1, \ldots, K\}$, and the parameter vector $\boldsymbol{\theta}$ contains the parameters π_z, $\boldsymbol{\mu}_z$, and $\boldsymbol{\Sigma}_z$, for every z. A parameter vector $\boldsymbol{\theta}$ is valid if and only if $\sum_{z'} \pi_{z'} = 1$, and, for every z, $\pi_z > 0$, and $\boldsymbol{\Sigma}_z$ is a valid covariance matrix. In other words, a Gaussian mixture model assumes that the probability density function over observations is a weighted average (mixture) of K Gaussian density functions.

Clustering is a typical application for Gaussian mixture models. Clustering is the task of partitioning a set of observations into so-called clusters (sets of observations), such that *similar* observations belong to the same cluster. In the case of clustering based on Gaussian mixture models, each $z \in \text{Val}(Z)$ corresponds to a cluster, and $p(z \mid \mathbf{x}, \boldsymbol{\theta}) \propto_z \pi_z \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$ represents the probability of observation $\mathbf{x}$ belonging to cluster z according to the parameter vector $\boldsymbol{\theta}$.

In a Gaussian mixture model, $\mathbb{E}[\mathbf{X} \mid \boldsymbol{\theta}] = \sum_z \pi_z \boldsymbol{\mu}_z$, since

$$
\mathbb{E}[X_j \mid \boldsymbol{\theta}] = \int_{\text{Val}(X_j)} x_j p(x_j \mid \boldsymbol{\theta}) \, dx_j = \int_{\text{Val}(X_j)} x_j \left[\int_{\text{Val}(\mathbf{X}_{-j})} \left[\sum_z \pi_z \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \right] d\mathbf{x}_{-j} \right] dx_j
$$

$$
= \int_{\text{Val}(\mathbf{X})} \left[\sum_z \pi_z x_j \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \right] d\mathbf{x} = \sum_z \pi_z \left[\int_{\text{Val}(\mathbf{X})} x_j \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_z \boldsymbol{\Sigma}_z) \, d\mathbf{x} \right] = \sum_z \pi_z \mu_{z,j}.
$$

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, z_1), \ldots, (\mathbf{x}_N, z_N)$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for an unknown $\boldsymbol{\theta}^*$. Furthermore, suppose the assignments $z_1, \ldots, z_N$ are unknown.

In that case, the (observed) log-likelihood $\ell(\boldsymbol{\theta}) = \log p(\mathcal{D} \mid \boldsymbol{\theta})$ of the partially observed dataset $\mathcal{D}$ given the parameter $\boldsymbol{\theta}$ under a Gaussian mixture model is given by

$$
\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left[\sum_z \pi_z \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) \right].
$$

It is possible to attempt to maximize $\ell(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ using expectation maximization. An appropriate auxiliary function Q is given by

$$
Q(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{i=1}^{N} \mathbb{E}_{p(Z_i \mid \mathbf{x}_i, \boldsymbol{\theta}')}[\log p(\mathbf{x}_i, Z_i \mid \boldsymbol{\theta})] = \sum_{i=1}^{N} \sum_{z_i=1}^{K} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log p(\mathbf{x}_i, z_i \mid \boldsymbol{\theta})
$$

$$
= \sum_{i=1}^{N} \sum_{z_i=1}^{K} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \left[p(z_i \mid \boldsymbol{\theta}) p(\mathbf{x}_i \mid z_i, \boldsymbol{\theta}) \right]
$$

$$
= \sum_{i=1}^{N} \sum_{z_i=1}^{K} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log p(z_i \mid \boldsymbol{\theta}) + \sum_{i=1}^{N} \sum_{z_i=1}^{K} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log p(\mathbf{x}_i \mid z_i, \boldsymbol{\theta})
$$

$$
= \sum_{z_i=1}^{K} \sum_{i=1}^{N} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \pi_{z_i} + \sum_{z_i=1}^{K} \sum_{i=1}^{N} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}),
$$

where $p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \propto_{z_i} p(\mathbf{x}_i \mid z_i, \boldsymbol{\theta}') p(z_i \mid \boldsymbol{\theta}')$, which is easy to compute for every $\boldsymbol{\theta}'$ and i. These computations constitute the so-called expectation step of the algorithm. We next discuss the so-called maximization step of the algorithm, which consists on maximizing $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ with respect to $\boldsymbol{\theta}$.

Firstly, consider the task of maximizing $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ with respect to $\boldsymbol{\pi}$, which is contained in $\boldsymbol{\theta}$. Because the remaining terms of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ are constant with respect to $\boldsymbol{\pi}$, this task is equivalent to maximizing

$$
\sum_{z_i=1}^{K} \sum_{i=1}^{N} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \pi_{z_i},
$$

with respect to $\boldsymbol{\pi}$, subject to the constraint $\sum_{z'} \pi_{z'} = 1$. As will become clear, the constraint $\pi_z > 0$, for every z, will be satisfied naturally assuming proper initialization. By the Lagrange multiplier theorem, if $\boldsymbol{\pi}$ is a local maximum subject to our constraint, then

$$
\nabla_{\boldsymbol{\pi}} \left[\sum_{z_i=1}^{K} \sum_{i=1}^{N} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \pi_{z_i} + \lambda \left(1 - \sum_{z'} \pi_{z'} \right) \right] = \mathbf{0},
$$

for some $\lambda \in \mathbb{R}$. Therefore, for every $z \in \mathrm{Val}(Z)$,

$$\frac{\partial}{\partial \pi_z} \left[\sum_{z_i=1}^{K} \sum_{i=1}^{N} p(z_i \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \pi_{z_i} + \lambda \left(1 - \sum_{z'} \pi_{z'} \right) \right] = \frac{1}{\pi_z} \left[\sum_{i=1}^{N} p(Z_i = z \mid \mathbf{x}_i, \boldsymbol{\theta}') \right] - \lambda = 0.$$

If $\boldsymbol{\pi}$ is a local maximum subject to our constraint, then

$$\lambda \pi_z = \sum_{i=1}^{N} p(Z_i = z \mid \mathbf{x}_i, \boldsymbol{\theta}'),$$

for every z. The fact that $\sum_{z'} \pi_{z'} = 1$ also implies that

$$\lambda = \sum_{z'} \lambda \pi_{z'} = \sum_{i=1}^{N} \sum_{z'=1}^{K} p(Z_i = z' \mid \mathbf{x}_i, \boldsymbol{\theta}') = N.$$

Finally, if $\boldsymbol{\pi}$ is a local maximum subject to our constraint, then

$$\pi_z = \frac{1}{N} \sum_{i=1}^{N} p(Z_i = z \mid \mathbf{x}_i, \boldsymbol{\theta}'),$$

for every z. Notice that $\pi_z > 0$, as long as the analogous statement is true for the previous parameter vector $\boldsymbol{\theta}'$, satisfying the missing constraint. In summary, the updated probability π_z of cluster z is the average probability of an observation belonging to class z according to the previous parameter vector $\boldsymbol{\theta}'$.

We now consider the task of maximizing $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ with respect to $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$, individually for each z. Because the remaining terms of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ are constant with respect to these parameters, this task is equivalent to maximizing

$$\sum_{i=1}^{N} p(Z_i = z \mid \mathbf{x}_i, \boldsymbol{\theta}') \log \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z),$$

with respect to $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$, subject to the constraint that $\boldsymbol{\Sigma}_z$ is a valid covariance matrix. This task is very similar to the task of maximizing the log-likelihood of a dataset that is iid according to a multivariate Gaussian distribution, which we discussed in Sec. 4.

It is possible to show that if $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ are local maxima of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ subject to our constraint, then

$$\boldsymbol{\mu}_z = \frac{1}{N \pi_z} \sum_{i=1}^{N} p(Z_i = z \mid \mathbf{x}_i, \boldsymbol{\theta}') \mathbf{x}_i,$$

$$\boldsymbol{\Sigma}_z = \frac{1}{N \pi_z} \left[\sum_{i=1}^{N} p(Z_i = z \mid \mathbf{x}_i, \boldsymbol{\theta}') \mathbf{x}_i \mathbf{x}_i^T \right] - \boldsymbol{\mu}_z \boldsymbol{\mu}_z^T.$$

Intuitively, the updated mean $\boldsymbol{\mu}_z$ associated to cluster z is the average observation weighted by how strongly class z is associated to each observation.

This completes the description of the steps involved in attempting to maximize the (observed) log-likelihood under a Gaussian mixture model. However, this procedure may lead to severe overfitting, which may be avoided by using expectation maximization to attempt to obtain a maximum a posteriori estimate.

It is important to notice that the (observed) log-likelihood under a Gaussian mixture model does not generally have a single global optimum. For instance, swapping $(\pi_z, \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$ with $(\pi_{z'}, \boldsymbol{\mu}_{z'}, \boldsymbol{\Sigma}_{z'})$ leads to the same (observed) log-likelihood, for any $z \neq z'$.

K-means is a widely employed iterative clustering algorithm that resembles expectation maximization applied to Gaussian mixture models. Consider a sequence $\boldsymbol{\mu}_1^{(t)}, \ldots, \boldsymbol{\mu}_K^{(t)}$ composed of K so-called cluster centers at iteration t. For a maximum number of iterations T, for every z, the K-means algorithm obtains the z-th cluster center at iteration $t+1$ given by

$$\boldsymbol{\mu}_z^{(t+1)} = \frac{1}{N_z} \sum_{i=1}^{N} \mathbb{I}(z_i = z) \mathbf{x}_i,$$

where $z_i = \arg\min_z ||\mathbf{x}_i - \boldsymbol{\mu}_z^{(t)}||$ is the closest cluster center at iteration t to observation $\mathbf{x}_i$, and $N_z = \sum_i \mathbb{I}(z_i = z)$ is the number of observations whose closest cluster center at iteration t is z. In other words, at each iteration, each cluster center is moved to the average of the observations assigned to it. The success of K-means is highly dependent on how the cluster centers are initialized.

The number of clusters K is a hyperparameter for Gaussian mixture models and for K-means.

In practice, for Gaussian mixture models, K is typically chosen from a finite set of alternatives by comparing their (observed) log-likelihood on a test set. Comparisons based on the training set are inconclusive, since larger K will generally achieve a higher value, which is another example of overfitting.

In practice, for K-means, K is typically chosen from a finite set of alternatives by comparing their average reconstruction error J on the training set. The average reconstruction error J is given by

$$J = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \boldsymbol{\mu}_{z_i}||^2,$$

where $\boldsymbol{\mu}_{z_i}$ is the closest cluster center to observation $\mathbf{x}_i$. The average reconstruction error will also typically decrease when K increases. Heuristically, K is typically selected when the *rate* of decrease in J becomes *small*.

A Bernoulli mixture model assumes that the probability $p(\mathbf{x} \mid \boldsymbol{\theta})$ associated to observation $\mathbf{x} \in \{0,1\}^D$ given the parameter vector $\boldsymbol{\theta}$ is given by

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_z p(z \mid \boldsymbol{\theta}) p(\mathbf{x} \mid z, \boldsymbol{\theta}) = \sum_z \pi_z \prod_{j=1}^{D} \mathrm{Ber}(x_j \mid \theta_{z,j}) = \sum_z \pi_z \prod_{j=1}^{D} \theta_{z,j}^{x_j} (1 - \theta_{z,j})^{1 - x_j},$$

where $\mathrm{Val}(Z) = \{1, \ldots, K\}$, and the parameter vector $\boldsymbol{\theta}$ contains π_z and $\theta_{z,j}$, for every z and j. A parameter vector $\boldsymbol{\theta}$ is valid if and only if $\sum_{z'} \pi_{z'} = 1$, $\pi_z > 0$, and $0 < \theta_{z,j} < 1$, for every z and j. In other words, a Bernoulli mixture model assumes that the joint probability mass function over observations is a weighted average (mixture) of K joint probability mass functions, each of which assumes that each feature is independent of the others. Analogously to a Gaussian mixture model, expectation maximization may be employed to attempt to maximize the (observed) log-likelihood under a Bernoulli mixture model.

Consider a regression model that assumes that the probability density $p(y \mid \mathbf{x}, \boldsymbol{\theta})$ associated to target $y \in \mathbb{R}$ given observation $\mathbf{x} \in \mathbb{R}^D$ and parameter vector $\boldsymbol{\theta}$ is given by

$$p(y \mid \mathbf{x}, \boldsymbol{\theta}) = \sum_z p(y, z \mid \mathbf{x}, \boldsymbol{\theta}) = \sum_z p(y \mid \mathbf{x}, z, \boldsymbol{\theta}) p(z \mid \mathbf{x}, \boldsymbol{\theta}) = \sum_z \mathcal{N}(y \mid \mathbf{w}_z \mathbf{x}, \sigma_z^2) \mathcal{S}(\mathbf{V}^T \mathbf{x})_z,$$

where $\mathcal{S} : \mathbb{R}^K \to (0,1)^K$ is the softmax function, $\mathrm{Val}(Z) = \{1, \ldots, K\}$, and the parameter vector $\boldsymbol{\theta}$ contains the $D \times K$ matrix $\mathbf{V}$, and the parameters $\mathbf{w}_z \in \mathbb{R}^D$ and $\sigma_z^2 > 0$, for every z. In other words, the probability density function over targets given an observation $\mathbf{x}$ is a weighted average (mixture) of K Gaussian probability density functions, whose weights depend on $\mathbf{x}$. This model is a particular instance of a mixture of experts, and is particularly useful when the probability density function over targets given observations is multimodal. In such cases, the posterior mean, which is the optimal prediction under a squared error loss function, is not generally a good prediction, and should be substituted by the posterior mode. Once again, expectation maximization may be employed to attempt to maximize the (conditional, observed) log-likelihood of such model.

The models described so far expected a dataset containing unknown assignments to a sequence of hidden random vectors (or variables) $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$.

In contrast, consider a dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathbb{R}^D$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for an unknown $\boldsymbol{\theta}^*$. Furthermore, suppose that each observation $\mathbf{x}_i$ can be partitioned into an assignment $\mathbf{x}_{i,v}$ to an observed random vector $\mathbf{X}_{i,v}$, and an unknown assignment $\mathbf{x}_{i,h}$ to a hidden random vector $\mathbf{X}_{i,h}$. In other words, there is not (necessarily) a correspondence between hidden random vectors across elements of the dataset.

In such case, it is important to consider the so-called observation model, which models the mechanism that hides assignments. Concretely, consider the dataset $\mathcal{D}$ mentioned above, and let $\mathbf{o}_i \in \{0,1\}^D$ represent which features are observed in observation $\mathbf{x}_i$, for every i, such that $o_{i,j} = 1$ if and only if $X_{i,j}$ is an observed random variable.

The corresponding probability density $p(\mathbf{x}_{i,v}, \mathbf{o}_i \mid \boldsymbol{\theta})$ associated to the assignment $\mathbf{x}_{i,v}$ given $\boldsymbol{\theta}$ when only $\mathbf{x}_{i,v}$ is observed (which is encoded in $\mathbf{o}_i$) is given by

$$p(\mathbf{x}_{i,v}, \mathbf{o}_i \mid \boldsymbol{\theta}) = \int_{\mathrm{Val}(\mathbf{X}_{i,h})} p(\mathbf{x}_{i,v}, \mathbf{x}_{i,h}, \mathbf{o}_i \mid \boldsymbol{\theta}) \, d\mathbf{x}_{i,h} = \int_{\mathrm{Val}(\mathbf{X}_{i,h})} p(\mathbf{o}_i \mid \mathbf{x}_{i,v}, \mathbf{x}_{i,h}, \boldsymbol{\theta}) p(\mathbf{x}_{i,v}, \mathbf{x}_{i,h} \mid \boldsymbol{\theta}) \, d\mathbf{x}_{i,h}.$$

The so-called missing at random assumption states that $\mathbf{O}_i \perp\!\!\!\perp \mathbf{X}_{i,h}, \mathbf{\Theta} \mid \mathbf{X}_{i,v}$, for every i. In other words, which variables are observed is independent of the assignment to the hidden variables (and parameter vector) given the assignment to the observed variables. Under such assumption,

$$p(\mathbf{x}_{i,v}, \mathbf{o}_i \mid \boldsymbol{\theta}) = \int_{\text{Val}(\mathbf{X}_{i,h})} p(\mathbf{o}_i \mid \mathbf{x}_{i,v}) p(\mathbf{x}_{i,v}, \mathbf{x}_{i,h} \mid \boldsymbol{\theta}) \, d\mathbf{x}_{i,h} = p(\mathbf{o}_i \mid \mathbf{x}_{i,v}) p(\mathbf{x}_{i,v} \mid \boldsymbol{\theta}).$$

Therefore, the probability density associated to the sequence of observed random vectors and their corresponding assignments in $\mathcal{D}$ given $\boldsymbol{\theta}$ under a missing at random assumption is given by

$$p(\mathbf{x}_{1,v}, \mathbf{o}_1, \ldots, \mathbf{x}_{N,v}, \mathbf{o}_N \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_{i,v}, \mathbf{o}_i \mid \boldsymbol{\theta}) = \left[\prod_{i=1}^{N} p(\mathbf{o}_i \mid \mathbf{x}_{i,v}) \right] \left[\prod_{i=1}^{N} p(\mathbf{x}_{i,v} \mid \boldsymbol{\theta}) \right].$$

Clearly, maximizing such probability density with respect to $\boldsymbol{\theta}$ is equivalent to maximizing the (observed) log-likelihood, which we define as

$$\ell(\boldsymbol{\theta}) = \log \prod_{i=1}^{N} p(\mathbf{x}_{i,v} \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\mathbf{x}_{i,v} \mid \boldsymbol{\theta}),$$

assuming that $p(\mathbf{x}_{i,v} \mid \boldsymbol{\theta}) > 0$, for every i and $\boldsymbol{\theta}$.

In summary, maximizing the (observed) log-likelihood is only appropriate when the observation model hides a random vector (or variable) disregarding its assignment. Expectation maximization may be employed to attempt to maximize the corresponding (observed) log-likelihood under many joint probability density/mass functions of practical interest, such as the multivariate Gaussian distribution.

Similarly, expectation maximization may also be employed to attempt to maximize the (observed) log-likelihood of partially observed data in the context of Bayesian networks. For details, see the corresponding notes.

12 Continuous hidden variables

Dimensionality reduction techniques address the task of representing a dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathbb{R}^D$, by a dataset $\mathcal{Z} = \mathbf{z}_1, \ldots, \mathbf{z}_N$, where $\mathbf{z}_i \in \mathbb{R}^d$, $d < D$, and each $\mathbf{z}_i$ corresponds to $\mathbf{x}_i$. Different techniques attempt to preserve different aspects of a dataset $\mathcal{D}$ in the corresponding dataset $\mathcal{Z}$.

We now present principal component analysis, a widely used technique for dimensionality reduction. This technique is highly related to a probabilistic model with continuous hidden variables, which is discussed later in this section.

Consider a dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$. Firstly, note that the mean projection onto the vector $\mathbf{u}_1 \in \mathbb{R}^D$ of the vectors in $\mathcal{D}$ can be written as

$$\frac{1}{N} \sum_{i=1}^{N} \mathbf{u}_1 \mathbf{x}_i = \mathbf{u}_1 \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i = \mathbf{u}_1 \bar{\mathbf{x}},$$

where $\bar{\mathbf{x}}$ is the empirical mean. Consider the task of finding a *unit* vector $\mathbf{u}_1$ such that the variance

$$v(\mathbf{u}_1) = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{u}_1 \mathbf{x}_i - \mathbf{u}_1 \bar{\mathbf{x}})^2$$

of the projections onto $\mathbf{u}_1$ of the vectors in $\mathcal{D}$ is maximum. It can be easily shown that $v(\mathbf{u}_1)$ can also be written as

$$v(\mathbf{u}_1) = \frac{1}{N} \sum_{i=1}^{N} \left[(\mathbf{u}_1 \mathbf{x}_i)^2 - 2(\mathbf{u}_1 \mathbf{x}_i)(\mathbf{u}_1 \bar{\mathbf{x}}) + (\mathbf{u}_1 \bar{\mathbf{x}})^2 \right] = \mathbf{u}_1^T \mathbf{\Sigma} \mathbf{u}_1,$$

where $\mathbf{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$ is the empirical covariance matrix of the dataset $\mathcal{D}$.

The maximization task outlined above corresponds to finding the maxima of v restricted to the D-dimensional unit sphere $S = \{\mathbf{u}_1 \in \mathbb{R}^D \mid g(\mathbf{u}_1) = 0\}$, where $g(\mathbf{u}_1) = 1 - \mathbf{u}_1 \mathbf{u}_1$. The Lagrange multiplier theorem states that if $\mathbf{u}_1$ is a local maximum of v restricted to S, and $\nabla g(\mathbf{u}_1) \neq 0$, then

$$\nabla v(\mathbf{u}_1) + \lambda_1 \nabla g(\mathbf{u}_1) = 0,$$

for some $\lambda_1 \in \mathbb{R}$. Thus,

$$0 = \nabla_{\mathbf{u}_1}\left[\mathbf{u}_1^T \mathbf{\Sigma} \mathbf{u}_1\right] + \lambda_1 \nabla_{\mathbf{u}_1}\left[1 - \mathbf{u}_1 \mathbf{u}_1\right] = (\mathbf{\Sigma} + \mathbf{\Sigma}^T)\mathbf{u}_1 - 2\lambda_1 \mathbf{u}_1 = 2\mathbf{\Sigma}\mathbf{u}_1 - 2\lambda_1 \mathbf{u}_1,$$

where $(\mathbf{\Sigma} + \mathbf{\Sigma}^T) = 2\mathbf{\Sigma}$ because $\mathbf{\Sigma}$ is symmetric. Therefore, if $\mathbf{u}_1$ is a local maximum of v restricted to S, then $\mathbf{\Sigma}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$ for some λ_1. In other words, $\mathbf{u}_1$ is an eigenvector of $\mathbf{\Sigma}$ with eigenvalue λ_1. Because the variance $v(\mathbf{u}_1)$ corresponding to such an eigenvector $\mathbf{u}_1$ is the corresponding eigenvalue $\lambda_1 = \mathbf{u}_1^T \mathbf{\Sigma} \mathbf{u}_1$, the eigenvector with maximum associated eigenvalue is the desired global maximum restricted to S. We omit the proof that such eigenvector is necessarily a constrained local maximum.

Consider an orthonormal list $L_{i-1} = (\mathbf{u}_1, \dots \mathbf{u}_{i-1})$ containing the largest eigenvectors (with respect to their corresponding eigenvalues), sorted in non-increasing order, of the empirical covariance matrix $\mathbf{\Sigma}$, for some $1 < i < D$. Furthermore, consider the task of maximizing $v(\mathbf{u}_i)$ with respect to $\mathbf{u}_i$, subject to the constraints that $0 = 1 - \mathbf{u}_i \mathbf{u}_i$ and $0 = \mathbf{u}_i \mathbf{u}_1 = \dots = \mathbf{u}_i \mathbf{u}_{i-1}$. In other words, we are interested in a unit vector $\mathbf{u}_i$ that is orthogonal to every vector in the list L_{i-1}, and that also has maximum possible variance $v(\mathbf{u}_i)$. By the Lagrange multiplier theorem, if $\mathbf{u}_i$ is a local maximum of v subject to these constraints, then

$$0 = \nabla_{\mathbf{u}_i}\left[\mathbf{u}_i^T \mathbf{\Sigma} \mathbf{u}_i\right] + \lambda_i \nabla_{\mathbf{u}_i}\left[1 - \mathbf{u}_i \mathbf{u}_i\right] + \sum_{j=1}^{i-1} \lambda_j \nabla_{\mathbf{u}_i}\left[\mathbf{u}_i \mathbf{u}_j\right] = 2\mathbf{\Sigma}\mathbf{u}_i - 2\lambda_i \mathbf{u}_i + \sum_{j=1}^{i-1} \lambda_j \mathbf{u}_j,$$

for some $\lambda_1, \dots, \lambda_i \in \mathbb{R}$. Consider any vector $\mathbf{u}_k$ in L_{i-1}. By left-multiplying the equation above by $\mathbf{u}_k^T$,

$$0 = 2\mathbf{u}_k^T \mathbf{\Sigma} \mathbf{u}_i - 2\lambda_i \mathbf{u}_k^T \mathbf{u}_i + \sum_{j=1}^{i-1} \lambda_j \mathbf{u}_k^T \mathbf{u}_j.$$

Since $\mathbf{u}_k$ and the desired $\mathbf{u}_i$ must be orthogonal, and since L_{i-1} is an orthogonal list of vectors,

$$0 = 2\mathbf{u}_k^T \mathbf{\Sigma} \mathbf{u}_i + \lambda_k.$$

Because $\mathbf{u}_k^T \mathbf{\Sigma} \mathbf{u}_i = (\mathbf{u}_k^T \mathbf{\Sigma} \mathbf{u}_i)^T = (\mathbf{\Sigma} \mathbf{u}_i)^T \mathbf{u}_k = \mathbf{u}_i^T \mathbf{\Sigma}^T \mathbf{u}_k = \mathbf{u}_i^T \mathbf{\Sigma} \mathbf{u}_k$, and because $\mathbf{\Sigma} \mathbf{u}_k = \alpha_k \mathbf{u}_k$ for some $\alpha_k \in \mathbb{R}$,

$$\lambda_k = -2\mathbf{u}_i^T \mathbf{\Sigma} \mathbf{u}_k = -2\alpha_k \mathbf{u}_i^T \mathbf{u}_k = 0.$$

Therefore, $\lambda_k = 0$ for any $1 \leq k \leq i - 1$. Recall that if $\mathbf{u}_i$ is a local maximum subject to our constraints, then

$$0 = 2\mathbf{\Sigma}\mathbf{u}_i - 2\lambda_i \mathbf{u}_i + \sum_{j=1}^{i-1} \lambda_j \mathbf{u}_j = 2\mathbf{\Sigma}\mathbf{u}_i - 2\lambda_i \mathbf{u}_i.$$

Therefore, $\mathbf{\Sigma}\mathbf{u}_i = \lambda_i \mathbf{u}_i$ for some $\lambda_i \in \mathbb{R}$. In other words, $\mathbf{u}_i$ is an eigenvector of $\mathbf{\Sigma}$ with associated eigenvalue $\lambda_i = v(\mathbf{u}_i)$. Because the vector $\mathbf{u}_i$ must be orthogonal to any vector already in $L_{i-1} = (\mathbf{u}_1, \dots \mathbf{u}_{i-1})$, the constrained global maximum is one of the remaining eigenvectors with maximum corresponding eigenvalue. Once again, we omit the proof that such eigenvector is necessarily a constrained local maximum. Because the list $L_i = (\mathbf{u}_1, \dots, \mathbf{u}_i)$ is composed of orthonormal eigenvectors of the empirical covariance matrix $\mathbf{\Sigma}$ with the highest corresponding eigenvalues and sorted in non-increasing order, we have completed the inductive step. We also note that $\mathbf{\Sigma}$ is positive definite whenever $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ spans $\mathbb{R}^D$.

Using such a list of eigenvectors L_d, any observation $\mathbf{x}_i \in \mathbb{R}^D$ can be represented by an observation $\mathbf{z}_i = (\mathbf{u}_1 \mathbf{x}_i, \dots, \mathbf{u}_d \mathbf{x}_i)$ in $\mathbb{R}^d$.

In practice, it is essential to transform an original dataset $\mathcal{D}'$ into a dataset $\mathcal{D}$ with empirical mean $\bar{\mathbf{x}} = 0$ before applying principal component analysis, since the increase in variance along a direction due to translations is usually irrelevant. In that case, principal component analysis also finds the linear map with minimum average reconstruction error [2]. In many applications, the dataset $\mathcal{D}$ should also have the same empirical variance across each feature (i.e., the dataset $\mathcal{D}'$ should also be standardized) to minimize the effect of the choice of units.

Consider a dataset $\mathcal{D} = \mathbf{x}_1, \dots, \mathbf{x}_N$ with empirical mean $\bar{\mathbf{x}} = \mathbf{0}$, and the corresponding $N \times D$ design matrix $\mathbf{X}$, where each observation corresponds to a row. A singular value decomposition of $\mathbf{X}$ is a factorization given by $\mathbf{X} = \mathbf{L}\mathbf{S}\mathbf{R}^T$, where $\mathbf{L}$ is an $N \times N$ orthogonal matrix ($\mathbf{L}\mathbf{L}^T = \mathbf{L}^T\mathbf{L} = \mathbf{I}$), $\mathbf{S}$ is a diagonal $N \times D$ matrix composed of non-negative elements, and $\mathbf{R}$ is a $D \times D$ orthogonal matrix. Such factorization always exists.

Since we assumed that $\mathcal{D}$ is centered, the corresponding empirical covariance matrix $\mathbf{\Sigma}$ can be written as

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T = \frac{1}{N}\mathbf{X}^T\mathbf{X} = \frac{1}{N}\left(\mathbf{L}\mathbf{S}\mathbf{R}^T\right)^T \mathbf{L}\mathbf{S}\mathbf{R}^T = \frac{1}{N}\mathbf{R}\mathbf{S}^T\mathbf{L}^T\mathbf{L}\mathbf{S}\mathbf{R}^T = \frac{1}{N}\mathbf{R}\mathbf{S}^T\mathbf{S}\mathbf{R}^T.$$

Finally, because $\mathbf{R}$ is an orthogonal matrix, $\mathbf{\Sigma R} = \frac{1}{N}\mathbf{R}\mathbf{S}^T\mathbf{S}$. Therefore, since $\mathbf{S}^T\mathbf{S}$ is a $D \times D$ diagonal matrix, this implies that $\mathbf{\Sigma r}_j = \frac{S_{j,j}^2}{N}\mathbf{r}_j$, where $\mathbf{r}_j$ is the j-th column of $\mathbf{R}$. In other words, the j-th column $\mathbf{r}_j$ of $\mathbf{R}$ is an eigenvector of $\mathbf{\Sigma}$ with eigenvalue $\frac{S_{j,j}^2}{N}$. Because $(\mathbf{r}_1,\ldots,\mathbf{r}_D)$ is also an orthonormal basis for $\mathbb{R}^D$, this result allows using a singular value decomposition of $\mathbf{X}$ to perform dimensionality reduction using principal component analysis.

We now present probabilistic principal component analysis. Consider a dataset $\mathcal{D} = (\mathbf{x}_1, \mathbf{z}_1),\ldots,(\mathbf{x}_N, \mathbf{z}_N)$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$, for some unknown $\boldsymbol{\theta}^*$. Furthermore, suppose that each $\mathbf{x}_i \in \mathbb{R}^D$ is an assignment to a continuous random vector $\mathbf{X}_i$, and that each $\mathbf{z}_i \in \mathbb{R}^d$ is an assignment to a continuous random vector $\mathbf{Z}_i$.

If $\mathbf{Z}_1,\ldots,\mathbf{Z}_N$ are hidden random vectors, then the assignments $\mathbf{z}_1,\ldots,\mathbf{z}_N$ are unknown, and the (observed) likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ of the partially observed dataset $\mathcal{D}$ given the parameter $\boldsymbol{\theta}$ is defined as

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = p(\mathbf{x}_1,\ldots,\mathbf{x}_N \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \boldsymbol{\theta}).$$

Probabilistic principal component analysis assumes that the probability density $p(\mathbf{z} \mid \boldsymbol{\theta})$ associated to a hidden assignment $\mathbf{z} \in \mathbb{R}^d$ given a parameter vector $\boldsymbol{\theta}$ is given by $p(\mathbf{z} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$. Furthermore, it assumes that the conditional probability density $p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta})$ associated to an observation $\mathbf{x} \in \mathbb{R}^D$ given the hidden assignment $\mathbf{z} \in \mathbb{R}^d$ and the parameter vector $\boldsymbol{\theta}$ is given by

$$p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x} \mid \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}),$$

such that $\boldsymbol{\theta}$ contains the $D \times d$ matrix $\mathbf{W}$, the vector $\boldsymbol{\mu} \in \mathbb{R}^D$, and $\sigma^2 > 0$.

Since this probabilistic model is a linear Gaussian system (Sec. 4), the probability density $p(\mathbf{x} \mid \boldsymbol{\theta})$ associated to an observation $\mathbf{x} \in \mathbb{R}^D$ given the parameter vector $\boldsymbol{\theta}$ is given by

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \int_{\text{Val}(\mathbf{Z})} p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z} \mid \boldsymbol{\theta}) \, d\mathbf{z} = \int_{\text{Val}(\mathbf{Z})} \mathcal{N}(\mathbf{x} \mid \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I}) \, d\mathbf{z} = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}).$$

Notice that such joint probability density function over observations only requires $Dd + D + 1$ parameters, whereas a multivariate Gaussian distribution over observations would require $D(D+1)/2 + D$ parameters.

Consider two parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, which are equal except possibly for their respective weight matrices $\mathbf{W}$ and $\mathbf{W}'$, and suppose that $\mathbf{W}' = \mathbf{W}\mathbf{R}$, for an arbitrary $d \times d$ orthogonal matrix $\mathbf{R}$. Clearly,

$$p(\mathbf{x} \mid \boldsymbol{\theta}') = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \mathbf{W}'\mathbf{W}'^T + \sigma^2 \mathbf{I}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{R}\mathbf{R}^T\mathbf{W}^T + \sigma^2 \mathbf{I}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}) = p(\mathbf{x} \mid \boldsymbol{\theta}).$$

Therefore, there is an infinite number of weight matrices that lead to each possible joint probability density function over observations according to this model.

It can also be shown that the probability density $p(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta})$ associated to the hidden assignment $\mathbf{z} \in \mathbb{R}^d$ given the observation $\mathbf{x} \in \mathbb{R}^D$ and the parameter vector $\boldsymbol{\theta}$ is given by

$$p(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{z} \mid \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}), \sigma^2 \mathbf{M}^{-1}),$$

where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2 \mathbf{I}$ is a $d \times d$ matrix.

Consider the task of maximizing the (observed) log-likelihood $\ell(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, which is given by

$$\ell(\boldsymbol{\theta}) = \log p(\mathcal{D} \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}).$$

Furthermore, consider a list $(\mathbf{u}_1,\ldots,\mathbf{u}_d)$ composed of orthonormal eigenvectors of the empirical covariance matrix $\mathbf{\Sigma}$ with the highest corresponding eigenvalues and sorted in non-increasing order. Let $\mathbf{U}$ denote a $D \times d$ matrix whose j-th column is $\mathbf{u}_j \in \mathbb{R}^D$, and let $\mathbf{\Lambda}$ denote a $d \times d$ diagonal matrix such that $\Lambda_{j,j} = \lambda_j$ is the eigenvalue associated to eigenvector $\mathbf{u}_j$. It can be shown that the (observed) log-likelihood is maximized when

$$\boldsymbol{\mu} = \bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$$

$$\sigma^2 = \frac{1}{D-d}\sum_{j=d+1}^{D} \lambda_j$$

$$\mathbf{W} = \mathbf{U}(\mathbf{\Lambda} - \sigma^2 \mathbf{I})^{\frac{1}{2}}.$$

47

Consider the expected value $\mathbb{E}[\mathbf{Z} \mid \mathbf{x}, \boldsymbol{\theta}]$ of the hidden random vector $\mathbf{Z}$ given the observation $\mathbf{x}$ and the parameter vector $\boldsymbol{\theta}$, which is given by

$$\mathbb{E}[\mathbf{Z} \mid \mathbf{x}, \boldsymbol{\theta}] = \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}) = (\mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I})^{-1}\mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}).$$

Using the maximum likelihood estimate $\boldsymbol{\theta}$ described above and the fact that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$,

$$\mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I} = (\boldsymbol{\Lambda} - \sigma^2\mathbf{I})^{\frac{1}{2}}\mathbf{U}^T\mathbf{U}(\boldsymbol{\Lambda} - \sigma^2\mathbf{I})^{\frac{1}{2}} + \sigma^2\mathbf{I} = \boldsymbol{\Lambda} - \sigma^2\mathbf{I} + \sigma^2\mathbf{I} = \boldsymbol{\Lambda}.$$

Therefore, using the same maximum likelihood estimate $\boldsymbol{\theta}$,

$$\mathbb{E}[\mathbf{Z} \mid \mathbf{x}, \boldsymbol{\theta}] = \boldsymbol{\Lambda}^{-1}(\boldsymbol{\Lambda} - \sigma^2\mathbf{I})^{\frac{1}{2}}\mathbf{U}^T(\mathbf{x} - \bar{\mathbf{x}}).$$

Furthermore, notice that when $\sigma^2 \to 0$, the corresponding expected value

$$\mathbb{E}[\mathbf{Z} \mid \mathbf{x}, \boldsymbol{\theta}] \to \boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T(\mathbf{x} - \bar{\mathbf{x}})$$

is very similar to d-dimensional representation of an observation $\mathbf{x} \in \mathbb{R}^D$ obtained by principal component analysis (after the dataset $\mathcal{D}$ is centered), except for the multiplication by the diagonal matrix $\boldsymbol{\Lambda}^{-\frac{1}{2}}$.

13   Sparse linear models

The success of supervised learning techniques is highly dependent on the appropriate choice of features. Using few features may lead to poor generalization, while using many features may be too expensive, or even introduce confounding information into the training data. The task of identifying features that are important for supervised learning is called feature selection.

In classification, the mutual information $I(X_j; Y)$ may be used to measure how much information a discrete feature variable $X_j \sim p_{X_j}$ has about the class variable $Y \sim p_Y$. This requires approximating the (typically unknown) probability mass functions p_{X_j}, p_Y and $p_{X_j, Y}$ by their corresponding empirical distributions given an iid dataset $\mathcal{D}$. However, these approximations may be unreliable, particularly when the number of observations in $\mathcal{D}$ is small. Furthermore, notice that this approach does not take into account whether a feature is important when considered together with other features.

Because there are $2^D - 1$ distinct non-empty subsets of a set of D features, evaluating the efficacy of every feature subset using cross-validation is generally infeasible. Instead, feature selection is typically performed by heuristic methods. This section describes feature selection heuristics based on logistic regression. Analogous heuristics exist based on other generalized linear models.

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, which is iid according to $p(\cdot \mid \boldsymbol{\theta}^*)$ for some unknown $\boldsymbol{\theta}^*$. Also, suppose $\mathbf{x}_i \in \mathbb{R}^D$, and $y_i \in \{0, 1\}$, for all i.

Recall that logistic regression is a binary classification technique that assumes that the probability $p(y \mid \mathbf{x}, \mathbf{w})$ of class $y \in \{0, 1\}$ given observation $\mathbf{x}$ and weight vector $\mathbf{w}$ is given by

$$p(y \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}\mathbf{x})^y(1 - \sigma(\mathbf{w}\mathbf{x}))^{1-y}, \tag{2}$$

where σ is the sigmoid function.

We have already shown that the corresponding log-likelihood $\log p(\mathcal{D} \mid \mathbf{w})$ associated to the dataset $\mathcal{D}$ given $\mathbf{w}$ is given by

$$\log p(\mathcal{D} \mid \mathbf{w}) = \sum_{i=1}^{N} y_i \log \sigma(\mathbf{w}\mathbf{x}_i) + (1 - y_i)\log(1 - \sigma(\mathbf{w}\mathbf{x}_i)).$$

However, the corresponding maximum (log-)likelihood estimate $\arg\max_{\mathbf{w}} p(\mathcal{D} \mid \mathbf{w})$ may have extremely large elements, particularly when the data is linearly separable. This leads to a classifier that is very sensitive to small changes in the input observations, which may be overfitted to the training data. This issue can be circumvented by finding a maximum a posteriori estimate $\arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathcal{D})$. Naturally, such estimate also maximizes $\log p(\mathcal{D} \mid \mathbf{w}) + \log p(\mathbf{w})$.

For this purpose, we have already considered a prior density $p(\mathbf{w})$ associated to a weight vector $\mathbf{w}$ given by

$$p(\mathbf{w}) = \prod_{j=1}^{D} \mathcal{N}(w_j \mid 0, \tau^2),$$

where $\tau^2 > 0$ is a hyperparameter. The corresponding maximum a posteriori estimate also maximizes (with respect to $\mathbf{w}$) the so-called l^2-regularized log-likelihood given by

$$\log p(\mathcal{D} \mid \mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||^2,$$

where $\lambda = 1/\tau^2$.

Alternatively, consider a prior density $p(\mathbf{w})$ associated to a weight vector $\mathbf{w}$ given by

$$p(\mathbf{w}) = \prod_{j=1}^{D} \text{Lap}(w_j \mid 0, \frac{1}{\lambda}) = \prod_{j=1}^{D} \frac{\lambda}{2} e^{-\lambda|w_j|} = \left(\frac{\lambda}{2}\right)^D \prod_{j=1}^{D} e^{-\lambda|w_j|}.$$

Clearly,

$$\log p(\mathbf{w}) = \log \left[\left(\frac{\lambda}{2}\right)^D \prod_{j=1}^{D} e^{-\lambda|w_j|} \right] = D \log \frac{\lambda}{2} - \lambda \sum_{j=1}^{D} |w_j|.$$

Therefore, by eliminating irrelevant constants, the corresponding maximum a posteriori estimate also maximizes (with respect to $\mathbf{w}$) the so-called l^1-regularized log-likelihood given by

$$\log p(\mathcal{D} \mid \mathbf{w}) - \lambda||\mathbf{w}||_1,$$

where $||\mathbf{w}||_1 = \sum_j |w_j|$ is the so-called l^1-norm of vector $\mathbf{w}$

In comparison to the l^2-regularized log-likelihood, the l^1-regularized log-likelihood favors a sparse solution $\mathbf{w}^*$ (where many elements are zero), although we omit the rigorous argument. Notice that the l^1-regularized log-likelihood is not differentiable with respect to $\mathbf{w}$, requiring optimization methods that we also omit.

Feature selection using logistic regression is based on the the idea that if $\mathbf{w}^*$ is a weight vector that maximizes the l^1-regularized log-likelihood, and $w_j^* = 0$, then feature j is (probably) not relevant for classification. Notice that this heuristic takes into consideration that a feature may be discriminative when combined with others even if it is not discriminative by itself.

However, notice that $w_j^* = 0$ does not necessarily imply that a feature is not relevant for classification. For instance, a feature k that is always identical to feature j could exist, and w_k^* could be nonzero. Clearly, this example generalizes to groups of features. Although this behavior is appropriate for selecting subsets of features, it is not appropriate for scoring features individually.

Randomized logistic regression deals with this issue by introducing randomness into the learning process. We discuss one possible implementation of the original approach, which is based on stability selection.

Consider a sequence of datasets $\mathcal{D}_1, \ldots, \mathcal{D}_S$, each obtained by randomly choosing subsets composed of $N' < N$ elements from an original iid dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$. Suppose a logistic regression classifier is fitted independently for each of these datasets, resulting in a sequence of weight vectors $\mathbf{w}_1^*, \ldots, \mathbf{w}_S^*$.

The score s_j of feature j is defined as the fraction of classifiers where feature j was associated to a *significant* coefficient, and is given by

$$s_j = \frac{1}{S} \sum_{i=1}^{S} \mathbb{I}(|w_{i,j}^*| \geq \epsilon), \tag{3}$$

where $\mathbb{I}$ is the indicator function, and $\epsilon \geq 0$ is a small constant. Intuitively, the variability introduced by subsampling is likely to affect which features are used by each classifier, avoiding the issue involving related features.

So far, we have omitted the main idea in stability selection applied to logistic regression, which is penalizing each feature differently when considering each dataset. This idea can be implemented by maximizing, for each dataset $\mathcal{D}_i$, the l^1-regularized log-likelihood with respect to $\mathbf{w}$ considering randomized penalties, which is given by

$$\log p(\mathcal{D} \mid \mathbf{w}) - \lambda \sum_{j=1}^{D} \frac{|w_j|}{\omega_j}, \tag{4}$$

where λ is the usual penalty hyperparameter, but each ω_j is chosen uniformly at random from the set $\{r, 1\}$, where $0 < r < 1$ is a hyperparameter of the feature selection method. Intuitively, this introduces even more variability in the set of features that is likely to be used by each classifier.

14   Kernel methods

We have already mentioned that it is often useful to apply a basis function $\phi : \mathbb{R}^D \to \mathbb{R}^{D'}$ to the observations in a dataset. In classification, for instance, an appropriate basis function can make a dataset linearly separable.

A Mercer kernel is a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ given by

$$\kappa(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})\phi(\mathbf{x}') = \phi(\mathbf{x}')\phi(\mathbf{x}),$$

for some basis function $\phi : \mathcal{X} \to \mathcal{V}$, and all observations $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. In contrast to our usual definition, the observations in $\mathcal{X}$ may be of any sort, and $\mathcal{V}$ may be an arbitrary inner product space (over the field of real numbers). For our purposes, the scalar $\kappa(\mathbf{x}, \mathbf{x}')$ should roughly correspond to a measure of similarity between $\mathbf{x}$ and $\mathbf{x}'$.

Consider a dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathcal{X}$. The Gram matrix $\mathbf{K}$ of κ for $\mathcal{D}$ is given by $K_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. It is possible to show that a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a Mercer kernel if and only if its Gram matrix is positive semidefinite for every possible dataset [2].

The following are examples of Mercer kernels.

The linear kernel $\kappa : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ given by $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}\mathbf{x}'$.

The polynomial kernel $\kappa : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ given by $\kappa(\mathbf{x}, \mathbf{x}') = (\mathbf{x}\mathbf{x}' + r)^M$, where $r > 0$ and $M \in \mathbb{N}_{>0}$ are hyperparameters.

The radial basis function kernel $\kappa : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ given by

$$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right),$$

where $\sigma^2 > 0$ is a hyperparameter. The corresponding basis function ϕ maps $\mathbb{R}^D$ into an infinite dimensional inner product space $\mathcal{V}$. As will become clear, kernels allow applying some traditional techniques as if observations were transformed by such intractable basis functions.

The string kernel $\kappa : \mathcal{A}^* \times \mathcal{A}^* \to \mathbb{R}$ given by

$$k(x, x') = \sum_{s \in \mathcal{A}^*} w_s \phi_s(x) \phi_s(x'),$$

where $\mathcal{A}^*$ is the set of all possible strings over the alphabet $\mathcal{A}$, $\phi_s(x) \geq 0$ is the number of times that string $s \in \mathcal{A}^*$ appears inside string $x \in \mathcal{A}^*$, and $w_s \geq 0$, for all s. As will become clear, kernels allow applying some traditional techniques to such observations.

Kernel machines are simple supervised learning models based on kernels. A kernel machine is a generalized linear model that receives observations transformed by a basis function $\phi : \mathcal{X} \to \mathbb{R}^K$ given by

$$\phi(\mathbf{x}) = (\kappa(\mathbf{x}, \boldsymbol{\mu}_1), \ldots, \kappa(\mathbf{x}, \boldsymbol{\mu}_K)),$$

for some function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (not necessarily a Mercer kernel), and pre-defined prototypes $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, where $\boldsymbol{\mu}_i \in \mathcal{X}$. In logistic regression, for instance, an appropriate choice of prototypes can make a dataset linearly separable. In linear regression, an appropriate choice of prototypes allows fitting arbitrary functions.

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathbb{R}$. A kernel machine whose prototypes are given by $\mathbf{x}_i = \boldsymbol{\mu}_i$, for every i, and that employs a sparsity promoting prior over weight vectors is called a sparse vector machine. After obtaining a maximum a posteriori weight vector $\mathbf{w} = \arg\max_{\mathbf{w}} p(\mathbf{w} \mid \mathcal{D})$, any observation $\mathbf{x}_i$ such that $w_i = 0$ is no longer needed to compute $\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}, \mathbf{W} = \mathbf{w}] = f(\mathbf{w}\phi(\mathbf{x}))$ for a new observation $\mathbf{x} \in \mathcal{X}$.

The kernel trick is one of the most important applications of kernels. As will become clear, the kernel trick consists on adapting a learning technique so that it is only affected by observations through inner products, which can be substituted by corresponding kernel evaluations. More concretely, the kernel trick allows a learning technique to receive the Gram matrix $\mathbf{K}$ of a Mercer kernel κ for a dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathcal{X}$, instead of such observations (together with the corresponding targets, in the case of supervised learning). If applicable, the kernel trick also allows predicting the target corresponding to a new observation $\mathbf{x} \in \mathcal{X}$ based only on the vector $(\kappa(\mathbf{x}, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}, \mathbf{x}_N))$ and on the scalar $\kappa(\mathbf{x}, \mathbf{x})$.

Kernelized nearest neighbors is a simple classification technique based on the kernel trick. It is analogous to (K-)nearest neighbors, except for the fact that distances are computed using kernels.

Concretely, recall that the squared Euclidean distance $||\mathbf{x} - \mathbf{x}'||^2$ between vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^D$ is given by

$$||\mathbf{x} - \mathbf{x}'||^2 = (\mathbf{x} - \mathbf{x}')(\mathbf{x} - \mathbf{x}') = \mathbf{x}\mathbf{x} + \mathbf{x}'\mathbf{x}' - 2\mathbf{x}\mathbf{x}'.$$

Analogously, the squared distance between vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$ in an arbitrary inner product space $\mathcal{V}$ can be defined as

$$||\phi(\mathbf{x}) - \phi(\mathbf{x}')||^2 = \phi(\mathbf{x})\phi(\mathbf{x}) + \phi(\mathbf{x}')\phi(\mathbf{x}') - 2\phi(\mathbf{x})\phi(\mathbf{x}') = \kappa(\mathbf{x}, \mathbf{x}) + \kappa(\mathbf{x}', \mathbf{x}') - 2\kappa(\mathbf{x}, \mathbf{x}'),$$

where $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the Mercer kernel corresponding to the basis function $\phi : \mathcal{X} \to \mathcal{V}$, and $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ are the original observations.

It is crucial to notice that computing distances between vectors in $\mathcal{V}$ using a kernel κ does not require evaluating the corresponding basis function ϕ. This allows dealing with a potentially infinite dimensional inner product space $\mathcal{V}$, and considering arbitrary sets of observations (as long as an appropriate kernel can be devised).

Kernelized K-medoids is a kernelized iterative clustering technique similar to K-means. Consider the dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathcal{X}$. The technique initially chooses K so-called centroids randomly from the dataset $\mathcal{D}$ (without replacement). Each observation in $\mathcal{D}$ is then assigned to its closest centroid (according to a kernelized distance), which partitions the dataset $\mathcal{D}$ into K clusters (groups of observations). Each centroid is then changed to the observation whose average kernelized distance to the other observations in the same cluster is minimum. This allows assigning each observation in $\mathcal{D}$ to its closest centroid once again, and the process may continue until convergence (or for a maximum number of iterations).

The kernel trick can also be applied to (l^2-regularized) linear regression, as we now show. We start by reformulating the original optimization problem, which leads to the kernel-based method by analogy.

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$. In l^2-regularized linear regression, a maximum a posteriori estimate may be obtained by minimizing the cost J given by

$$J(\mathbf{w}) = \sum_{i=1}^{N}(y_i - \mathbf{w}\mathbf{x}_i)^2 + \lambda||\mathbf{w}|| = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda\mathbf{w}\mathbf{w},$$

where $\mathbf{X}$ is the $N \times D$ design matrix, $\mathbf{y} = (y_1, \ldots, y_N)$ is the target vector, and $\lambda \geq 0$ is a hyperparameter.

We have already shown that if $\lambda\boldsymbol{I} + \mathbf{X}^T\mathbf{X}$ is invertible, then the desired minimum $\mathbf{w}$ is given by

$$\mathbf{w} = (\lambda\boldsymbol{I} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Let $\boldsymbol{\alpha} = (\lambda\mathbf{I} + \mathbf{X}\mathbf{X}^T)^{-1}\mathbf{y}$. It can be shown that $\mathbf{w}$ can also be written as

$$\mathbf{w} = \mathbf{X}^T(\lambda\mathbf{I} + \mathbf{X}\mathbf{X}^T)^{-1}\mathbf{y} = \mathbf{X}^T\boldsymbol{\alpha} = \sum_{i=1}^{N}\alpha_i\mathbf{x}_i.$$

Finally, the prediction for a new observation $\mathbf{x} \in \mathbb{R}^D$ using the estimate $\mathbf{w} \in \mathbb{R}^D$ is given by

$$\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}, \mathbf{W} = \mathbf{w}] = \mathbf{w}\mathbf{x} = \sum_{i=1}^{N}\alpha_i\mathbf{x}_i\mathbf{x}.$$

We are now ready to introduce kernelized linear regression by analogy with linear regression. We omit a rigorous derivation of kernelized linear regression.

Let $\mathbf{K}$ be the Gram matrix of a Mercer kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ for the dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathbb{R}$. Furthermore, let $\phi : \mathcal{X} \to \mathcal{V}$ denote the basis function corresponding to κ.

If $\mathcal{X} \subseteq \mathbb{R}^D$, then $\mathbf{X}\mathbf{X}^T$ would be equivalent to the Gram matrix of a linear kernel for $\mathcal{D}$. By analogy, substituting $\mathbf{X}\mathbf{X}^T$ by the Gram matrix $\mathbf{K}$, it is possible to compute $\boldsymbol{\alpha} = (\lambda\mathbf{I} + \mathbf{K})^{-1}\mathbf{y}$. Although it is generally impossible to compute $\mathbf{w} \in \mathcal{V}$, we can compute the prediction $\mathbf{w}\phi(\mathbf{x})$ for a new observation $\mathbf{x} \in \mathcal{X}$ using

$$\mathbf{w}\phi(\mathbf{x}) = \sum_{i=1}^{N}\alpha_i\phi(\mathbf{x}_i)\phi(\mathbf{x}) = \sum_{i=1}^{N}\alpha_i\kappa(\mathbf{x}_i, \mathbf{x}).$$

In summary, it is possible to fit the model using only the Gram matrix $\mathbf{K}$ and the targets, and perform prediction for a new observation $\mathbf{x}$ using only $\kappa(\mathbf{x}_1, \mathbf{x}), \ldots, \kappa(\mathbf{x}_N, \mathbf{x})$.

The kernel trick can also be applied to principal component analysis, as we now show. Once again, we start by reformulating the original method, which leads to the kernel-based method by analogy.

Consider a dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathbb{R}^D$. The empirical covariance matrix $\boldsymbol{\Sigma}$ of $\mathcal{D}$ is given by

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = \frac{1}{N}\mathbf{X}^T\mathbf{C}\mathbf{X},$$

where $\mathbf{X}$ is the $N \times D$ design matrix corresponding to $\mathcal{D}$, and $\mathbf{C}$ is the $N \times N$ centering matrix given by

$$\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T,$$

where $\mathbf{I}$ is the $N \times N$ identity matrix, and $\mathbf{1} \in \mathbb{R}^N$ is a vector whose elements are all equal to one. It is worth mentioning that $\mathbf{CX}$ is the centered design matrix (whose columns sum to zero) corresponding to the design matrix $\mathbf{X}$, and that $\mathbf{C}$ is symmetric and idempotent.

Recall that principal component analysis finds a list containing orthonormal eigenvectors of the empirical covariance matrix $\boldsymbol{\Sigma}$ sorted in non-increasing order of corresponding eigenvalues. The kernel trick requires an alternative way to find such eigenvectors, which we now discuss.

Consider the $N \times N$ matrix $\mathbf{M} = \mathbf{CX}(\mathbf{CX})^T = \mathbf{CXX}^T\mathbf{C}^T = \mathbf{CXX}^T\mathbf{C}$. Because $\mathbf{M}$ is real symmetric, it can be written as $\mathbf{M} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where each column j of the $N \times N$ matrix $\boldsymbol{U}$ is an eigenvector $\mathbf{u}_j$ of $\mathbf{M}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix such that $\Lambda_{j,j} = \lambda_j$ is the eigenvalue that corresponds to $\mathbf{u}_j$. Furthermore, $\boldsymbol{U}$ is chosen so that $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$, which means that $(\mathbf{u}_1, \ldots, \mathbf{u}_N)$ is an orthonormal basis for $\mathbb{R}^N$.

By the definition of eigenvector, $\left(\mathbf{CXX}^T\mathbf{C}\right)\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda}$. By left-multiplying these expressions by $\mathbf{X}^T\mathbf{C}$,

$$\mathbf{X}^T\mathbf{C}\left(\mathbf{CXX}^T\mathbf{C}\right)\mathbf{U} = \mathbf{X}^T\mathbf{CX}\left(\mathbf{X}^T\mathbf{CU}\right) = N\boldsymbol{\Sigma}\left(\mathbf{X}^T\mathbf{CU}\right) = \left(\mathbf{X}^T\mathbf{CU}\right)\boldsymbol{\Lambda}$$

where we have used the fact that $\mathbf{CC} = \mathbf{C}$, and the definition of the empirical covariance matrix $\boldsymbol{\Sigma}$.

Therefore, each column j of $\mathbf{X}^T\mathbf{CU}$ corresponds to an eigenvector of the matrix $N\boldsymbol{\Sigma}$ with eigenvalue λ_j. This implies that each column j of $\mathbf{X}^T\mathbf{CU}$ corresponds to an eigenvector of the empirical covariance matrix $\boldsymbol{\Sigma}$ with eigenvalue $\frac{\lambda_j}{N}$. Although we omit the details, a $D \times K$ matrix $\mathbf{V}$ containing orthonormal eigenvectors of the empirical covariance matrix $\boldsymbol{\Sigma}$ can be obtained from these columns, where K is the rank of the centered design matrix $\mathbf{CX}$.

We are now ready to introduce kernel principal component analysis by analogy with principal component analysis. We omit a rigorous derivation of kernel principal component analysis.

Let $\mathbf{K}$ be the Gram matrix of an arbitrary Mercer kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ for the dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathcal{X}$. Once again, we use the fact that if $\mathcal{X} \subseteq \mathbb{R}^D$, then $\mathbf{XX}^T$ would be equivalent to the Gram matrix of a linear kernel for $\mathcal{D}$. By analogy, substituting $\mathbf{XX}^T$ by the Gram matrix $\mathbf{K}$, it is possible to compute the matrix $\mathbf{U}$ containing eigenvectors of $\mathbf{M} = \mathbf{CKC}$. Although it is generally impossible to compute the corresponding matrix $\mathbf{V}$ containing orthonormal eigenvectors of the empirical covariance matrix $\boldsymbol{\Sigma}$, it is possible to compute the alternative representation $\mathbf{z}$ of a new observation $\mathbf{x}$ using only $\kappa(\mathbf{x}_1, \mathbf{x}), \ldots, \kappa(\mathbf{x}_N, \mathbf{x})$, although we omit the details.

Support vector machines are supervised learning models that also benefit from the kernel trick. In what follows, we present both classification and regression based on these models.

Consider the dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^D$, and $y_i \in \{-1, 1\}$. A hard margin support vector machine is a classification technique that assumes that there is a hyperplane that separates the observations in such dataset by class. Furthermore, the technique finds the separating hyperplane such that its distance to the nearest observation is maximum, a choice motivated by statistical learning theory.

Any (affine) hyperplane S in $\mathbb{R}^D$ can be written as $S = \{\mathbf{x} \in \mathbb{R}^D \mid \mathbf{wx} + b = 0\}$, for some nonzero weight vector $\mathbf{w} \in \mathbb{R}^D$, and intercept $b \in \mathbb{R}$. Consider any $\mathbf{x}, \mathbf{x}' \in S$. Clearly, $\mathbf{wx} + b = \mathbf{wx}' + b$, which implies $\mathbf{w}(\mathbf{x} - \mathbf{x}') = 0$. Intuitively, $\mathbf{w}$ is orthogonal to any vector pointing between vectors in S.

Consider any vector $\mathbf{x} \in \mathbb{R}^D$, and let $\mathbf{x}_\perp$ denote its closest vector in S. Recall that $\mathbf{x}$ can be written as

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{||\mathbf{w}||},$$

where $r \in \mathbb{R}$ is the so-called signed distance between $\mathbf{x}$ and S. By introducing $\mathbf{w}$ and b into the equation above,

$$\mathbf{wx} + b = \mathbf{wx}_\perp + b + r\frac{||\mathbf{w}||^2}{||\mathbf{w}||}$$

$$\frac{\mathbf{wx} + b}{||\mathbf{w}||} = r,$$

since $\mathbf{wx}_\perp + b = 0$. Notice that the signed distance between S and the origin is given by $\frac{b}{||\mathbf{w}||}$.

Consider the dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, and the task of finding the parameters $\mathbf{w}$ and b of a separating hyperplane S such that $\mathbf{wx}_i + b > 0$ if $y_i = 1$ and $\mathbf{wx}_i + b < 0$ if $y_i = -1$, for all i, assuming that such hyperplane exists. This task is equivalent to finding parameters that satisfy the constraint $y_i(\mathbf{wx}_i + b) > 0$, for all i.

The margin $m(\mathbf{w}, b)$ of a hyperplane $S = \{\mathbf{x} \in \mathbb{R}^D \mid \mathbf{w}\mathbf{x} + b = 0\}$ that satisfies the constraints is defined as

$$m(\mathbf{w}, b) = \min_i \frac{y_i(\mathbf{w}\mathbf{x}_i + b)}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||} \min_i y_i(\mathbf{w}\mathbf{x}_i + b).$$

Because we assumed that the hyperplane S satisfies the constraints, the margin corresponds to the (unsigned) distance between the hyperplane S and the closest observation in $\mathcal{D}$.

Notice that if the hyperplane S defined by $\mathbf{w}$ and b satisfies the constraints, then the hyperplane S' defined by $\kappa\mathbf{w}$ and κb also satisfies the constraints, for any $\kappa > 0$. Both S and S' also have the same margin, since

$$m(\kappa\mathbf{w}, \kappa b) = \min_i \frac{y_i(\kappa\mathbf{w}\mathbf{x}_i + \kappa b)}{||\kappa\mathbf{w}||} = \frac{\kappa}{\kappa||\mathbf{w}||} \min_i y_i(\mathbf{w}\mathbf{x}_i + b) = \frac{1}{||\mathbf{w}||} \min_i y_i(\mathbf{w}\mathbf{x}_i + b) = m(\mathbf{w}, b).$$

Consider the task of finding the parameters $\mathbf{w}^*$ and b^* of a separating hyperplane with maximum margin. In other words, maximizing $m(\mathbf{w}, b)$ with respect to $\mathbf{w}$ and b subject to $y_i(\mathbf{w}\mathbf{x}_i + b) > 0$, for all i. The margin $m(\mathbf{w}^*, b^*)$ of such a hyperplane is given by

$$m(\mathbf{w}^*, b^*) = \frac{1}{||\mathbf{w}^*||} \min_i y_i(\mathbf{w}^*\mathbf{x}_i + b^*) = \max_{\mathbf{w}} \max_b \frac{1}{||\mathbf{w}||} \min_i y_i(\mathbf{w}\mathbf{x}_i + b).$$

Consider the task of maximizing $m(\mathbf{w}, b)$ with respect to $\mathbf{w}$ and b subject to $\min_i y_i(\mathbf{w}\mathbf{x}_i + b) = 1$. This constraint is clearly stronger than the constraint that $y_i(\mathbf{w}\mathbf{x}_i + b) > 0$, for all i. Suppose that a hyperplane S defined by $\mathbf{w}$ and b satisfies the weaker constraints, and let $\kappa = 1/\min_i y_i(\mathbf{w}\mathbf{x}_i + b)$, where the denominator is certainly positive. Consider the hyperplane S' defined by $\kappa\mathbf{w}$ and κb. As shown previously, $m(\mathbf{w}, b) = m(\kappa\mathbf{w}, \kappa b)$. Since

$$\min_i \kappa y_i(\mathbf{w}\mathbf{x}_i + b) = \min_i \frac{y_i(\mathbf{w}\mathbf{x}_i + b)}{\min_j y_j(\mathbf{w}\mathbf{x}_j + b)} = 1,$$

any hyperplane S that satisfies the weaker constraints has a corresponding hyperplane S' with the same margin that satisfies the stronger constraint. Thus, we can maximize $m(\mathbf{w}, b)$ with respect to $\mathbf{w}$ and b subject to the stronger constraint without loss of generality.

The constraint $\min_i y_i(\mathbf{w}\mathbf{x}_i + b) = 1$ implies that the maximum margin hyperplane defined by $\mathbf{w}^*$ and b^* is given by

$$m(\mathbf{w}^*, b^*) = \max_{\mathbf{w}} \max_b \frac{1}{||\mathbf{w}||} \min_i y_i(\mathbf{w}\mathbf{x}_i + b) = \max_{\mathbf{w}} \frac{1}{||\mathbf{w}||}.$$

Instead, we consider the equivalent, and more convenient, task of minimizing $\frac{1}{2}||\mathbf{w}||^2$ with respect to $\mathbf{w}$ and b subject to the same constraint.

We will change constraints one last time. Consider minimizing $\frac{1}{2}||\mathbf{w}||^2$ with respect to $\mathbf{w}$ and b subject to $y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1$, for all i. These constraints are apparently weaker than the previous. However, suppose that $\mathbf{w}$ and b minimize $\frac{1}{2}||\mathbf{w}||^2$, and $y_i(\mathbf{w}\mathbf{x}_i + b) > 1$, for all i. Let $\kappa = 1/\min_i y_i(\mathbf{w}\mathbf{x}_i + b)$. A hyperplane defined by $\kappa\mathbf{w}$ and κb satisfies the new constraints, since $y_i(\kappa\mathbf{w}\mathbf{x}_i + \kappa b) = \kappa y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1$, for all i. However, $\frac{1}{2}||k\mathbf{w}||^2 < \frac{1}{2}||\mathbf{w}||^2$, since $0 < \kappa < 1$, which is a contradiction because we assumed $\mathbf{w}$ and b corresponded to a minimum.. Therefore, a minimum that satisfies the new constraints also satisfies the previous constraint.

In summary, the parameters of the separating hyperplane with maximum margin are given by minimizing $\frac{1}{2}||\mathbf{w}||^2$ with respect to $\mathbf{w}$ and b subject to $y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1$, for all i. Given the optimum $\mathbf{w}$ and b, a new observation $\mathbf{x}$ can be classified as 1 if $\mathbf{w}\mathbf{x} + b > 0$, and as -1 otherwise.

The optimization task stated above is a quadratic programming problem, a class of widely studied optimization problems. However, the kernel trick requires restating this optimization task using the Lagrangian dual, which we cover next.

Let the generalized Lagrangian L be given by

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} a_i \big[y_i(\mathbf{w}\mathbf{x}_i + b) - 1 \big].$$

The Karush-Kuhn-Tucker conditions state that if $\mathbf{w}$ and b correspond to a local minimum subject to the constraints, then $\nabla L(\mathbf{w}, b, \mathbf{a}) = \mathbf{0}$ for some vector $\mathbf{a}$ whose elements are all nonnegative. The relevant gradients are given by

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \mathbf{a}) = \mathbf{w} - \sum_{i=1}^{N} a_i y_i \mathbf{x}_i$$

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \mathbf{a}) = \sum_{i=1}^{N} a_i y_i.$$

Therefore, if $\mathbf{w}$ and b correspond to a local minimum subject to the constraints, then

$$\mathbf{w} = \sum_{i=1}^{N} a_i y_i \mathbf{x}_i$$

$$0 = \sum_{i=1}^{N} a_i y_i,$$

for some $\mathbf{a}$ whose elements are all nonnegative. The Lagrangian dual $\tilde{L}$ is obtained by substituting the expressions above into the generalized Lagrangian L, and is given by

$$\tilde{L}(\mathbf{a}) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j \mathbf{x}_i \mathbf{x}_j.$$

It can be shown that if $\mathbf{a}$ maximizes $\tilde{L}$ subject to $a_i \geq 0$, for all i, and $\sum_i a_i y_i = 0$, then the corresponding $\mathbf{w} = \sum_i a_i y_i \mathbf{x}_i$ is the desired local minimum subject to constraints, and thus represent the maximum margin separating hyperplane.

The support vectors are the observations in $\mathcal{D}$ that are closest to the maximum margin separating hyperplane. It can be shown that only the coefficients a_i associated to these vectors are nonzero. Therefore, given the expression for $\mathbf{w}$ in terms of $\mathbf{a}$, only the support vectors directly affect classification. Intuitively, the other observations in $\mathcal{D}$ can move freely as long as they do not affect the margin, resulting in the same maximum margin separating hyperplane. The intercept b can be obtained by noting that $y_i(\mathbf{w}\mathbf{x}_i + b) = 1$, for any support vector $\mathbf{x}_i$.

Although maximizing the Lagrangian dual $\tilde{L}$ subject to the constraints is also a quadratic programming problem, the observations in $\mathcal{D}$ only affect $\tilde{L}$ through inner products.

Consider a Mercer kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and let $\boldsymbol{\phi} : \mathcal{X} \to \mathcal{V}$ denote the corresponding basis function. By using the fact that $\boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, it is possible to find the maximum margin separating hyperplane for observations transformed by $\boldsymbol{\phi}$ without evaluating $\boldsymbol{\phi}$. In that case, the maximum margin separating hyperplane in the transformed space would not necessarily be a hyperplane in the original space. The same trick also allows classifying new observations, since $\mathbf{w}\boldsymbol{\phi}(\mathbf{x}) + b = b + \sum_i a_i y_i \boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}) = b + \sum_i a_i y_i \kappa(\mathbf{x}_i, \mathbf{x})$, for any $\mathbf{x} \in \mathcal{X}$.

There are two very simple strategies to adapt support vector machines for classification problems with $C > 2$ classes. One of them is to train C one-vs-rest classifiers, and to classify a new observation by the class for which the observation is further away from the corresponding maximum margin separating hyperplane (on the correct side). Another heuristic is to train $C(C-1)/2$ one-vs-one classifiers, and to classify a new observation according to the class that receives more *votes*. Both strategies offer few guarantees in the general case.

A soft margin support vector machine is a technique that finds a separating hyperplane that may ignore some observations (subject to a penalty), as long as doing so increases the margin. Intuitively, this alternative formulation is more robust to outliers.

Recall that the parameters of the separating hyperplane with maximum margin are given by minimizing $\frac{1}{2}||\mathbf{w}||^2$ with respect to $\mathbf{w}$ and b subject to $y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1$, for all i.

In a soft margin support vector machine, the parameters $\mathbf{w}$ and b of the desired hyperplane are obtained by introducing a vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)$ containing so-called slack variables and minimizing

$$\frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \xi_i$$

with respect to $\mathbf{w}, b$ and $\boldsymbol{\xi}$, subject to $y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1 - \xi_i$, and $\xi_i \geq 0$, for all i. Intuitively, the hyperparameter $C \geq 0$ penalizes using the slack variables to *relax* the constraints. Soft margin support vector machines are also compatible with the kernel trick, although we omit the details.

Support vector machine regression predicts the target $y \in \mathbb{R}$ associated to an observation $\mathbf{x} \in \mathbb{R}^D$ using $y = \mathbf{w}\mathbf{x} + b$, where $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$ are the model parameters.

Consider a dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$. Support vector machine regression minimizes the cost function J given by

$$J(\mathbf{w}, b) = \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} L(y_i, \mathbf{w}\mathbf{x}_i + b),$$

where C is a hyperparameter, and $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is the ϵ-insensitive loss function given by

$$L(y, \hat{y}) = \begin{cases} 0 & \text{if } |y - \hat{y}| < \epsilon, \\ |y - \hat{y}| - \epsilon & \text{if } |y - \hat{y}| \geq \epsilon, \end{cases}$$

for some hyperparameter $\epsilon \geq 0$.

Equivalently, it is possible to introduce the vectors $\boldsymbol{\xi}^+, \boldsymbol{\xi}^- \in \mathbb{R}^N$ containing so-called slack variables, and minimize the cost function J given by

$$J(\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-) = \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \left(\xi_i^+ + \xi_i^- \right)$$

subject to the constraints $y_i \leq \mathbf{w}\mathbf{x}_i + b + \epsilon + \xi_i^+$, $y_i \geq \mathbf{w}\mathbf{x}_i + b - \epsilon - \xi_i^-$, $\xi_i^+ \geq 0$, and $\xi_i^- \geq 0$, for all i. Intuitively, the prediction for $\mathbf{x}_i$ must be in $[y_i - \epsilon - \xi_i^-, y_i + \epsilon + \xi_i^+]$, for all i.

The aforementioned optimization task is a quadratic programming problem. It is possible to show that the desired minimum $\mathbf{w}$ is given by $\mathbf{w} = \sum_i \alpha_i \mathbf{x}_i$, where $\boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^N$ is typically sparse. Therefore, the prediction y associated to an observation $\mathbf{x} \in \mathbb{R}^D$ is given by $\mathbf{w}\mathbf{x} + b = b + \sum_i \alpha_i \mathbf{x}_i \mathbf{x}$. Support vector machine regression is also compatible with the kernel trick, although we omit the details.

The term kernel has an additional meaning. A smoothing kernel is a zero-mean joint probability density function $\kappa : \mathbb{R}^D \to \mathbb{R}_{\geq 0}$. A typical example is the Gaussian smoothing kernel $\kappa : \mathbb{R}^D \to \mathbb{R}$ given by

$$\kappa(\mathbf{x}) = \frac{1}{h^D (2\pi)^{\frac{D}{2}}} \prod_{j=1}^{D} \exp \left(-\frac{x_j^2}{2h^2} \right),$$

where $h > 0$ is a hyperparameter called bandwidth.

Consider an iid dataset $\mathcal{D} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\mathbf{x}_i \in \mathbb{R}^D$. Kernel density estimation supposes that $\mathcal{D}$ is distributed according to a joint probability density function p given by

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \kappa(\mathbf{x} - \mathbf{x}_i),$$

where κ is a smoothing kernel. If κ is a Gaussian smoothing kernel, the model is analogous to a Gaussian mixture model composed of N multivariate Gaussian density functions whose means are given by the observations in $\mathcal{D}$.

Kernel regression is a regression technique based on modeling the joint probability density function over observations and targets using kernel density estimation, and predicting the target associated to an observation $\mathbf{x} \in \mathbb{R}^D$ using $\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]$.

15 Decision trees

In the context of supervised learning, a decision tree assigns observations to targets according to a sequence of logical tests that involve their features. We focus on building classification trees that perform inequality tests using the so-called CART (classification and regression trees) approach.

A classification tree is a full rooted binary tree $T = (V, E)$. By definition, T either has a single vertex (root), or can be built by connecting (by two edges) a single vertex (root) to the roots of two other binary trees.

Consider the task of classifying observations in $\mathbb{R}^D$. Each vertex u in the classification tree T is associated to a set $R_u \subseteq \mathbb{R}^D$. Suppose v and w are children of u. The vertex v is associated to the set $R_v = \{\mathbf{x} \in R_u \mid x_j < \tau\}$, and w is associated to the set $R_w = \{\mathbf{x} \in R_u \mid x_j \geq \tau\}$, for some feature j and constant threshold τ. The root r is associated to the set $R_r = \mathbb{R}^D$. Therefore, the leaves of T partition $\mathbb{R}^D$ into hyperrectangular regions. If each leaf of T is also associated to a class, an observation $\mathbf{x}$ can be classified by finding the leaf u such that $\mathbf{x} \in R_u$, by following the appropriate branches starting from the root.

Consider the dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, and the task of building a classification tree that is expected to generalize well to new observations. A typical strategy is to find a pure classification tree T, whose leaves are all pure. A leaf u is pure if R_u contains only observations from $\mathcal{D}$ that belong to the same class.

Suppose a dataset has as many distinct classes as there are observations. In that case, define the cost of a pure classification tree T for the dataset $\mathcal{D}$ as the cumulative number of tests required to classify each observation in $\mathcal{D}$

using T. Determining whether there is a classification tree with cost less or equal to k given $\mathcal{D}$ and pre-defined pairs of features and thresholds is an NP-complete problem, among other related problems. The computational burden of finding *optimal* trees is typically avoided by employing greedy heuristic methods.

We now describe a common heuristic to choose the feature j^* and threshold τ^* for the root of a classification tree given the dataset $\mathcal{D}$.

For each feature j and threshold τ, define the cost $C(j, \tau)$ of partitioning a dataset $\mathcal{D}$ into non-empty datasets $\mathcal{D}_v = \{(\mathbf{x}, y) \in \mathcal{D} \mid x_j < \tau\}$ and $\mathcal{D}_w = \mathcal{D} - \mathcal{D}_v = \{(\mathbf{x}, y) \in \mathcal{D} \mid x_j \geq \tau\}$ as

$$C(j, \tau) = c(\mathcal{D}_v) + c(\mathcal{D}_w),$$

where c is a pre-defined cost function, which we discuss later. We choose by exhaustive search a feature j^* and threshold τ^* such that

$$C(j^*, \tau^*) = \min_j \min_\tau C(j, \tau).$$

Notice that the cost C of at most $D(N-1)$ pairs of features and thresholds needs to be computed to find j^* and τ^*, even if the features are real-valued (since at most $N-1$ distinct thresholds would affect the partitioning of $\mathcal{D}$ into non-empty $\mathcal{D}_v$ and $\mathcal{D}_w$, for each feature).

A typical choice of cost function c is the entropy of the empirical distribution of classes in $\mathcal{D}$, given by

$$c(\mathcal{D}) = -\sum_y \pi_y \log \pi_y,$$

where π_y is the fraction of observations in $\mathcal{D}$ that belongs to class y, and $0 \log 0$ is substituted by 0. Intuitively, such entropy is minimized whenever all observations in $\mathcal{D}$ belong to the same class.

After choosing j^* and τ^* for the root vertex, its children u and w can be seen as roots of two distinct classification trees. By restricting each children v to its corresponding dataset $\mathcal{D}_v \subset R_v$, the procedure described above can be applied recursively. The procedure should not create children for a root when the dataset is already pure, since that would not affect future classifications. This completes a recursive algorithm for building a classification tree.

It is also possible to stop creating children whenever a pre-defined tree depth is achieved, or whenever the remaining dataset contains few observations. In such cases, a new observation $\mathbf{x}$ is classified according to the class majority in the leaf u such that $\mathbf{x} \in R_u$. The objective of these stop criteria is to make the classifier more robust to small changes in the training data. Another way to prevent overfitting is to *prune* the resulting classification tree, by eliminating branches according to their effect on training set accuracy. In all of these cases, the associated hyperparameters may be chosen by cross-validation.

Classification trees have some highly desirable properties. For instance, they are insensitive to monotonic transformation of features, since they are based on thresholds. More importantly, classification trees are more interpretable than many other classifiers, whose outputs cannot be easily understood in terms of the original features.

Consider sampling N elements (with replacement) from the elements in the dataset $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ to create each dataset in a sequence $\mathcal{D}_1, \ldots, \mathcal{D}_S$, and training a distinct classification tree for each dataset in this sequence. Suppose also that any observation $\mathbf{x} \in \mathbb{R}^D$ is classified according to the class that receives more votes from the S independent classifiers. This strategy is called bagging, and the meta-classifier is a type of ensemble. Intuitively, aggregating votes from classification trees trained using distinct datasets is typically more robust than depending on a classification tree that learned rules that may be overly specific for a particular dataset.

A random forest classifier is an ensemble of classification trees based on bagging. Each classification tree in the ensemble also considers only a random subset of $d < D$ features in each step of finding a (locally) optimum pair of feature j^* and threshold τ^*. As before, the goal is to create robust meta-classifiers. Random forests have shown remarkable empirical efficacy results.

License

References

[1] Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*, 2012.

[2] Bishop, Christopher M. *Pattern Recognition and Machine Learning*, Springer-Verlag New York, 2006.

[3] Hastie, T. and Tibshirani, R. and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition, Springer, 2009.

[4] Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*, 2009.

[5] Wasserman, L. *All of Statistics: A Concise Course in Statistical Inference.* Springer, 2010.

[6] Bertsekas, Dimitri P. *Nonlinear Programming*, 2nd edition, Athena Scientific, 1995.

[7] Meinshausen, Nicolai and Bühlmann, Peter. *Stability selection*, Journal of the Royal Statistical Society: Series B, 2010.