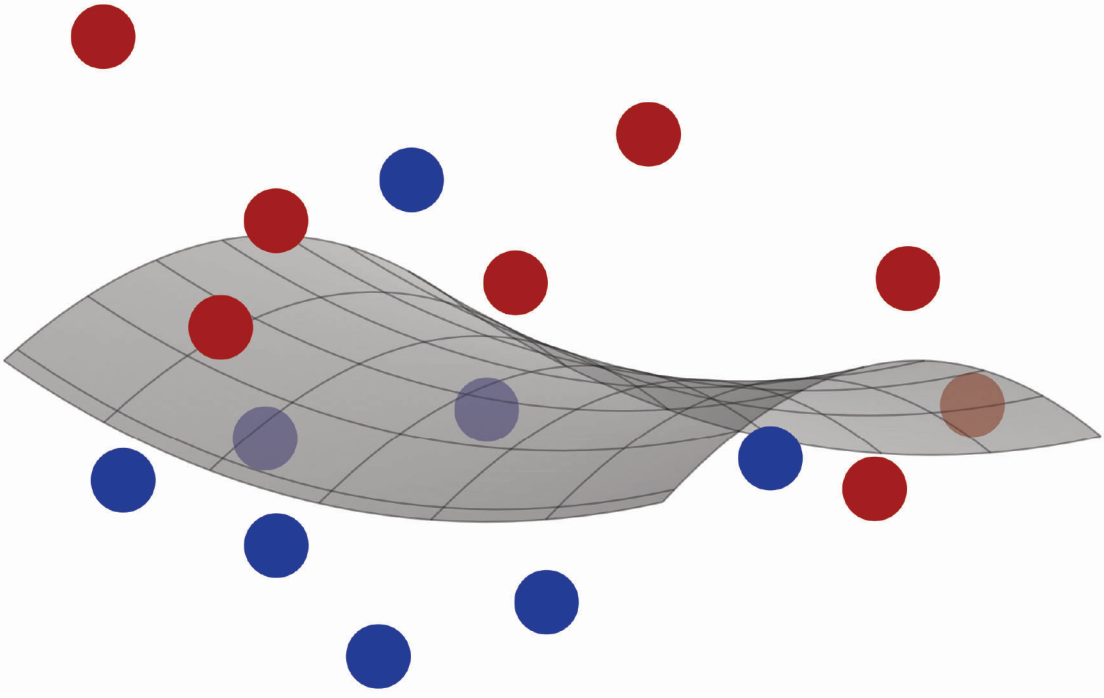
Foundations of Machine Learning

second edition



Mehryar Mohri,
Afshin Rostamizadeh,
and Ameet Talwalkar

13 Conditional Maximum Entropy Models

This chapter presents algorithms for estimating the conditional probability of a class given an example, rather than only predicting the class label for that example. This is motivated by several applications where confidence values are sought, in addition to the class prediction. The algorithms discussed, *conditional Maxent models*, also known as *multinomial logistic regression* algorithms, are among the most well-known and most widely used multi-class classification algorithms. In the special case of two classes, the algorithm is known as *logistic regression*.

As suggested by their name, these algorithms can be viewed as Maxent models for conditional probabilities. To introduce them, we will extend the ideas discussed in the previous chapter (Chapter 12), starting from an extension of the Maxent principle to the conditional case. Next, we will prove a duality theorem leading to an equivalent dual optimization problem for conditional Maxent. We will specifically discuss different aspects of multi-class classification using conditional Maxent and reserve a special section to the analysis of logistic regression.

13.1 Learning problem

We consider a multi-class classification problem with c classes, $c \geq 1$. Let $\mathcal{Y} = \{1, \ldots, c\}$ denote the output space and $\mathcal{D}$ a distribution over $\mathcal{X} \times \mathcal{Y}$. The learner receives a labeled training sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$ drawn i.i.d. according to $\mathcal{D}$. As in Chapter 12, we assume that, additionally, the learner has access to a feature mapping $\mathbf{\Phi} \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^N$ with $\mathbb{R}^N$ a normed vector space and with $\|\mathbf{\Phi}\|_\infty \leq r$. We will denote by $\mathcal{H}$ a family of real-valued functions containing the component feature functions Φ_j with $j \in [N]$. Note that in the most general case, we may have $N = +\infty$. The problem consists of using the training sample S to learn an accurate conditional probability $\mathsf{p}[\cdot|x]$, for any $x \in \mathcal{X}$.

13.2 Conditional Maxent principle

As for Maxent models, conditional Maxent or logistic regression models can be derived from a key concentration inequality. By the general Rademacher complexity bound (Theorem 3.3), for any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$ over the choice of a sample of size m:

$$\left\| \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\boldsymbol{\Phi}(x,y)] - \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}[\boldsymbol{\Phi}(x,y)] \right\|_\infty \leq 2\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}, \tag{13.1}$$

where we denote by $\widehat{\mathcal{D}}$ the empirical distribution defined by the sample S. We will also denote by $\widehat{\mathcal{D}}^1(x)$ the empirical distribution of x in the sample S. For any $x \in \mathcal{X}$, let $\mathsf{p}_0[\cdot|x]$ denote a conditional probability, often chosen to be the uniform distribution. Then, the *conditional Maxent principle* consists of seeking conditional probabilities $\mathsf{p}[\cdot|x]$ that are as agnostic as possible, that is as close as possible to the uniform distribution, or, more generally, to priors $\mathsf{p}_0[\cdot|x]$, while verifying an inequality similar to (13.1):

$$\left\| \mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1 \\ y\sim\mathsf{p}[\cdot|x]}}[\boldsymbol{\Phi}(x,y)] - \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}[\boldsymbol{\Phi}(x,y)] \right\|_\infty \leq \lambda, \tag{13.2}$$

where $\lambda \geq 0$ is a parameter. Here, closeness is defined via the *conditional relative entropy* (appendix E) based on the empirical marginal distribution $\widehat{\mathcal{D}}^1$ of input points. Choosing $\lambda = 0$ corresponds to standard *conditional Maxent* or *unregularized conditional Maxent* and to requiring the expectation of the features based on $\mathcal{D}^1$ and the conditional probabilities $\mathsf{p}[\cdot|x]$ to precisely match the empirical averages. As we will see later, its relaxation, that is the inequality case ($\lambda \neq 0$), translates into a regularization. Notice that the conditional Maxent principle does not require specifying a family of conditional probability distributions $\mathcal{P}$ to choose from.

13.3 Conditional Maxent models

Let Δ denote the simplex of the probability distributions over $\mathcal{Y}$, $\mathcal{X}_1 = \mathrm{supp}(\widehat{\mathcal{D}}^1)$ the support of $\widehat{\mathcal{D}}^1$, and $\bar{\mathsf{p}} \in \Delta^{\mathcal{X}_1}$ the family of conditional probabilities, $\bar{\mathsf{p}} = (\mathsf{p}[\cdot|x])_{x\in\mathcal{X}_1}$. Then, the conditional Maxent principle can be formulated as the following optimiza-

tion problem:

$$\min_{\bar{\mathsf{p}} \in \Delta^{\mathcal{X}_1}} \sum_{x \in \mathcal{X}_1} \widehat{\mathcal{D}}^1(x)\, \mathsf{D}\big(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big) \tag{13.3}$$

$$\text{s.t.} \left\| \mathop{\mathbb{E}}_{\substack{x \sim \widehat{\mathcal{D}}^1 \\ y \sim \mathsf{p}[\cdot|x]}} [\boldsymbol{\Phi}(x,y)] - \mathop{\mathbb{E}}_{(x,y) \sim \widehat{\mathcal{D}}} [\boldsymbol{\Phi}(x,y)] \right\|_\infty \leq \lambda.$$

This defines a convex optimization problem since the objective is a positive sum of relative entropies and since the relative entropy D is convex with respect to its arguments (appendix E), since the constraints are affine functions of $\bar{\mathsf{p}}$, and since $\Delta^{\mathcal{X}_1}$ is a convex set. The solution is in fact unique, since the objective is strictly convex as a positive sum of relative entropies, each strictly convex. The empirical conditional probabilities $\widehat{\mathcal{D}}^1(\cdot|x)$, $x \in \mathcal{X}_1$, clearly form a feasible solution, thus problem (12.7) is feasible.

For uniform priors $\mathsf{p}_0[\cdot|x]$, problem (13.3) can be equivalently formulated as a conditional entropy maximization, which explains the name given to these models. Let $\bar{\mathsf{H}}(\bar{\mathsf{p}}) = -\mathbb{E}_{x \sim \widehat{\mathcal{D}}^1} \big[\sum_{y \in \mathcal{Y}} \mathsf{p}[y|x] \log \mathsf{p}[y|x] \big]$ denote the conditional entropy of p with respect to the marginal $\widehat{\mathcal{D}}^1$. Then, the objective function of (12.7) can be rewritten as follows:

$$\begin{aligned}
\mathsf{D}\big(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big) &= \mathop{\mathbb{E}}_{x \sim \widehat{\mathcal{D}}^1} \left[\sum_{y \in \mathcal{Y}} \mathsf{p}[y|x] \log \frac{\mathsf{p}[y|x]}{\mathsf{p}_0[y|x]} \right] \\
&= \mathop{\mathbb{E}}_{x \sim \widehat{\mathcal{D}}^1} \left[-\sum_{y \in \mathcal{Y}} \mathsf{p}[y|x] \log(1/c) + \sum_{y \in \mathcal{Y}} \mathsf{p}[y|x] \log \mathsf{p}[y|x] \right] \\
&= \log(c) - \bar{\mathsf{H}}(\bar{\mathsf{p}}).
\end{aligned}$$

Thus, since $\log(c)$ is a constant, minimizing the objective is then equivalent to maximizing $\bar{\mathsf{H}}(\bar{\mathsf{p}})$.

Conditional Maxent models are the solutions of the optimization problem just described. As in the non-conditional case, they admit two important benefits: they are based on a fundamental theoretical guarantee of closeness of empirical and true feature averages, and they do not require specifying a particular family of distributions $\mathcal{P}$. In the next sections, we will further analyze the properties of conditional Maxent models.

13.4 Dual problem

Here, we derive an equivalent dual problem for (13.3) which, as we will show, can be formulated as a regularized conditional maximum likelihood problem over the family of *Gibbs distributions*.

The Maxent optimization problem (13.3) can be equivalently expressed as the unconstrained optimization problem $\min_{\bar{\mathsf{p}}} F(\bar{\mathsf{p}})$ with, for all $\bar{\mathsf{p}} = (\mathsf{p}[\cdot|x] \in (\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}$,

$$F(\bar{\mathsf{p}}) = \mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}^1}\left[\widetilde{\mathsf{D}}\big(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big)\right] + I_{\mathcal{C}}\left(\mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}[\cdot|x]}}[\boldsymbol{\Phi}(x,y)]\right), \tag{13.4}$$

with $\widetilde{\mathsf{D}}\big(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big) = \mathsf{D}\big(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big)$ if $\mathsf{p}[\cdot|x]$ is in Δ, $\widetilde{\mathsf{D}}\big(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big) = +\infty$ otherwise, and with $\mathcal{C} = \{\mathbf{u} \in \mathbb{R}^N \colon \|\mathbf{u} - \mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}}[\boldsymbol{\Phi}(x,y)]\|_\infty \leq \lambda\}$, which is a convex set.

Let G be the function defined for all $\mathbf{w} \in \mathbb{R}^N$ by

$$G(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} \log\left[\frac{\mathsf{p}_{\mathbf{w}}[y_i|x_i]}{\mathsf{p}_0[y_i|x_i]}\right] - \lambda\|\mathbf{w}\|_1\,, \tag{13.5}$$

with, for all $x \in \mathcal{X}_1$ and $y \in \mathcal{Y}$,

$$\mathsf{p}_{\mathbf{w}}[y|x] = \frac{\mathsf{p}_0[y|x]e^{\mathbf{w}\cdot\boldsymbol{\Phi}(x,y)}}{Z(\mathbf{w},x)} \quad \text{and} \quad Z(\mathbf{w},x) = \sum_{y\in\mathcal{Y}} \mathsf{p}_0[y|x]e^{\mathbf{w}\cdot\boldsymbol{\Phi}(x,y)}. \tag{13.6}$$

Then, the following theorem gives a result similar to the duality theorem presented in the non-conditional case (Theorem 12.2, Section 12.5).

Theorem 13.1 *Problem* (13.3) *is equivalent to the dual optimization problem* $\sup_{\mathbf{w}\in\mathbb{R}^N} G(\mathbf{w})$*:*

$$\sup_{\mathbf{w}\in\mathbb{R}^N} G(\mathbf{w}) = \min_{\bar{\mathsf{p}}\in(\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}} F(\bar{\mathsf{p}}). \tag{13.7}$$

Furthermore, let $\bar{\mathsf{p}}^* = \operatorname{argmin}_{\bar{\mathsf{p}}} F(\bar{\mathsf{p}})$*. Then, for any* $\epsilon > 0$ *and any* $\mathbf{w}$ *such that* $|G(\mathbf{w}) - \sup_{\mathbf{w}\in\mathbb{R}^N} G(\mathbf{w})| < \epsilon$*, we have* $\mathbb{E}_{x\sim\widehat{\mathcal{D}}^1}\left[\mathsf{D}\big(\bar{\mathsf{p}}^*[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big)\right] \leq \epsilon$*.*

The proof is similar to that of Theorem 12.2 and is given at the end of this chapter since it is somewhat longer (Section 13.9).

In view of the theorem, if $\mathbf{w}$ is an ϵ-solution of the dual optimization problem, then $\mathbb{E}_{x\sim\widehat{\mathcal{D}}^1}\left[\mathsf{D}\big(\mathsf{p}^*[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big)\right] \leq \epsilon$, which, by Jensen's inequality and Pinsker's inequality (Proposition E.7) implies that

$$\mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}^1}\left[\big\|\mathsf{p}^*[\cdot|x] - \mathsf{p}_{\mathbf{w}}[\cdot|x]\big\|_1\right] \leq \sqrt{\mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}^1}\left[\big\|\mathsf{p}^*[\cdot|x] - \mathsf{p}_{\mathbf{w}}[\cdot|x]\big\|_1^2\right]} \leq \sqrt{2\epsilon}.$$

Thus, $\mathsf{p}_{\mathbf{w}}[\cdot|x]$ is then $\sqrt{2\epsilon}$-close in $\widehat{\mathcal{D}}^1$-averaged L_1-norm to the optimal solution of the primal and the theorem suggests that the solution of the conditional Maxent problem can be determined by solving the dual problem, which can be written equivalently as follows for a uniform prior:

$$\inf_{\mathbf{w}} \lambda\|\mathbf{w}\|_1 - \frac{1}{m}\sum_{i=1}^{m} \log\big[\mathsf{p}_{\mathbf{w}}[y_i|x_i]\big]. \tag{13.8}$$

Similar remarks to those made for non-conditional Maxent models apply here. In particular, the solution may not be achieved for any finite $\mathbf{w}$ for $\lambda = 0$, which is why the infimum is needed. Also, this result may seem surprising since it shows that conditional Maxent coincides with conditional Maximum Likelihood ($\lambda = 0$) or regularized conditional Maximum Likelihood ($\lambda > 0$) using for the family $\mathcal{P}$ of conditional probabilities to choose from that of Gibbs distributions, while the conditional Maxent principle does not explicitly specify any family of conditional probabilities $\mathcal{P}$. The reason is the specific choice of the conditional relative entropy as the measure of closeness of $\mathsf{p}[\cdot|x]$ to the prior conditional distributions $\mathsf{p}_0[\cdot|x]$. Other measures of closeness between distributions lead to different forms for the solution. Thus, in some sense, the choice of the measure of closeness is the (dual) counterpart of that of the family of conditional distributions in maximum likelihood. Also, as already mentioned in the standard Maxent case, Gibbs distributions form a very rich family.

Notice that both the primal and the dual optimization problems for conditional Maxent involve only conditional probabilities $\mathsf{p}[\cdot|x]$ for x in $\mathcal{X}_1$, that is for x in the training sample. Thus, they do not provide us with any information about other conditional probabilities. However, the dual shows that, for x in $\mathcal{X}_1$, the solution admits the same general form $\mathsf{p}_{\mathbf{w}}[\cdot|x]$, which only depends on the weight vector $\mathbf{w}$. In view of that, we extend the definition of Maxent conditional probabilities to all $x \in \mathcal{X}$ by using the same general form $\mathsf{p}_{\mathbf{w}}[\cdot|x]$ and the same vector $\mathbf{w}$ for all x.

Observe also that in the definition of the primal or dual problems we could have used some other distribution $\mathcal{Q}$ over $\mathcal{X}$ in lieu of $\widehat{\mathcal{D}}^1$. It is straightforward to verify that the duality theorem would continue to hold in that case using the same proof. In fact, ideally, we would have chosen $\mathcal{Q}$ to be $\mathcal{D}^1$. However, that optimization problem would require knowledge of the feature vectors for all $x \in \mathrm{supp}(\mathcal{D}^1)$, which of course is not accessible to us given a finite sample. The weighted vector $\mathbf{w}$ found when using $\widehat{\mathcal{D}}^1$ can be viewed as an approximation of the one obtained if using $\mathcal{D}^1$.

13.5 Properties

In this section, we discuss several aspects of conditional Maxent models, including the form of the dual optimization problems, the feature vectors used, and prediction with these models.

13.5.1 Optimization problem

L_1-regularized conditional Maxent models are therefore conditional probability models solutions of the primal problem (13.3) or, equivalently, models defined by

$$\mathsf{p}_{\mathbf{w}}[y|x] = \frac{e^{\mathbf{w}\cdot\mathbf{\Phi}(x,y)}}{Z(x)} \quad \text{and} \quad Z(x) = \sum_{y\in\mathcal{Y}} e^{\mathbf{w}\cdot\mathbf{\Phi}(x,y)}, \tag{13.9}$$

where $\mathbf{w}$ is solution of the dual problem

$$\min_{\mathbf{w}\in\mathbb{R}^N} \lambda\|\mathbf{w}\|_1 - \frac{1}{m}\sum_{i=1}^{m} \log \mathsf{p}_{\mathbf{w}}[y_i|x_i],$$

with $\lambda \geq 0$ is a parameter. Using the expression of the conditional probabilities, this optimization problem can be written more explicitly as

$$\min_{\mathbf{w}\in\mathbb{R}^N} \lambda\|\mathbf{w}\|_1 + \frac{1}{m}\sum_{i=1}^{m} \log\left[\sum_{y\in\mathcal{Y}} e^{\mathbf{w}\cdot\mathbf{\Phi}(x_i,y)-\mathbf{w}\cdot\mathbf{\Phi}(x_i,y_i)}\right]. \tag{13.10}$$

or, equivalently, as

$$\min_{\mathbf{w}\in\mathbb{R}^N} \lambda\|\mathbf{w}\|_1 - \mathbf{w}\cdot\frac{1}{m}\sum_{i=1}^{m}\mathbf{\Phi}(x_i,y_i) + \frac{1}{m}\sum_{i=1}^{m} \log\left[\sum_{y\in\mathcal{Y}} e^{\mathbf{w}\cdot\mathbf{\Phi}(x_i,y)}\right]. \tag{13.11}$$

By definition of the dual problem, this is an unconstrained convex optimization problem in $\mathbf{w}$. This can be also seen from the fact that the log-sum function $\mathbf{w} \mapsto \log\left[\sum_{y\in\mathcal{Y}} e^{\mathbf{w}\cdot\mathbf{\Phi}(x,y)}\right]$ is convex for any $x \in \mathcal{X}$.

There are many optimization solutions available for this problem, including several special-purpose algorithms, general first-order and second-order solutions, and special-purpose distributed solutions. One common method is simply to use stochastic gradient descent (SGD), which has been reported to be more efficient than most special-purpose methods in applications. When the dimension of the feature vectors $\mathbf{\Phi}$ (or the cardinality of the family of feature functions $\mathcal{H}$) is very large, these methods are typically inefficient. An alternative method then consists of applying coordinate descent to solve this problem. In that case, the resulting algorithm coincides with the version of L_1-regularized boosting where, instead of the exponential function, the logistic function is used.

13.5.2 Feature vectors

Using feature vectors $\mathbf{\Phi}(x,y)$ depending on both the input x and the output y is often important in applications. For example, in machine translation, it is convenient to use features whose values may depend on the presence of some words in the input sentence and some others in the output sequence. A common choice of the feature vector is however one where the column vectors $\mathbf{\Phi}(x,y)$ and $\mathbf{w}$ admit c

blocks of equal size and where only the block in $\mathbf{\Phi}(x, y)$ corresponding to the class y is non-zero and equal to a feature vector $\mathbf{\Gamma}(x)$ independent of the class labels:

$$\mathbf{\Phi}(x, y) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{\Gamma}(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_{y-1} \\ \mathbf{w}_y \\ \mathbf{w}_{y+1} \\ \vdots \\ \mathbf{w}_c \end{bmatrix}.$$

In view of that, the inner product of $\mathbf{w}$ and $\mathbf{\Phi}(x, y)$ can be expressed in terms of the feature vector $\mathbf{\Gamma}(x)$, which only depends on x, but with a distinct parameter vector $\mathbf{w}_y$:

$$\mathbf{w} \cdot \mathbf{\Phi}(x, y) = \mathbf{w}_y \cdot \mathbf{\Gamma}(x).$$

The optimization problem for L_1-regularized conditional Maxent can then be written in terms of the vectors $\mathbf{w}_y$ as follows:

$$\min_{\mathbf{w} \in \mathbb{R}^N} \lambda \sum_{y \in \mathcal{Y}} \|\mathbf{w}_y\|_1 + \frac{1}{m} \sum_{i=1}^{m} \log \left[\sum_{y \in \mathcal{Y}} e^{\mathbf{w}_y \cdot \mathbf{\Gamma}(x_i) - \mathbf{w}_{y_i} \cdot \mathbf{\Gamma}(x_i)} \right]. \tag{13.12}$$

Notice that, if the vectors $\mathbf{w}_y$ were not correlated via the second term of the objective function (for example if, instead of the log of the sum, this term were replaced by the sum of the logs), then the problem would be reduced to c separate optimization functions learning a distinct weight vector for each class, as in the one-vs-all setup of multi-class classification.

13.5.3 Prediction

Finally, note that the class $\widehat{y}(x)$ predicted by a conditional Maxent model with parameter $\mathbf{w}$ is given by

$$\widehat{y}(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} \mathsf{p}_{\mathbf{w}}[y|x] = \operatorname*{argmax}_{y \in \mathcal{Y}} \mathbf{w} \cdot \mathbf{\Phi}(x, y). \tag{13.13}$$

Thus, conditional Maxent models define linear classifiers. Conditional Maxent models are also sometimes referred to as log-*linear models*.

13.6 Generalization bounds

In this section, we will present learning guarantees for conditional Maxent models in two different settings: one where the dimension of the feature vectors $\mathbf{\Phi}$ (or the cardinality of the family of feature functions $\mathcal{H}$) is infinite or extremely large and where a coordinate-descent or boosting-type algorithm is more suitable, and another one where the dimension of the feature vectors $\mathbf{\Phi}$ is finite and not too large.

We start with the case where the dimension of the feature vectors $\mathbf{\Phi}$ is very large. The following margin-based guarantee holds in that case.

Theorem 13.2 *For any $\delta > 0$, with probability at least $1 - \delta$ over the draw of an i.i.d. sample S of size m, the following holds for all $\rho > 0$ and $f \in \mathcal{F} = \{(x, y) \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x, y) \colon \|\mathbf{w}\|_1 \leq 1\}$:*

$$R(f) \leq \frac{1}{m} \sum_{i=1}^{m} \log_{u_0}\left(\sum_{y \in \mathcal{Y}} e^{\frac{f(x_i,y) - f(x_i,y_i)}{\rho}} \right) + \frac{8c}{\rho} \mathfrak{R}_m(\Pi_1(\mathcal{H})) + \sqrt{\frac{\log \log_2 \frac{4r}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}},$$

where $u_0 = \log(1 + 1/e)$ and $\Pi_1(\mathcal{H}) = \{x \mapsto \phi(x, y) \colon \phi \in \mathcal{H}, y \in \mathcal{Y}\}$.

Proof: For any $f \colon (x, y) \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x, y)$ and $i \in [m]$, let $\rho_f(x_i, y_i)$ denote the margin of f at (x_i, y_i):

$$\rho_f(x_i, y_i) = \min_{y \neq y_i} f(x_i, y_i) - f(x_i, y) = \min_{y \neq y_i} \mathbf{w} \cdot (\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, y)).$$

Fix $\rho > 0$. Then, by Theorem 9.2, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds for all $f \in \mathcal{H}$ and $\rho \in (0, 2r]$:

$$R(f) \leq \frac{1}{m} \sum_{i=1}^{m} 1_{\rho_f(x_i,y_i) \leq \rho} + \frac{4c}{\rho} \mathfrak{R}_m(\Pi_1(\mathcal{F})) + \sqrt{\frac{\log \log_2 \frac{4r}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}},$$

where $\Pi_1(\mathcal{F}) = \{x \mapsto f(x, y) \colon y \in \mathcal{Y}, f \in \mathcal{H}\}$. The inequality trivially holds for all $\rho > 0$ since for $\rho \geq 2r$, by Hölder's inequality, we have $|\mathbf{w} \cdot \mathbf{\Phi}(x, y)| \leq \|\mathbf{w}\|_1 \|\mathbf{\Phi}(x, y)\|_\infty \leq r$ for $\|\mathbf{w}\|_1 \leq 1$, and thus $\min_{y \neq y_i} f(x_i, y_i) - f(x_i, y) \leq 2r \leq \rho$ for all $i \in [m]$ and $y \in \mathcal{Y}$. Now, for any $\rho > 0$, the ρ-margin loss can be upper bounded by the ρ-logistic loss:

$$\forall u \in \mathbb{R}, 1_{u \leq \rho} = 1_{\frac{u}{\rho} - 1 \leq 0} \leq \log_{u_0}(1 + e^{-\frac{u}{\rho}}).$$

Thus, the ρ-margin loss of f at (x_i, y_i) can be upper bounded as follows:

$$\begin{aligned}
1_{\rho_f(x_i,y_i) \leq \rho} &\leq \log_{u_0}(1 + e^{-\frac{\rho(f, x_i, y_i)}{\rho}}) \\
&= \log_{u_0}(1 + \max_{y \neq y_i} e^{\frac{f(x_i,y) - f(x_i,y_i)}{\rho}}) \\
&\leq \log_{u_0}\left(1 + \sum_{y \neq y_i} e^{\frac{f(x_i,y) - f(x_i,y_i)}{\rho}}\right) = \log_{u_0}\left(\sum_{y \in \mathcal{Y}} e^{\frac{f(x_i,y) - f(x_i,y_i)}{\rho}}\right).
\end{aligned}$$

Thus, with probability at least $1 - \delta$, the following inequality holds for all $f \in \mathcal{H}$ and $\rho > 0$:

$$R(f) \leq \frac{1}{m} \sum_{i=1}^{m} \log_{u_0}\left(\sum_{y \in \mathcal{Y}} e^{\frac{f(x_i,y) - f(x_i,y_i)}{\rho}} \right) + \frac{4c}{\rho} \mathfrak{R}_m(\Pi_1(\mathcal{F})) + \sqrt{\frac{\log \log_2 \frac{4r}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

For any sample $S = (x_1, \ldots, x_m)$ of size m, the empirical Rademacher complexity of $\Pi_1(\mathcal{F})$ can be bounded as follows:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(\Pi_1(\mathcal{F})) &= \frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{\|\mathbf{w}\|_1 \leq 1 \\ y \in \mathcal{Y}}} \sum_{i=1}^{m} \sigma_i \sum_{j=1}^{N} w_j \Phi_j(x_i, y)\right] \\
&= \frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{\|\mathbf{w}\|_1 \leq 1 \\ y \in \mathcal{Y}}} \sum_{j=1}^{N} w_j \sum_{i=1}^{m} \sigma_i \Phi_j(x_i, y)\right] \\
&= \frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{j \in [N] \\ y \in \mathcal{Y}}} \left|\sum_{i=1}^{m} \sigma_i \Phi_j(x_i, y)\right|\right] \\
&\leq \frac{1}{m}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\substack{\Phi \in \mathcal{H} \\ y \in \mathcal{Y}}} \left|\sum_{i=1}^{m} \sigma_i \Phi(x_i, y)\right|\right] \leq 2\,\widehat{\mathfrak{R}}_S(\Pi_1(\mathcal{H})),
\end{aligned}$$

which completes the proof. □

The learning guarantee of the theorem is remarkable since it does not depend on the dimension N and since it only depends on the complexity of the family $\mathcal{H}$ of feature functions (or base hypotheses). Since for any $\rho > 0$, f/ρ admits the same generalization error as f, the theorem implies that with probability at least $1 - \delta$, the following inequality holds for all $f \in \{(x, y) \mapsto \mathbf{w} \cdot \boldsymbol{\Phi}(x, y) \colon \|\mathbf{w}\|_1 \leq \frac{1}{\rho}\}$ and $\rho > 0$:

$$R(f) \leq \frac{1}{m}\sum_{i=1}^{m} \log_{u_0}\left(\sum_{y \in \mathcal{Y}} e^{f(x_i, y) - f(x_i, y_i)}\right) + \frac{8c}{\rho}\mathfrak{R}_m(\Pi_1(\mathcal{H})) + \sqrt{\frac{\log\log_2 \frac{4r}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

This inequality can be used to derive an algorithm that selects $\mathbf{w}$ and $\rho > 0$ to minimize the right-hand side. The minimization with respect to ρ does not lead to a convex optimization and depends on theoretical constant factors affecting the second and third term. Thus, instead, ρ is left as a free parameter of the algorithm, typically determined via cross-validation.

Now, since only the first term of the right-hand side depends on $\mathbf{w}$, for any $\rho > 0$, the bound suggests selecting $\mathbf{w}$ as the solution of the following optimization problem:

$$\min_{\|\mathbf{w}\|_1 \leq \frac{1}{\rho}} \frac{1}{m}\sum_{i=1}^{m} \log\left(\sum_{y \in \mathcal{Y}} e^{\mathbf{w} \cdot \boldsymbol{\Phi}(x_i, y) - \mathbf{w} \cdot \boldsymbol{\Phi}(x_i, y_i)}\right). \tag{13.14}$$

Introducing a Lagrange variable $\lambda \geq 0$, the optimization problem can be written equivalently as

$$\min_{\mathbf{w}} \lambda \|\mathbf{w}\|_1 + \frac{1}{m} \sum_{i=1}^{m} \log \left(\sum_{y \in \mathcal{Y}} e^{\mathbf{w} \cdot \mathbf{\Phi}(x_i, y) - \mathbf{w} \cdot \mathbf{\Phi}(x_i, y_i)} \right). \tag{13.15}$$

Since for any choice of ρ in the constraint of (13.14), there exists an equivalent dual variable λ in the formulation of (13.15) that achieves the same optimal $\mathbf{w}$, λ can be freely selected via cross-validation. The resulting algorithm precisely coincides with conditional Maxent.

When the dimension N of the feature vectors $\mathbf{\Phi}$ is finite, the following margin-based guarantee holds.

Theorem 13.3 *For any $\delta > 0$, with probability at least $1 - \delta$ over the draw of an i.i.d. sample S of size m, the following holds for all $\rho > 0$ and $f \in \mathcal{F} = \{(x, y) \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x, y) \colon \|\mathbf{w}\|_1 \leq 1\}$:*

$$R(f) \leq \frac{1}{m} \sum_{i=1}^{m} \log_{u_0} \left(\sum_{y \in \mathcal{Y}} e^{\frac{f(x_i, y) - f(x_i, y_i)}{\rho}} \right) + \frac{4cr\sqrt{2 \log(2cN)}}{\rho} + \sqrt{\frac{\log \log_2 \frac{4r}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}},$$

where $u_0 = \log(1 + 1/e)$.

Proof: The proof coincides with that of Theorem 13.2, modulo the upper bound on $\mathfrak{R}_m(\Pi_1(\mathcal{F}))$. For any sample $S = (x_1, \ldots, x_m)$ of size m, the empirical Rademacher complexity of $\Pi_1(\mathcal{F})$ can be bounded as follows:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(\Pi_1(\mathcal{F})) &= \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\substack{\|\mathbf{w}\|_1 \leq 1 \\ y \in \mathcal{Y}}} \sum_{i=1}^{m} \sigma_i \mathbf{w} \cdot \mathbf{\Phi}(x_i, y) \right] \\
&= \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\substack{\|\mathbf{w}\|_1 \leq 1 \\ y \in \mathcal{Y}}} \mathbf{w} \cdot \sum_{i=1}^{m} \sigma_i \mathbf{\Phi}(x_i, y) \right] \\
&= \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{y \in \mathcal{Y}} \left\| \sum_{i=1}^{m} \sigma_i \mathbf{\Phi}(x_i, y) \right\|_\infty \right] \\
&= \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\substack{j \in [N] \\ y \in \mathcal{Y}, s \in \{-1, +1\}}} s \sum_{i=1}^{m} \sigma_i \Phi_j(x_i, y) \right] \\
&\leq r\sqrt{2 \log(2cN)},
\end{aligned}$$

where the third equality holds by definition of the dual norm, and the last inequality by the maximal inequality (Corollary D.11), since the supremum is taken over $2cN$ choices. □

This learning guarantee of the theorem is very favorable even for relatively high-dimensional problems since its dependency on the dimension N is only logarithmic.

13.7 Logistic regression

The binary case of conditional Maxent models ($c = 2$) is known as *logistic regression* and is one of the most well-known algorithms for binary classification.

13.7.1 Optimization problem

In the binary case, the sum appearing in the optimization problem of conditional Maxent models can be simplified as follows:

$$\begin{aligned}\sum_{y\in\mathcal{Y}} e^{\mathbf{w}\cdot\boldsymbol{\Phi}(x_i,y)-\mathbf{w}\cdot\boldsymbol{\Phi}(x_i,y_i)} &= e^{\mathbf{w}\cdot\boldsymbol{\Phi}(x_i,+1)-\mathbf{w}\cdot\boldsymbol{\Phi}(x_i,y_i)} + e^{\mathbf{w}\cdot\boldsymbol{\Phi}(x_i,-1)-\mathbf{w}\cdot\boldsymbol{\Phi}(x_i,y_i)}\\ &= 1 + e^{-y_i\mathbf{w}\cdot[\boldsymbol{\Phi}(x_i,+1)-\boldsymbol{\Phi}(x_i,-1)]}\\ &= 1 + e^{-y_i\mathbf{w}\cdot\boldsymbol{\Psi}(x_i)},\end{aligned}$$

where for all $x \in \mathcal{X}$, $\boldsymbol{\Psi}(x) = \boldsymbol{\Phi}(x,+1) - \boldsymbol{\Phi}(x,-1)$. This leads to the following optimization problem, which defines *L_1-regularized logistic regression*:

$$\min_{\mathbf{w}\in\mathbb{R}^N} \lambda\|\mathbf{w}\|_1 + \frac{1}{m}\sum_{i=1}^{m} \log\left[1 + e^{-y_i\mathbf{w}\cdot\boldsymbol{\Psi}(x_i)}\right]. \tag{13.16}$$

As discussed in the general case, this is a convex optimization problem which admits a variety of different solutions. A common solution is SGD, another one is coordinate descent. When coordinate descent is used, then the algorithm coincides with the alternative to AdaBoost where the logistic loss is used instead of the exponential loss ($\phi(-u) = \log_2(1 + e^{-u}) \geq 1_{u\leq 0}$).

13.7.2 Logistic model

In the binary case, the conditional probability defined by the weight vector $\mathbf{w}$ can be expressed as follows:

$$\mathsf{p}_{\mathbf{w}}[y = +1 \mid x] = \frac{e^{\mathbf{w}\cdot\boldsymbol{\Phi}(x,+1)}}{Z(x)}, \tag{13.17}$$

with $Z(x) = e^{\mathbf{w}\cdot\boldsymbol{\Phi}(x,+1)} + e^{\mathbf{w}\cdot\boldsymbol{\Phi}(x,-1)}$. Thus, prediction is based on a linear decision rule defined by the sign of log-odds ratio:

$$\log\frac{\mathsf{p}_{\mathbf{w}}[y = +1 \mid x]}{\mathsf{p}_{\mathbf{w}}[y = -1 \mid x]} = \mathbf{w}\cdot\big(\boldsymbol{\Phi}(x,+1) - \boldsymbol{\Phi}(x,-1)\big) = \mathbf{w}\cdot\boldsymbol{\Psi}(x).$$

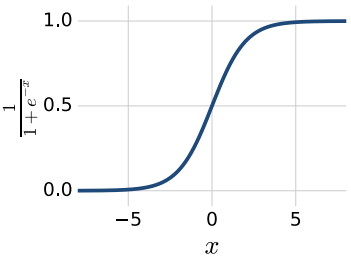


Figure 13.1
Plot of the logistic function f_{logistic}.

This is why logistic regression is also known as a log-*linear model*. Observe also that the conditional probability admits the following *logistic form*:

$$\mathsf{p}_{\mathbf{w}}[y = +1 \mid x] = \frac{1}{1 + e^{-\mathbf{w}\cdot[\boldsymbol{\Phi}(x,+1)-\boldsymbol{\Phi}(x,-1)]}} = \frac{1}{1 + e^{-\mathbf{w}\cdot\boldsymbol{\Psi}(x)}} = f_{\text{logistic}}\big(\mathbf{w}\cdot\boldsymbol{\Psi}(x)\big),$$

where f_{logistic} is the function defined over $\mathbb{R}$ by $f_{\text{logistic}} \colon x \mapsto \frac{1}{1+e^{-x}}$. Figure 13.1 shows the plot of this function. The logistic function maps the images of the linear function $x \mapsto \boldsymbol{\Psi}(x)$ to the interval $[0, 1]$, which makes them interpretable as probabilities.

L_1-regularized logistic regression benefits from the strong learning guarantees already presented for conditional maxent models, in the special case of two classes ($c = 2$). The learning guarantees for L_2-regularized logistic regression will be similarly special cases of those presented in the next section.

13.8 L_2-regularization

A common variant of conditional Maxent models is one where the dimension N is finite and where the regularization is based on the norm-2 squared of the weight vector $\mathbf{w}$. The optimization problem is thus given by

$$\min_{\mathbf{w}\in\mathbb{R}^N} \lambda\|\mathbf{w}\|_2^2 - \frac{1}{m}\sum_{i=1}^{m} \log \mathsf{p}_{\mathbf{w}}[y_i|x_i],$$

where for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$\mathsf{p}_{\mathbf{w}}[y|x] = \frac{\exp(\mathbf{w}\cdot\boldsymbol{\Phi}(x,y))}{Z(x)} \quad \text{and} \quad Z(x) = \sum_{y\in\mathcal{Y}} \exp(\mathbf{w}\cdot\boldsymbol{\Phi}(x,y)). \tag{13.18}$$

As for the norm-1 regularization, there are many optimization solutions available for this problem, including special-purpose algorithms, general first-order and second-

order solutions, and special-purpose distributed solutions. Here, the objective is additionally differentiable. A common optimization method is simply stochastic gradient descent (SGD).

In contrast to norm-1-regularized conditional Maxent models, which lead to sparser weight vectors, norm-2 conditional Maxent models lead to non-sparse solutions, which may be preferable and lead to more accurate solutions in some applications such as natural language processing. The following margin-based guarantee holds for norm-2 regularized conditional Maxent, assuming that the norm-2 of the feature vector is bounded.

Theorem 13.4 *For any $\delta > 0$, with probability at least $1 - \delta$ over the draw of an i.i.d. sample S of size m, the following holds for all $\rho > 0$ and $f \in \mathcal{F} = \{(x, y) \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x, y) \colon \|\mathbf{w}\|_2 \leq 1\}$:*

$$R(f) \leq \frac{1}{m} \sum_{i=1}^{m} \log_{u_0} \left(\sum_{y \in \mathcal{Y}} e^{\frac{f(x_i, y) - f(x_i, y_i)}{\rho}} \right) + \frac{4 r_2 c^2}{\rho \sqrt{m}} + \sqrt{\frac{\log \log_2 \frac{4 r_2}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}},$$

where $u_0 = \log(1 + 1/e)$ and $r_2 = \sup_{(x,y)} \|\mathbf{\Phi}(x, y)\|_2$.

Proof: The proof is similar to that of Theorem 13.3, modulo the observation that here $|\mathbf{w} \cdot \mathbf{\Phi}(x, y)| \leq \|\mathbf{w}\|_2 \|\mathbf{\Phi}(x, y)\|_2 \leq r_2$ and modulo the upper bound on $\mathfrak{R}_m(\Pi_1(\mathcal{F}))$. For any sample $S = (x_1, \ldots, x_m)$ of size m, the empirical Rademacher complexity of $\Pi_1(\mathcal{F})$ can be bounded as follows:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(\Pi_1(\mathcal{F})) &= \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\substack{\|\mathbf{w}\|_2 \leq 1 \\ y \in \mathcal{Y}}} \sum_{i=1}^{m} \sigma_i \mathbf{w} \cdot \mathbf{\Phi}(x_i, y) \right] \\
&= \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\substack{\|\mathbf{w}\|_2 \leq 1 \\ y \in \mathcal{Y}}} \mathbf{w} \cdot \sum_{i=1}^{m} \sigma_i \mathbf{\Phi}(x_i, y) \right] \\
&= \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{y \in \mathcal{Y}} \left\| \sum_{i=1}^{m} \sigma_i \mathbf{\Phi}(x_i, y) \right\|_2 \right] \\
&\leq \frac{1}{m} \sum_{y \in \mathcal{Y}} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{\Phi}(x_i, y) \right\|_2 \right] \\
&\leq \frac{1}{m} \sum_{y \in \mathcal{Y}} \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{\Phi}(x_i, y) \right\|_2^2 \right]} \\
&= \frac{1}{m} \sum_{y \in \mathcal{Y}} \sqrt{\sum_{i=1}^{m} \|\mathbf{\Phi}(x_i, y)\|_2^2} \leq \frac{r_2 c}{\sqrt{m}},
\end{aligned}$$

where the third equality holds by definition of the dual norm, and the second inequality by Jensen's inequality. □

The learning guarantee of the theorem for L_2-regularized conditional maxent models admits the advantage that the bound does not depend on the dimension. It can be very favorable for r_2 relatively small. The algorithm can then be very effective, provided that a small error can be achieved by a non-sparse weight vector.

13.9 Proof of the duality theorem

In this section, we give the full proof of Theorem 13.1.

Proof: The proof is similar to that of Theorem 12.2 and follows by application of the Fenchel duality theorem (theorem B.39) to the optimization problem (13.4) with the functions f and g defined for all $\bar{\mathsf{p}} \in (\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}$ and $\mathbf{u} \in \mathbb{R}^N$ by $f(\bar{\mathsf{p}}) = \mathbb{E}_{x\sim\widehat{\mathcal{D}}^1}\Big[\widetilde{\mathsf{D}}\big(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]\big)\Big]$, $g(\mathbf{u}) = I_{\mathcal{C}}(\mathbf{u})$ and $A\mathsf{p} = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \widehat{\mathcal{D}}^1(x)\mathsf{p}[y|x]\mathbf{\Phi}(x,y)$. A is a bounded linear map since we have $\|A\bar{\mathsf{p}}\| \leq \|\bar{\mathsf{p}}\|_1 \sup_{x\in\mathcal{X},y\in\mathcal{Y}} \|\mathbf{\Phi}(x,y)\|_\infty \leq r\|\bar{\mathsf{p}}\|_1$ for any $\bar{\mathsf{p}} \in (\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}$. Also, notice that the conjugate of A is given for all $\mathbf{w} \in \mathbb{R}^N$ and $(x,y) \in \mathcal{X}_1 \times \mathcal{Y}$ by $(A^*\mathbf{w})(x,y) = \mathbf{w} \cdot \big(\widehat{\mathcal{D}}^1(x)\mathbf{\Phi}(x,y)\big)$.

Consider $\mathbf{u}_0 \in \mathbb{R}^N$ defined by $\mathbf{u}_0 = \mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}}[\mathbf{\Phi}(x,y)] = A\bar{\mathsf{p}}_0$ with $\bar{\mathsf{p}}_0 = (\mathcal{D}(\cdot|x))_{x\in\mathcal{X}_1}$. Since $\bar{\mathsf{p}}_0$ is in $\mathrm{dom}(f) = \Delta^{\mathcal{X}_1}$, $\mathbf{u}_0$ is in $A(\mathrm{dom}(f))$. Furthermore, since λ is positive, $\mathbf{u}_0$ is contained in $\mathrm{int}(\mathcal{C})$. $g = I_{\mathcal{C}}$ equals zero over $\mathrm{int}(\mathcal{C})$ and is therefore continuous over $\mathrm{int}(\mathcal{C})$, thus g is continuous at $\mathbf{u}_0$ and we have $\mathbf{u}_0 \in A(\mathrm{dom}(f)) \cap \mathrm{cont}(g)$. Thus, the assumptions of Theorem B.39 hold.

The conjugate function of f is defined for all $\bar{\mathsf{q}} = (\mathsf{q}[\cdot|x])_{x\in\mathcal{X}_1} \in (\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}$ by

$$\begin{aligned}
f^*(\bar{\mathsf{q}}) &= \sup_{\bar{\mathsf{p}}\in(\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}} \left\{ \langle \mathsf{p}, \mathsf{q}\rangle - \sum_{x\in\mathcal{X}} \widehat{\mathcal{D}}^1(x)\, \widetilde{\mathsf{D}}(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]) \right\} \\
&= \sup_{\bar{\mathsf{p}}\in(\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}} \left\{ \sum_{x\in\mathcal{X}_1} \widehat{\mathcal{D}}^1[x] \sum_{y\in\mathcal{Y}} \frac{\mathsf{p}[y|x]\mathsf{q}[y|x]}{\widehat{\mathcal{D}}^1[x]} - \sum_{x\in\mathcal{X}_1} \widehat{\mathcal{D}}^1[x]\, \widetilde{\mathsf{D}}(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]) \right\} \\
&= \sum_{x\in\mathcal{X}_1} \widehat{\mathcal{D}}^1(x) \sup_{\bar{\mathsf{p}}\in(\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}} \left\{ \sum_{y\in\mathcal{Y}} \mathsf{p}[y|x] \left[\frac{\mathsf{q}[y|x]}{\widehat{\mathcal{D}}^1(x)}\right] - \widetilde{\mathsf{D}}(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]) \right\} \\
&= \sum_{x\in\mathcal{X}_1} \widehat{\mathcal{D}}^1(x) f_x^*\left(\frac{\mathsf{q}[y|x]}{\widehat{\mathcal{D}}^1(x)}\right),
\end{aligned}$$

where, for all $x \in \mathcal{X}_1$ and $\mathsf{p} \in \mathbb{R}^{\mathcal{X}_1}$, f_x is defined by $f_x(\bar{\mathsf{p}}) = \widetilde{\mathsf{D}}(\mathsf{p}[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x])$. By Lemma B.37, the conjugate function f_x^* is given for all $\bar{\mathsf{q}} \in (\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}$ by $f_x^*\left(\frac{\mathsf{q}[y|x]}{\widehat{\mathcal{D}}^1(x)}\right) = \log\left(\sum_{y\in\mathcal{Y}} \mathsf{p}_0[y|x] e^{\frac{\mathsf{q}[y|x]}{\widehat{\mathcal{D}}^1(x)}}\right)$. Thus, f^* is given for all $\bar{\mathsf{q}} \in (\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}$ by

$$f^*(\mathsf{q}) = \mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}^1}\left[\log\Big(\sum_{y\in\mathcal{Y}} \mathsf{p}_0[y|x] e^{\frac{\mathsf{q}[y|x]}{\widehat{\mathcal{D}}^1(x)}}\Big)\right].$$

As in the proof of Theorem 12.2, the conjugate function of $g = I_{\mathcal{C}}$ is given for all $\mathbf{w} \in \mathbb{R}^N$ by $g^*(\mathbf{w}) = \mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}}[\mathbf{w}\cdot\mathbf{\Phi}(x,y)] + \lambda\|\mathbf{w}\|_1$. In view of these identities, we can write, for all $\mathbf{w} \in \mathbb{R}^N$,

$$
\begin{aligned}
&- f^*(A^*\mathbf{w}) - g^*(-\mathbf{w}) \\
&= - \mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}^1}\left[\log\Big(\sum_{y\in\mathcal{Y}} \mathsf{p}_0[y|x] e^{\mathbf{w}\cdot\mathbf{\Phi}(x,y)}\Big)\right] + \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}[\mathbf{w}\cdot\mathbf{\Phi}(x,y)] - \lambda\|\mathbf{w}\|_1 \\
&= - \mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}^1}[\log Z(\mathbf{w},x)] + \frac{1}{m}\sum_{i=1}^m \mathbf{w}\cdot\mathbf{\Phi}(x_i,y_i) - \lambda\|\mathbf{w}\|_1 \\
&= \frac{1}{m}\sum_{i=1}^m \log\frac{e^{\mathbf{w}\cdot\mathbf{\Phi}(x_i,y_i)}}{Z(\mathbf{w},x_i)} - \lambda\|\mathbf{w}\|_1 \\
&= \frac{1}{m}\sum_{i=1}^m \log\left[\frac{\mathsf{p}_{\mathbf{w}}[y_i|x_i]}{\mathsf{p}_0[y_i|x_i]}\right] - \lambda\|\mathbf{w}\|_1 = G(\mathbf{w}),
\end{aligned}
$$

which proves that $\sup_{\mathbf{w}\in\mathbb{R}^N} G(\mathbf{w}) = \min_{\bar{\mathsf{p}}\in(\mathbb{R}^{\mathcal{Y}})^{\mathcal{X}_1}} F(\bar{\mathsf{p}})$.

The second part of the proof is similar to that of Theorem 12.2. For any $\mathbf{w} \in \mathbb{R}^N$, we can write

$$
\begin{aligned}
&G(\mathbf{w}) - \mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}^1}[\mathsf{D}(\mathsf{p}^*[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x])] + \mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}^1}[\mathsf{D}(\mathsf{p}^*[\cdot|x] \,\|\, \mathsf{p}_{\mathbf{w}}[\cdot|x])] \\
&= \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}\left[\log\frac{\mathsf{p}_{\mathbf{w}}[y|x]}{\mathsf{p}_0[y|x]}\right] - \lambda\|\mathbf{w}\|_1 - \mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}^*[\cdot|x]}}\left[\log\frac{\mathsf{p}^*[y|x]}{\mathsf{p}_0[y|x]}\right] + \mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}^*[\cdot|x]}}\left[\log\frac{\mathsf{p}^*[y|x]}{\mathsf{p}_{\mathbf{w}}[y|x]}\right] \\
&= -\lambda\|\mathbf{w}\|_1 + \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}\left[\log\frac{\mathsf{p}_{\mathbf{w}}[y|x]}{\mathsf{p}_0[y|x]}\right] - \mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}^*[\cdot|x]}}\left[\log\frac{\mathsf{p}_{\mathbf{w}}[y|x]}{\mathsf{p}_0[y|x]}\right] \\
&= -\lambda\|\mathbf{w}\|_1 + \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}[\mathbf{w}\cdot\mathbf{\Phi}(x,y) - \log Z(\mathbf{w},x)] - \mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}^*[\cdot|x]}}[\mathbf{w}\cdot\mathbf{\Phi}(x,y) - \log Z(\mathbf{w},x)] \\
&= -\lambda\|\mathbf{w}\|_1 + \mathbf{w}\cdot\Big[\mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}[\mathbf{\Phi}(x,y)] - \mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}^*[\cdot|x]}}[\mathbf{\Phi}(x,y)]\Big].
\end{aligned}
$$

As the solution of the primal optimization, $\bar{\mathsf{p}}^*$ verifies $I_{\mathcal{C}}\Big(\mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}^*[\cdot|x]}}[\mathbf{\Phi}(x,y)]\Big) = 0$, that is $\Big\|\mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}^*[\cdot|x]}}[\mathbf{\Phi}(x,y)] - \mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}}[\mathbf{\Phi}(x,y)]\Big\|_\infty \leq \lambda$. By Hölder's inequality, this implies the following inequality:

$$
-\|\mathbf{w}\|_1 + \mathbf{w}\cdot\left[\mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}[\mathbf{w}\cdot\mathbf{\Phi}(x,y)] - \mathop{\mathbb{E}}_{\substack{x\sim\widehat{\mathcal{D}}^1\\ y\sim\mathsf{p}^*[\cdot|x]}}[\mathbf{w}\cdot\mathbf{\Phi}(x,y)]\right] \leq -\|\mathbf{w}\|_1 + \|\mathbf{w}\|_1 = 0.
$$

Thus, we can write, for any $\mathbf{w} \in \mathbb{R}^N$,

$$\mathop{\mathbb{E}}_{x \sim \widehat{\mathcal{D}}_1} \left[\mathsf{D}(\mathsf{p}^*[\cdot|x] \,\|\, \mathsf{p}_{\mathbf{w}}[\cdot|x]) \right] \leq \mathop{\mathbb{E}}_{x \sim \widehat{\mathcal{D}}_1} \left[\mathsf{D}(\mathsf{p}^*[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x]) \right] - G(\mathbf{w}).$$

Now, assume that $\mathbf{w}$ verifies $|G(\mathbf{w}) - \sup_{\mathbf{w} \in \mathbb{R}^N} G(\mathbf{w})| \leq \epsilon$ for some $\epsilon > 0$. Then, $\mathbb{E}_{x \sim \widehat{\mathcal{D}}_1}[\mathsf{D}(\mathsf{p}^*[\cdot|x] \,\|\, \mathsf{p}_0[\cdot|x])] - G(\mathbf{w}) = (\sup_{\mathbf{w}} G(\mathbf{w})) - G(\mathbf{w}) \leq \epsilon$ implies the inequality $\mathbb{E}_{x \sim \widehat{\mathcal{D}}_1}[\mathsf{D}(\mathsf{p}^*[\cdot|x] \,\|\, \mathsf{p}_{\mathbf{w}}[\cdot|x])] \leq \epsilon$. This concludes the proof of the theorem. □

13.10 Chapter notes

The logistic regression model is a classical model in statistics. The term *logistic* was introduced by the Belgian mathematician Verhulst [1838, 1845]. An early reference for logistic regression is the publication of Berkson [1944] who advocated the use of the logistic function, instead of the cumulative distribution function of the standard normal distribution (*probit model*).

Conditional maximum entropy models in natural language processing were introduced by Berger et al. [1996] and were widely adopted for a variety of different tasks, including part-of-speech tagging, parsing, machine translation, and text categorization (see tutorial by Manning and Klein [2003]). Our presentation of the conditional Maxent principle, including their regularized variants, the duality theorem for conditional Maxent models (Theorem 13.1) and their theoretical justifications are based on [Cortes, Kuznetsov, Mohri, and Syed, 2015]. This chapter provided two types of justification for these models: one based on the conditional Maxent principle, another based on standard generalization bounds.

As in the case of Maxent models for density estimation, conditional Maxent models can be extended by using other Bregman divergences [Lafferty, Pietra, and Pietra, 1997] and other regularizations. Lafferty [1999] presented a general framework for incremental algorithms based on Bregman divergences that admits logistic regression as as special case, see also [Collins et al., 2002] who showed that boosting and logistic regression were special instances of a common framework based on Bregman divergences. The regularized conditional Maxent models presented in this chapter can be extended similarly using other Bregman divergences. In the binary classification case, when coordinate descent is used to solve the optimization problem of regularized conditional Maxent models, the algorithm coincides with L_1-regularized AdaBoost modulo the use of the logistic loss instead of the exponential loss.

Cortes, Kuznetsov, Mohri, and Syed [2015] presented a more general family of conditional probability models, *conditional structural Maxent models*, for which they also presented a duality theorem and gave strong learning guarantees. These Maxent models are based on feature functions selected from a union of possibly

very complex sub-families. The resulting algorithms coincide with the DeepBoost algorithms of Cortes, Mohri, and Syed [2014] in the binary classification case or the multi-class DeepBoost algorithm of Kuznetsov, Mohri, and Syed [2014] in the multi-class classification case, when the logistic function is used as a convex surrogate loss function.

13.11 Exercises

13.1 Extension to Bregman divergences.

(a) Show how conditional Maxent models can be extended by using arbitrary Bregman divergences instead of the (unnormalized) relative entropy.

(b) Prove a duality theorem similar to Theorem 13.1 for theses extensions.

(c) Derive theoretical guarantees for these extensions. What additional property is needed for the Bregman divergence so that your learning guarantees hold?

13.2 Stability analysis for L_2-regularized conditional Maxent.

(a) Give an upper bound on the stability of the L_2-regularized conditional Maxent in terms of the sample size and λ (*Hint*: use the techniques and results of Chapter 14).

(b) Use the previous question to derive a stability-based generalization guarantee for the algorithm.

13.3 Maximum conditional Maxent. An alternative measure of closeness, instead of the conditional relative entropy, is the maximum relative entropy over all $x \in \mathcal{X}_1$.

(a) Write the primal optimization problem for this maximum conditional Maxent formulation. Show that it is a convex optimization problem, and discuss its feasibility and the uniqueness of its solution.

(b) Prove a duality theorem for maximum conditional Maxent and write the equivalent dual problem.

(c) Analyze the properties of maximum conditional Maxent and give a generalization bound for the algorithm.

13.4 Conditional Maxent with other marginal distributions: discuss and analyze conditional Maxent models when using a distribution $\mathcal{Q}$ over $\mathcal{X}$ instead of $\widehat{\mathcal{D}}^1$. Prove that a duality theorem similar to Theorem 13.1 holds.

14 Algorithmic Stability

In chapters 2–5 and several subsequent chapters, we presented a variety of generalization bounds based on different measures of the complexity of the hypothesis set $\mathcal{H}$ used for learning, including the Rademacher complexity, the growth function, and the VC-dimension. These bounds ignore the specific algorithm used, that is, they hold for any algorithm using $\mathcal{H}$ as a hypothesis set.

One may ask if an analysis of the properties of a specific algorithm could lead to finer guarantees. Such an algorithm-dependent analysis could have the benefit of a more informative guarantee. On the other hand, it could be inapplicable to other algorithms using the same hypothesis set. Alternatively, as we shall see in this chapter, a more general property of the learning algorithm could be used to incorporate algorithm-specific properties while extending the applicability of the analysis to other learning algorithms with similar properties.

This chapter uses the property of *algorithmic stability* to derive *algorithm-dependent* learning guarantees. We first present a generalization bound for any algorithm that is sufficiently stable. Then, we show that the wide class of kernel-based regularization algorithms enjoys this property and derive a general upper bound on their stability coefficient. Finally, we illustrate the application of these results to the analysis of several algorithms both in the regression and classification settings, including kernel ridge regression (KRR), SVR, and SVMs.

14.1 Definitions

We start by introducing the notation and definitions relevant to our analysis of algorithmic stability. We denote by z a labeled example $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The hypotheses h we consider map $\mathcal{X}$ to a set $\mathcal{Y}'$ sometimes different from $\mathcal{Y}$. In particular, for classification, we may have $\mathcal{Y} = \{-1, +1\}$ while the hypothesis h learned takes values in $\mathbb{R}$. The loss functions L we consider are therefore defined over $\mathcal{Y}' \times \mathcal{Y}$, with $\mathcal{Y}' = \mathcal{Y}$ in most cases. For a loss function $L \colon \mathcal{Y}' \times \mathcal{Y} \to \mathbb{R}_+$, we denote the loss of

a hypothesis h at point z by $L_z(h) = L(h(x), y)$. We denote by $\mathcal{D}$ the distribution according to which samples are drawn and by $\mathcal{H}$ the hypothesis set. The empirical error or loss of $h \in \mathcal{H}$ on a sample $S = (z_1, \ldots, z_m)$ and its generalization error are defined, respectively, by

$$\widehat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} L_{z_i}(h) \quad \text{and} \quad R(h) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[L_z(h)].$$

Given an algorithm $\mathcal{A}$, we denote by h_S the hypothesis $h_S \in \mathcal{H}$ returned by $\mathcal{A}$ when trained on sample S. We will say that the loss function L is bounded by $M \geq 0$ if for all $h \in \mathcal{H}$ and $z \in \mathcal{X} \times \mathcal{Y}$, $L_z(h) \leq M$. For the results presented in this chapter, a weaker condition suffices, namely that $L_z(h_S) \leq M$ for all hypotheses h_S returned by the algorithm $\mathcal{A}$.

We are now able to define the notion of uniform stability, the algorithmic property used in the analyses of this chapter.

Definition 14.1 (Uniform stability) *Let S and S' be any two training samples that differ by a single point. Then, a learning algorithm $\mathcal{A}$ is* uniformly β-stable *if the hypotheses it returns when trained on any such samples S and S' satisfy*

$$\forall z \in \mathcal{Z}, \quad |L_z(h_S) - L_z(h_{S'})| \leq \beta.$$

The smallest such β satisfying this inequality is called the stability coefficient *of $\mathcal{A}$.*

In other words, when $\mathcal{A}$ is trained on two similar training sets, the losses incurred by the corresponding hypotheses returned by $\mathcal{A}$ should not differ by more than β. Note that a uniformly β-stable algorithm is often referred to as being *β-stable* or even just *stable* (for some unspecified β). In general, the coefficient β depends on the sample size m. We will see in section 14.2 that $\beta = o(1/\sqrt{m})$ is necessary for the convergence of the stability-based learning bounds presented in this chapter. In section 14.3, we will show that a more favorable condition holds, that is, $\beta = O(1/m)$, for a wide family of algorithms.

14.2 Stability-based generalization guarantee

In this section, we show that exponential bounds can be derived for the generalization error of stable learning algorithms. The main result is presented in theorem 14.2.

Theorem 14.2 *Assume that the loss function L is bounded by $M \geq 0$. Let $\mathcal{A}$ be a β-stable learning algorithm and let S be a sample of m points drawn i.i.d. according*

to distribution $\mathcal{D}$. Then, with probability at least $1-\delta$ over the sample S drawn, the following holds:

$$R(h_S) \leq \widehat{R}_S(h_S) + \beta + (2m\beta + M)\sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

Proof: The proof is based on the application of McDiarmid's inequality (theorem D.8) to the function Φ defined for all samples S by $\Phi(S) = R(h_S) - \widehat{R}_S(h_S)$. Let S' be another sample of size m with points drawn i.i.d. according to $\mathcal{D}$ that differs from S by exactly one point. We denote that point by z_m in S, z'_m in S', i.e.,

$$S = (z_1, \ldots, z_{m-1}, z_m) \quad \text{and} \quad S' = (z_1, \ldots, z_{m-1}, z'_m).$$

By definition of Φ, the following inequality holds:

$$|\Phi(S') - \Phi(S)| \leq |R(h_{S'}) - R(h_S)| + |\widehat{R}_{S'}(h_{S'}) - \widehat{R}_S(h_S)|. \tag{14.1}$$

We bound each of these two terms separately. By the β-stability of $\mathcal{A}$, we have

$$|R(h_S) - R(h_{S'})| = |\mathop{\mathbb{E}}_z[L_z(h_S)] - \mathop{\mathbb{E}}_z[L_z(h_{S'})]| \leq \mathop{\mathbb{E}}_z[|L_z(h_S) - L_z(h_{S'})|] \leq \beta.$$

Using the boundedness of L along with β-stability of $\mathcal{A}$, we also have

$$\begin{aligned}
|\widehat{R}_S(h_S) - \widehat{R}_{S'}(h_{S'})| &= \frac{1}{m}\left|\left(\sum_{i=1}^{m-1} L_{z_i}(h_S) - L_{z_i}(h_{S'})\right) + L_{z_m}(h_S) - L_{z'_m}(h_{S'})\right| \\
&\leq \frac{1}{m}\left[\left(\sum_{i=1}^{m-1} |L_{z_i}(h_S) - L_{z_i}(h_{S'})|\right) + |L_{z_m}(h_S) - L_{z'_m}(h_{S'})|\right] \\
&\leq \frac{m-1}{m}\beta + \frac{M}{m} \leq \beta + \frac{M}{m}.
\end{aligned}$$

Thus, in view of (14.1), Φ satisfies the condition $|\Phi(S) - \Phi(S')| \leq 2\beta + \frac{M}{m}$. By applying McDiarmid's inequality to $\Phi(S)$, we can bound the deviation of Φ from its mean as

$$\mathbb{P}\left[\Phi(S) \geq \epsilon + \mathop{\mathbb{E}}_S[\Phi(S)]\right] \leq \exp\left(\frac{-2m\epsilon^2}{(2m\beta + M)^2}\right),$$

or, equivalently, with probability $1-\delta$,

$$\Phi(S) < \epsilon + \mathop{\mathbb{E}}_S[\Phi(S)], \tag{14.2}$$

where $\delta = \exp\left(\frac{-2m\epsilon^2}{(2m\beta+M)^2}\right)$. If we solve for ϵ in this expression for δ, plug into (14.2) and rearrange terms, then, with probability $1-\delta$, we have

$$\Phi(S) \leq \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\Phi(S)] + (2m\beta + M)\sqrt{\frac{\log \frac{1}{\delta}}{2m}}. \tag{14.3}$$

We now bound the expectation term, first noting that by linearity of expectation $\mathbb{E}_S[\Phi(S)] = \mathbb{E}_S[R(h_S)] - \mathbb{E}_S[\widehat{R}_S(h_S)]$. By definition of the generalization error,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[R(h_S)] = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\Big[\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[L_z(h_S)]\Big] = \mathop{\mathbb{E}}_{S,z\sim\mathcal{D}^{m+1}}[L_z(h_S)]. \tag{14.4}$$

By the linearity of expectation,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\widehat{R}_S(h_S)] = \frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{z_i}(h_S)] = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{z_1}(h_S)], \tag{14.5}$$

where the second equality follows from the fact that the z_i are drawn i.i.d. and thus the expectations $\mathbb{E}_{S\sim\mathcal{D}^m}[L_{z_i}(h_S)]$, $i \in [m]$, are all equal. The last expression in (14.5) is the expected loss of a hypothesis on one of its training points. We can rewrite it as $\mathbb{E}_{S\sim D^m}[L_{z_1}(h_S)] = \mathbb{E}_{S,z\sim\mathcal{D}^{m+1}}[L_z(h_{S'})]$, where S' is a sample of m points containing z extracted from the $m+1$ points formed by S and z. Thus, in view of (14.4) and by the β-stability of $\mathcal{A}$, it follows that

$$\begin{aligned}
\Big|\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\Phi(S)]\Big| &= \Big|\mathop{\mathbb{E}}_{S,z\sim\mathcal{D}^{m+1}}[L_z(h_S)] - \mathop{\mathbb{E}}_{S,z\sim\mathcal{D}^{m+1}}[L_z(h_{S'})]\Big| \\
&\leq \mathop{\mathbb{E}}_{S,z\sim\mathcal{D}^{m+1}}\big[|L_z(h_S) - L_z(h_{S'})|\big] \\
&\leq \mathop{\mathbb{E}}_{S,z\sim\mathcal{D}^{m+1}}[\beta] = \beta.
\end{aligned}$$

We can thus replace $\mathbb{E}_S[\Phi(S)]$ by β in (14.3), which completes the proof. □

The bound of the theorem converges for $(m\beta)/\sqrt{m} = o(1)$, that is $\beta = o(1/\sqrt{m})$. In particular, when the stability coefficient β is in $O(1/m)$, the theorem guarantees that $R(h_S) - \widehat{R}_S(h_S) = O(1/\sqrt{m})$ with high probability. In the next section, we show that kernel-based regularization algorithms precisely admit this property under some general assumptions.

14.3 Stability of kernel-based regularization algorithms

Let K be a positive definite symmetric kernel, $\mathbb{H}$ the reproducing kernel Hilbert space associated to K, and $\|\cdot\|_K$ the norm induced by K in $\mathbb{H}$. A kernel-based regularization algorithm is defined by the minimization over $\mathbb{H}$ of an objective function F_S based on a training sample $S = (z_1, \ldots, z_m)$ and defined for all $h \in \mathbb{H}$ by:

$$F_S(h) = \widehat{R}_S(h) + \lambda\|h\|_K^2. \tag{14.6}$$

In this equation, $\widehat{R}_S(h) = \frac{1}{m}\sum_{i=1}^{m} L_{z_i}(h)$ is the empirical error of hypothesis h with respect to a loss function L and $\lambda \geq 0$ a trade-off parameter balancing the emphasis on the empirical error versus the regularization term $\|h\|_K^2$. The hypothesis set $\mathcal{H}$

is the subset of $\mathbb{H}$ formed by the hypotheses possibly returned by the algorithm. Algorithms such as KRR, SVR and SVMs all fall under this general model.

We first introduce some definitions and tools needed for a general proof of an upper bound on the stability coefficient of kernel-based regularization algorithms. Our analysis will assume that the loss function L is convex and that it further verifies the following Lipschitz-like smoothness condition.

Definition 14.3 (σ-admissibility) *A loss function L is* σ-admissible *with respect to the hypothesis class $\mathcal{H}$ if there exists $\sigma \in \mathbb{R}_+$ such that for any two hypotheses $h, h' \in \mathcal{H}$ and for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$,*

$$|L(h'(x), y) - L(h(x), y)| \leq \sigma |h'(x) - h(x)|. \tag{14.7}$$

This assumption holds for the quadratic loss and most other loss functions where the hypothesis set and the set of output labels are bounded by some $M \in \mathbb{R}_+$: $\forall h \in \mathcal{H}, \forall x \in \mathcal{X}, |h(x)| \leq M$ and $\forall y \in \mathcal{Y}, |y| \leq M$.

We will use the notion of *Bregman divergence*, B_F which can be defined for any convex and differentiable function $F \colon \mathbb{H} \to \mathbb{R}$ as follows: for all $f, g \in \mathbb{H}$,

$$\mathsf{B}_F(f \| g) = F(f) - F(g) - \langle f - g, \nabla F(g) \rangle .$$

Section E.4 presents the properties of the Bregman divergence in more detail and also contains figure E.2 which illustrates the geometric interpretation of the Bregman divergence. We generalize the definition of Bregman divergence to cover the case of convex but non-differentiable loss functions F by using the notion of *subgradient*. For a convex function $F \colon \mathbb{H} \to \mathbb{R}$, we denote by $\partial F(h)$ the *subdifferential* of F at h, which is defined as follows:

$$\partial F(h) = \{g \in \mathbb{H} \colon \forall h' \in \mathbb{H}, F(h') - F(h) \geq \langle h' - h, g \rangle\}.$$

Thus, $\partial F(h)$ is the set of vectors g defining a hyperplane supporting function F at point h (see figure 14.1). Elements of the subdifferential are called subgradients (see section B.4.1 for more discussion). Note, the subgradient found in $\partial F(h)$ coincides with $\nabla F(h)$ when F is differentiable at h, i.e. $\partial F(h) = \{\nabla F(h)\}$. Furthermore, at a point h where F is minimal, 0 is an element of $\partial F(h)$. The subgradient is additive, that is, for two convex function F_1 and F_2, $\partial(F_1 + F_2)(h) = \{g_1 + g_2 \colon g_1 \in \partial F_1(h), g_2 \in \partial F_2(h)\}$. For any $h \in \mathbb{H}$, we fix $\delta F(h)$ to be an (arbitrary) element of $\partial F(h)$. For any such choice of δF, we can define the *generalized Bregman divergence* associated to F by:

$$\forall h', h \in \mathbb{H}, \mathsf{B}_F(h' \,\|\, h) = F(h') - F(h) - \langle h' - h, \delta F(h) \rangle . \tag{14.8}$$

Note that by definition of the subgradient, $\mathsf{B}_F(h' \,\|\, h) \geq 0$ for all $h', h \in \mathbb{H}$.

Starting from (14.6), we can now define the generalized Bregman divergence of F_S. Let N denote the convex function $h \to \|h\|_K^2$. Since N is differentiable,

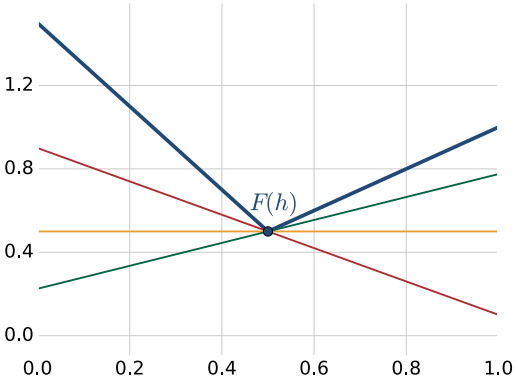


Figure 14.1
Illustration of the notion of subgradient: supporting hyperplanes (shown in red, orange and green) for the function F (shown in blue) at point h are defined by elements of the subdifferential $\partial F(h)$.

$\delta N(h) = \nabla N(h)$ for all $h \in \mathbb{H}$, and thus δN (as well as B_N) is uniquely defined. To make the definition of the Bregman divergences for F_S and $\widehat{R}_S$ compatible so that $\mathsf{B}_{F_S} = \mathsf{B}_{\widehat{R}_S} + \lambda \mathsf{B}_N$, we define $\delta \widehat{R}_S$ in terms of δF_S by: $\delta \widehat{R}_S(h) = \delta F_S(h) - \lambda \nabla N(h)$ for all $h \in \mathbb{H}$. Furthermore, we choose $\delta F_S(h)$ to be 0 for any point h where F_S is minimal and let $\delta F_S(h)$ be an arbitrary element of $\partial F_S(h)$ for all other $h \in \mathbb{H}$. We proceed in a similar way to define the Bregman divergences for $F_{S'}$ and $\widehat{R}_{S'}$ so that $\mathsf{B}_{F_{S'}} = \mathsf{B}_{\widehat{R}_{S'}} + \lambda \mathsf{B}_N$.

We will use the notion of generalized Bregman divergence for the proof of the following general upper bound on the stability coefficient of kernel-based regularization algorithms.

Proposition 14.4 *Let K be a positive definite symmetric kernel such that for all $x \in \mathcal{X}$, $K(x,x) \leq r^2$ for some $r \in \mathbb{R}_+$ and let L be a convex and σ-admissible loss function. Then, the kernel-based regularization algorithm defined by the minimization* (14.6) *is β-stable with the following upper bound on β:*

$$\beta \leq \frac{\sigma^2 r^2}{m\lambda}.$$

Proof: Let h be a minimizer of F_S and h' a minimizer of $F_{S'}$, where samples S and S' differ exactly by one point, z_m in S and z'_m in S'. Since the generalized Bregman divergence is non-negative and since $\mathsf{B}_{F_S} = \mathsf{B}_{\widehat{R}_S} + \lambda \mathsf{B}_N$ and $\mathsf{B}_{F_{S'}} = \mathsf{B}_{\widehat{R}_{S'}} + \lambda \mathsf{B}_N$, we can write

$$\mathsf{B}_{F_S}(h' \| h) + \mathsf{B}_{F_{S'}}(h \| h') \geq \lambda \big(\mathsf{B}_N(h' \| h) + \mathsf{B}_N(h \| h') \big).$$

Observe that $\mathsf{B}_N(h'\|h) + \mathsf{B}_N(h\|h') = -\langle h'-h, 2h\rangle - \langle h-h', 2h'\rangle = 2\|h'-h\|_K^2$. Let Δh denote $h'-h$, then we can write

$$\begin{aligned}
&2\lambda||\Delta h\|_K^2 \\
&\quad \leq \mathsf{B}_{F_S}(h' \,\|\, h) + \mathsf{B}_{F_{S'}}(h \,\|\, h') \\
&\quad = F_S(h') - F_S(h) - \langle h'-h, \delta F_S(h)\rangle + F_{S'}(h) - F_{S'}(h') - \langle h-h', \delta F_{S'}(h')\rangle \\
&\quad = F_S(h') - F_S(h) + F_{S'}(h) - F_{S'}(h') \\
&\quad = \widehat{R}_S(h') - \widehat{R}_S(h) + \widehat{R}_{S'}(h) - \widehat{R}_{S'}(h').
\end{aligned}$$

The second equality follows from the definition of h' and h as minimizers and our choice of the subgradients for minimal points which together imply $\delta F_{S'}(h') = 0$ and $\delta F_S(h) = 0$. The last equality follows from the definitions of F_S and $F_{S'}$. Next, we express the resulting inequality in terms of the loss function L and use the fact that S and S' differ by only one point along with the σ-admissibility of L to get

$$\begin{aligned}
2\lambda\|\Delta h\|_K^2 &\leq \frac{1}{m}[L_{z_m}(h') - L_{z_m}(h) + L_{z'_m}(h) - L_{z'_m}(h')] \\
&\leq \frac{\sigma}{m}[|\Delta h(x_m)| + |\Delta h(x'_m)|]. \qquad (14.9)
\end{aligned}$$

By the reproducing kernel property and the Cauchy-Schwarz inequality, for all $x \in \mathcal{X}$,

$$\Delta h(x) = \langle \Delta h, K(x,\cdot)\rangle \leq \|\Delta h\|_K \|K(x,\cdot)\|_K = \sqrt{K(x,x)}\|\Delta h\|_K \leq r\|\Delta h\|_K.$$

In view of (14.9), this implies $\|\Delta h\|_K \leq \frac{\sigma r}{\lambda m}$. By the σ-admissibility of L and the reproducing property, the following holds:

$$\forall z \in \mathcal{X}\times\mathcal{Y}, |L_z(h') - L_z(h)| \leq \sigma|\Delta h(x)| \leq r\sigma\|\Delta h\|_K,$$

which gives

$$\forall z \in \mathcal{X}\times\mathcal{Y}, |L_z(h') - L_z(h)| \leq \frac{\sigma^2 r^2}{m\lambda},$$

and concludes the proof. □

Thus, under the assumptions of the proposition, for a fixed λ, the stability coefficient of kernel-based regularization algorithms is in $O(1/m)$.

14.3.1 Application to regression algorithms: SVR and KRR

Here, we analyze more specifically two widely used regression algorithms, Support Vector Regression (SVR) and Kernel Ridge Regression (KRR), which are both special instances of the family of kernel-based regularization algorithms.

SVR is based on the ϵ-insensitive loss L_ϵ defined for all $(y, y') \in \mathcal{Y} \times \mathcal{Y}$ by:

$$L_\epsilon(y', y) = \begin{cases} 0 & \text{if } |y' - y| \leq \epsilon; \\ |y' - y| - \epsilon & \text{otherwise.} \end{cases} \tag{14.10}$$

We now present a stability-based bound for SVR assuming that L_ϵ is bounded for the hypotheses returned by SVR (which, as we shall later see in lemma 14.7, is indeed the case when the label set $\mathcal{Y}$ is bounded).

Corollary 14.5 (Stability-based learning bound for SVR) *Assume that $K(x, x) \leq r^2$ for all $x \in \mathcal{X}$ for some $r \geq 0$ and that L_ϵ is bounded by $M \geq 0$. Let h_S denote the hypothesis returned by SVR when trained on an i.i.d. sample S of size m. Then, for any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$:*

$$R(h_S) \leq \widehat{R}_S(h_S) + \frac{r^2}{m\lambda} + \Big(\frac{2r^2}{\lambda} + M\Big)\sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

Proof: We first show that $L_\epsilon(\cdot) = L_\epsilon(\cdot, y)$ is 1-Lipschitz for any $y \in \mathcal{Y}$. For any $y', y'' \in \mathcal{Y}$, we must consider four cases. First, if $|y' - y| \leq \epsilon$ and $|y'' - y| \leq \epsilon$, then $|L_\epsilon(y'') - L_\epsilon(y')| = 0$. Second, if $|y' - y| > \epsilon$ and $|y'' - y| > \epsilon$, then $|L_\epsilon(y'') - L_\epsilon(y')| = ||y'' - y| - |y' - y|| \leq |y'' - y'|$, by the triangle inequality. Third, if $|y' - y| \leq \epsilon$ and $|y'' - y| > \epsilon$, then $|L_\epsilon(y'') - L_\epsilon(y')| = ||y'' - y| - \epsilon| = |y'' - y| - \epsilon \leq |y'' - y| - |y' - y| \leq |y'' - y'|$. Fourth, if $|y'' - y| \leq \epsilon$ and $|y' - y| > \epsilon$, by symmetry the same inequality is obtained as in the previous case.

Thus, in all cases, $|L_\epsilon(y'', y) - L_\epsilon(y', y)| \leq |y'' - y'|$. This implies in particular that L_ϵ is σ-admissible with $\sigma = 1$ for any hypothesis set $\mathcal{H}$. By proposition 14.4, under the assumptions made, SVR is β-stable with $\beta \leq \frac{r^2}{m\lambda}$. Plugging this expression into the bound of theorem 14.2 yields the result. □

We next present a stability-based bound for KRR, which is based on the square loss L_2 defined for all $y', y \in \mathcal{Y}$ by:

$$L_2(y', y) = (y' - y)^2. \tag{14.11}$$

As in the SVR setting, we assume in our analysis that L_2 is bounded for the hypotheses returned by KRR (which, as we shall later see again in lemma 14.7, is indeed the case when the label set $\mathcal{Y}$ is bounded).

Corollary 14.6 (Stability-based learning bound for KRR) *Assume that $K(x, x) \leq r^2$ for all $x \in \mathcal{X}$ for some $r \geq 0$ and that L_2 is bounded by $M \geq 0$. Let h_S denote the hypothesis returned by KRR when trained on an i.i.d. sample S of size m. Then, for any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$:*

$$R(h_S) \leq \widehat{R}_S(h_S) + \frac{4Mr^2}{\lambda m} + \Big(\frac{8Mr^2}{\lambda} + M\Big)\sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

Proof: For any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $h, h' \in \mathcal{H}$,

$$\begin{aligned}|L_2(h'(x), y) - L_2(h(x), y)| &= \left|(h'(x) - y)^2 - (h(x) - y)^2\right| \\ &= \left|\big[h'(x) - h(x)\big]\big[(h'(x) - y) + (h(x) - y)\big]\right| \\ &\leq (|h'(x) - y| + |h(x) - y|)|h(x) - h'(x)| \\ &\leq 2\sqrt{M}|h(x) - h'(x)|,\end{aligned}$$

where we used the M-boundedness of the loss. Thus, L_2 is σ-admissible with $\sigma = 2\sqrt{M}$. Therefore, by proposition 14.4, KRR is β-stable with $\beta \leq \frac{4r^2 M}{m\lambda}$. Plugging this expression into the bound of theorem 14.2 yields the result. □

The previous two corollaries assumed bounded loss functions. We now present a lemma that implies in particular that the loss functions used by SVR and KRR are bounded when the label set is bounded.

Lemma 14.7 *Assume that $K(x, x) \leq r^2$ for all $x \in \mathcal{X}$ for some $r \geq 0$ and that for all $y \in \mathcal{Y}$, $L(0, y) \leq B$ for some $B \geq 0$. Then, the hypothesis h_S returned by a kernel-based regularization algorithm trained on a sample S is bounded as follows:*

$$\forall x \in \mathcal{X}, |h_S(x)| \leq r\sqrt{B/\lambda}.$$

Proof: By the reproducing kernel property and the Cauchy-Schwarz inequality, we can write

$$\forall x \in \mathcal{X}, |h_S(x)| = \langle h_S, K(x, \cdot)\rangle \leq \|h_S\|_K \sqrt{K(x, x)} \leq r\|h_S\|_K. \tag{14.12}$$

The minimization (14.6) is over $\mathbb{H}$, which includes 0. Thus, by definition of F_S and h_S, the following inequality holds:

$$F_S(h_S) \leq F_S(0) = \frac{1}{m}\sum_{i=1}^{m} L(0, y_i) \leq B.$$

Since the loss L is non-negative, we have $\lambda\|h_S\|_K^2 \leq F_S(h_S)$ and thus $\lambda\|h_S\|_K^2 \leq B$. Combining this inequality with (14.12) yields the result. □

14.3.2 Application to classification algorithms: SVMs

This section presents a generalization bound for SVMs, when using the standard hinge loss defined for all $y \in \mathcal{Y} = \{-1, +1\}$ and $y' \in \mathbb{R}$ by

$$L_{\text{hinge}}(y', y) = \begin{cases} 0 & \text{if } 1 - yy' \leq 0; \\ 1 - yy' & \text{otherwise.} \end{cases} \tag{14.13}$$

Corollary 14.8 (Stability-based learning bound for SVMs) *Assume that $K(x, x) \leq r^2$ for all $x \in \mathcal{X}$ for some $r \geq 0$. Let h_S denote the hypothesis returned by SVMs when*

trained on an i.i.d. sample S of size m. Then, for any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$:

$$R(h_S) \leq \widehat{R}_S(h_S) + \frac{r^2}{m\lambda} + \Big(\frac{2r^2}{\lambda} + \frac{r}{\sqrt{\lambda}} + 1\Big)\sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

Proof: It is straightforward to verify that $L_{\text{hinge}}(\cdot, y)$ is 1-Lipschitz for any $y \in \mathcal{Y}$ and therefore that it is σ-admissible with $\sigma = 1$. Therefore, by proposition 14.4, SVMs is β-stable with $\beta \leq \frac{r^2}{m\lambda}$. Since $|L_{\text{hinge}}(0, y)| \leq 1$ for any $y \in \mathcal{Y}$, by lemma 14.7, $\forall x \in \mathcal{X}, |h_S(x)| \leq r/\sqrt{\lambda}$. Thus, for any sample S and any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, the loss is bounded as follows: $L_{\text{hinge}}(h_S(x), y) \leq r/\sqrt{\lambda} + 1$. Plugging this value of M and the one found for β into the bound of theorem 14.2 yields the result. □

Since the hinge loss upper bounds the binary loss, the bound of the corollary 14.8 also applies to the generalization error of h_S measured in terms of the standard binary loss used in classification.

14.3.3 Discussion

Note that the learning bounds presented for kernel-based regularization algorithms are of the form $R(h_S) - \widehat{R}_S(h_S) \leq O\big(\frac{1}{\lambda\sqrt{m}}\big)$. Thus, these bounds are informative only when $\lambda \gg 1/\sqrt{m}$. The regularization parameter λ is a function of the sample size m: for larger values of m, it is expected to be smaller, decreasing the emphasis on regularization. The magnitude of λ affects the norm of the linear hypotheses used for prediction, with a larger value of λ implying a smaller hypothesis norm. In this sense, λ is a measure of the complexity of the hypothesis set and the condition required for λ can be interpreted as stating that a less complex hypothesis set guarantees better generalization.

Note also that our analysis of stability in this chapter assumed a fixed λ: the regularization parameter is assumed to be invariant to the change of one point of the training sample. While this is a mild assumption, it may not hold in general.

14.4 Chapter notes

The notion of algorithmic stability was first used by Devroye, Rogers and Wagner [Rogers and Wagner, 1978, Devroye and Wagner, 1979a,b] for the k-nearest neighbor algorithm and other k-local rules. Kearns and Ron [1999] later gave a formal definition of stability and used it to provide an analysis of the leave-one-out error. Much of the material presented in this chapter is based on Bousquet and Elisseeff [2002]. Our proof of proposition 14.4 is novel and generalizes the results of Bousquet and Elisseeff [2002] to the case of non-differentiable convex losses. Moreover, stability-based generalization bounds have been extended to ranking algorithms [Agarwal and Niyogi, 2005, Cortes et al., 2007b], as well as to the non-i.i.d.

scenario of stationary Φ- and β-mixing processes [Mohri and Rostamizadeh, 2010], and to the transductive setting [Cortes et al., 2008a]. Additionally, exercise 14.5 is based on Cortes et al. [2010b], which introduces and analyzes stability with respect to the choice of the kernel function or kernel matrix.

Note that while, as shown in this chapter, uniform stability is sufficient for deriving generalization bounds, it is not a necessary condition. Some algorithms may generalize well in the supervised learning scenario but may not be uniformly stable, for example, the Lasso algorithm [Xu et al., 2008]. Shalev-Shwartz et al. [2009] have used the notion of stability to provide necessary and sufficient conditions for a technical condition of learnability related to PAC-learning, even in general scenarios where learning is possible only by using non-ERM rules.

14.5 Exercises

14.1 Tighter stability bounds

(a) Assuming the conditions of theorem 14.2 hold, can one hope to guarantee a generalization with slack better than $O(1/\sqrt{m})$ even if the algorithm is very stable, i.e. $\beta \to 0$?

(b) Can you show an $O(1/m)$ generalization guarantee if L is bounded by $C/\sqrt{m}$ (a very strong condition)? If so, how stable does the learning algorithm need to be?

14.2 Quadratic hinge loss stability. Let L denote the quadratic hinge loss function defined for all $y \in \{+1, -1\}$ and $y' \in \mathbb{R}$ by

$$L(y', y) = \begin{cases} 0 & \text{if } 1 - y'y \leq 0; \\ (1 - y'y)^2 & \text{otherwise.} \end{cases}$$

Assume that $L(h(x), y)$ is bounded by M, $1 \leq M < \infty$, for all $h \in \mathcal{H}$, $x \in \mathcal{X}$, and $y \in \{+1, -1\}$, which also implies a bound on $|h(x)|$ for all $h \in \mathcal{H}$ and $x \in \mathcal{X}$. Derive a stability-based generalization bound for SVMs with the quadratic hinge loss.

14.3 Stability of linear regression.

(a) How does the stability bound in corollary 14.6 for ridge regression (i.e. kernel ridge regression with a linear kernel) behave as $\lambda \to 0$?

(b) Can you show a stability bound for linear regression (i.e. ridge regression with $\lambda = 0$)? If not, show a counter-example.

14.4 Kernel stability. Suppose an approximation of the kernel matrix $\mathbf{K}$, denoted $\mathbf{K}'$, is used to train the hypothesis h' (and let h denote the non-approximate hypothesis). At test time, no approximation is made, so if we let $\mathbf{k}_x = \big[K(x, x_1), \ldots, K(x, x_m)\big]^\top$ we can write $h(x) = \boldsymbol{\alpha}^\top \mathbf{k}_x$ and $h'(x) = \boldsymbol{\alpha}'^\top \mathbf{k}_x$. Show that if $\forall x, x' \in \mathcal{X}, K(x, x') \leq r$ then

$$|h'(x) - h(x)| \leq \frac{rmM}{\lambda^2} \|\mathbf{K}' - \mathbf{K}\|_2 \,.$$

(*Hint*: Use exercise 10.3)

14.5 Stability of relative-entropy regularization.

(a) Consider an algorithm that selects a distribution g over a hypothesis class which is parameterized by $\theta \in \Theta$. Given a point $z = (x, y)$ the expected loss is defined as

$$H(g, z) = \int_\Theta L(h_\theta(x), y) g(\theta) \, d\theta \,,$$

with respect to a base loss function L. Assuming the loss function L is bounded by M, show that the expected loss H is M-admissible, i.e. show $|H(g, z) - H(g', z)| \leq M \int_\Theta |g(\theta) - g'(\theta)| \, d\theta$.

(b) Consider an algorithm that minimizes the *entropy regularized* objective over the choice of distribution g:

$$F_S(g) = \underbrace{\frac{1}{m} \sum_{i=1}^{m} H(g, z_i)}_{\widehat{R}_S(g)} + \lambda K(g, f_0) \,.$$

Here, K is the Kullback-Leibler divergence (or relative entropy) between two distributions,

$$K(g, f_0) = \int_\Theta g(\theta) \log \frac{g(\theta)}{f_0(\theta)} \, d\theta \,, \tag{14.14}$$

and f_0 is some fixed distribution. Show that such an algorithm is stable by performing the following steps:

i. First use the fact $\frac{1}{2}(\int_\Theta |g(\theta) - g'(\theta)| \, d\theta)^2 \leq K(g, g')$ (Pinsker's inequality), to show

$$\Big(\int_\Theta |g_S(\theta) - g_{S'}(\theta)| \, d\theta \Big)^2 \leq B_{K(\cdot, f_0)}(g \| g') + B_{K(\cdot, f_0)}(g' \| g) \,.$$

ii. Next, let g be the minimizer of F_S and g' the minimizer of $F_{S'}$, where S and S' differ only at the index m. Show that

$$\begin{aligned} B_{K(.,f_0)}(g\|g') + B_{K(.,f_0)}(g'\|g) \\ \leq \frac{1}{m\lambda}\big|H(g', z_m) - H(g, z_m) + H(g, z'_m) - H(g', z'_m)\big| \\ \leq \frac{2M}{m\lambda}\int_\Theta |g(\theta) - g'(\theta)|\, d\theta\,. \end{aligned}$$

iii. Finally, combine the results above to show that the entropy regularized algorithm is $\frac{2M^2}{m\lambda}$-stable.

15 Dimensionality Reduction

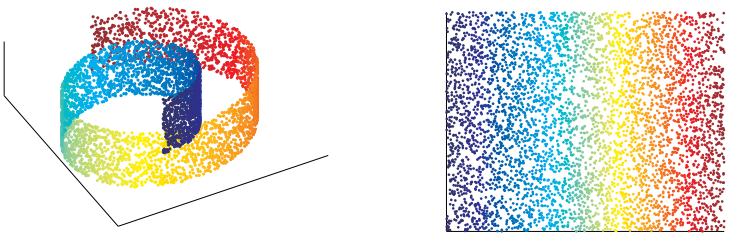
In settings where the data has a large number of features, it is often desirable to reduce its dimension, or to find a lower-dimensional representation preserving some of its properties. The key arguments for dimensionality reduction (or manifold learning) techniques are:

- *Computational*: to compress the initial data as a preprocessing step to speed up subsequent operations on the data.
- *Visualization*: to visualize the data for exploratory analysis by mapping the input data into two- or three-dimensional spaces.
- *Feature extraction*: to hopefully generate a smaller and more effective or useful set of features.

The benefits of dimensionality reduction are often illustrated via simulated data, such as the Swiss roll dataset. In this example, the input data, depicted in figure 15.1a, is three-dimensional, but it lies on a two-dimensional manifold that is "unfolded" in two-dimensional space as shown in figure 15.1b. It is important to note, however, that exact low-dimensional manifolds are rarely encountered in practice. Hence, this idealized example is more useful to illustrate the concept of dimensionality reduction than to verify the effectiveness of dimensionality reduction algorithms.

Dimensionality reduction can be formalized as follows. Consider a sample $S = (x_1, \ldots, x_m)$, a feature mapping $\mathbf{\Phi} \colon \mathcal{X} \to \mathbb{R}^N$ and the data matrix $\mathbf{X} \in \mathbb{R}^{N \times m}$ defined as $(\mathbf{\Phi}(x_1), \ldots, \mathbf{\Phi}(x_m))$. The ith data point is represented by $\mathbf{x}_i = \mathbf{\Phi}(x_i)$, or the ith column of $\mathbf{X}$, which is an N-dimensional vector. Dimensionality reduction techniques broadly aim to find, for $k \ll N$, a k-dimensional representation of the data, $\mathbf{Y} \in \mathbb{R}^{k \times m}$, that is in some way faithful to the original representation $\mathbf{X}$.

In this chapter we will discuss various techniques that address this problem. We first present the most commonly used dimensionality reduction technique called *principal component analysis* (PCA). We then introduce a kernelized version of PCA (KPCA) and show the connection between KPCA and manifold learning



(a) (b)

Figure 15.1
The "Swiss roll" dataset. (a) high-dimensional representation. (b) lower-dimensional representation.

algorithms. We conclude with a presentation of the Johnson-Lindenstrauss lemma, a classical theoretical result that has inspired a variety of dimensionality reduction methods based on the concept of random projections. The discussion in this chapter relies on basic matrix properties that are reviewed in appendix A.

15.1 Principal component analysis

Fix $k \in [N]$ and let $\mathbf{X}$ be a mean-centered data matrix, that is, $\sum_{i=1}^{m} \mathbf{x}_i = \mathbf{0}$. Define $\mathcal{P}_k$ as the set of N-dimensional rank-k orthogonal projection matrices. PCA consists of projecting the N-dimensional input data onto the k-dimensional linear subspace that minimizes *reconstruction error*, that is the sum of the squared L_2-distances between the original data and the projected data. Thus, the PCA algorithm is completely defined by the orthogonal projection matrix solution $\mathbf{P}^*$ of the following minimization problem:

$$\min_{\mathbf{P} \in \mathcal{P}_k} \|\mathbf{PX} - \mathbf{X}\|_F^2 . \tag{15.1}$$

The following theorem shows that PCA coincides with the projection of each data point onto the k top singular vectors of the sample covariance matrix, i.e., $\mathbf{C} = \frac{1}{m}\mathbf{X}\mathbf{X}^\top$ for the mean-centered data matrix $\mathbf{X}$. Figure 15.2 illustrates the basic intuition behind PCA, showing how two-dimensional data points with highly correlated features can be more succinctly represented with a one-dimensional representation that captures most of the variance in the data.

Theorem 15.1 *Let $\mathbf{P}^* \in \mathcal{P}_k$ be the PCA solution, i.e., the orthogonal projection matrix solution of* (15.1)*. Then, $\mathbf{P}^* = \mathbf{U}_k\mathbf{U}_k^\top$, where $\mathbf{U}_k \in \mathbb{R}^{N \times k}$ is the matrix*

formed by the top k singular vectors of $\mathbf{C} = \frac{1}{m}\mathbf{X}\mathbf{X}^\top$, the sample covariance matrix corresponding to $\mathbf{X}$. Moreover, the associated k-dimensional representation of $\mathbf{X}$ is given by $\mathbf{Y} = \mathbf{U}_k^\top \mathbf{X}$.

Proof: Let $\mathbf{P} = \mathbf{P}^\top$ be an orthogonal projection matrix. By the definition of the Frobenius norm, the linearity of the trace operator and the fact that $\mathbf{P}$ is idempotent, i.e., $\mathbf{P}^2 = \mathbf{P}$, we observe that

$$\begin{aligned}\|\mathbf{P}\mathbf{X} - \mathbf{X}\|_F^2 &= \operatorname{Tr}[(\mathbf{P}\mathbf{X} - \mathbf{X})^\top(\mathbf{P}\mathbf{X} - \mathbf{X})] = \operatorname{Tr}[\mathbf{X}^\top\mathbf{P}^2\mathbf{X} - 2\mathbf{X}^\top\mathbf{P}\mathbf{X} + \mathbf{X}^\top\mathbf{X}] \\ &= -\operatorname{Tr}[\mathbf{X}^\top\mathbf{P}\mathbf{X}] + \operatorname{Tr}[\mathbf{X}^\top\mathbf{X}]\,.\end{aligned}$$

Since $\operatorname{Tr}[\mathbf{X}^\top\mathbf{X}]$ is a constant with respect to $\mathbf{P}$, we have

$$\operatorname*{argmin}_{\mathbf{P}\in\mathcal{P}_k} \|\mathbf{P}\mathbf{X} - \mathbf{X}\|_F^2 = \operatorname*{argmax}_{\mathbf{P}\in\mathcal{P}_k} \operatorname{Tr}[\mathbf{X}^\top\mathbf{P}\mathbf{X}]\,. \tag{15.2}$$

By definition of orthogonal projections in $\mathcal{P}_k$, $\mathbf{P} = \mathbf{U}\mathbf{U}^\top$ for some $\mathbf{U} \in \mathbb{R}^{N\times k}$ containing orthogonal columns. Using the invariance of the trace operator under cyclic permutations and the orthogonality of the columns of $\mathbf{U}$, we have

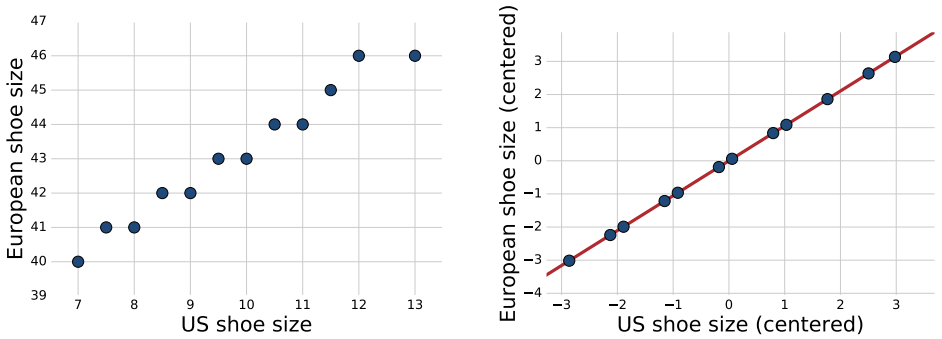
$$\operatorname{Tr}[\mathbf{X}^\top\mathbf{P}\mathbf{X}] = \operatorname{Tr}[\mathbf{U}^\top\mathbf{X}\mathbf{X}^\top\mathbf{U}] = \sum_{i=1}^{k} \mathbf{u}_i^\top\mathbf{X}\mathbf{X}^\top\mathbf{u}_i\,,$$

where $\mathbf{u}_i$ is the ith column of $\mathbf{U}$. By the Rayleigh quotient (section A.2.3), it is clear that the largest k singular vectors of $\mathbf{X}\mathbf{X}^\top$ maximize the rightmost sum above. Since $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{C}$ differ only by a scaling factor, they have the same singular vectors, and thus $\mathbf{U}_k$ maximizes this sum, which proves the first statement of the theorem. Finally, since $\mathbf{P}\mathbf{X} = \mathbf{U}_k\mathbf{U}_k^\top\mathbf{X}$, $\mathbf{Y} = \mathbf{U}_k^\top\mathbf{X}$ is a k-dimensional representation of $\mathbf{X}$ with $\mathbf{U}_k$ as the basis vectors. □

By definition of the covariance matrix, the top singular vectors of $\mathbf{C}$ are the directions of maximal variance in the data, and the associated singular values are equal to these variances. Hence, PCA can also be viewed as projecting onto the subspace of maximal variance. Under this interpretation, the first principal component is derived from projection onto the direction of maximal variance, given by the top singular vector of $\mathbf{C}$. Similarly, the ith principal component, for $1 \le i \le k$, is derived from projection onto the ith direction of maximal variance, subject to orthogonality constraints to the previous $i-1$ directions of maximal variance (see exercise 15.1 for more details).

15.2 Kernel principal component analysis (KPCA)

In the previous section, we presented the PCA algorithm, which involved projecting onto the singular vectors of the sample covariance matrix $\mathbf{C}$. In this section, we



(a) (b)

Figure 15.2
Example of PCA. (a) Two-dimensional data points with features capturing shoe size measured with different units. (b) One-dimensional representation that captures the most variance in the data, generated by projecting onto largest principal component (red line) of the mean-centered data points.

present a kernelized version of PCA, called KPCA. In the KPCA setting, $\mathbf{\Phi}$ is a feature mapping to an arbitrary RKHS (not necessarily to $\mathbb{R}^N$) and we work exclusively with a kernel function K corresponding to the inner product in this RKHS. The KPCA algorithm can thus be defined as a generalization of PCA in which the input data is projected onto the top principle components in this RKHS. We will show the relationship between PCA and KPCA by drawing upon the deep connections among the SVDs of $\mathbf{X}$, $\mathbf{C}$ and $\mathbf{K}$. We then illustrate how various manifold learning algorithms can be interpreted as special instances of KPCA.

Let K be a PDS kernel defined over $\mathcal{X} \times \mathcal{X}$ and define the kernel matrix as $\mathbf{K} = \mathbf{X}^\top \mathbf{X}$. Since $\mathbf{X}$ admits the following singular value decomposition: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, $\mathbf{C}$ and $\mathbf{K}$ can be rewritten as follows:

$$\mathbf{C} = \frac{1}{m}\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top \qquad \mathbf{K} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top, \tag{15.3}$$

where $\mathbf{\Lambda} = \mathbf{\Sigma}^2$ is the diagonal matrix of the singular values (equivalently eigenvalues) of $m\mathbf{C}$ and $\mathbf{U}$ is the matrix of the singular vectors (equivalently eigenvectors) of $\mathbf{C}$ (and $m\mathbf{C}$).

Starting with the SVD of $\mathbf{X}$, note that right multiplying by $\mathbf{V}\mathbf{\Sigma}^{-1}$ and using the relationship between $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ yields $\mathbf{U} = \mathbf{X}\mathbf{V}\mathbf{\Lambda}^{-1/2}$. Thus, the singular vector $\mathbf{u}$ of $\mathbf{C}$ associated to the singular value λ/m coincides with $\frac{\mathbf{X}\mathbf{v}}{\sqrt{\lambda}}$, where $\mathbf{v}$ is the singular vector of $\mathbf{K}$ associated to λ. Now fix an arbitrary feature vector $\mathbf{x} = \mathbf{\Phi}(x)$ for $x \in \mathcal{X}$. Then, following the expression for $\mathbf{Y}$ in theorem 15.1, the one-dimensional

representation of $\mathbf{x}$ derived by projection onto $\mathbf{P}_u = \mathbf{u}\mathbf{u}^\top$ is defined by

$$\mathbf{x}^\top \mathbf{u} = \mathbf{x}^\top \frac{\mathbf{X}\mathbf{v}}{\sqrt{\lambda}} = \frac{\mathbf{k}_x^\top \mathbf{v}}{\sqrt{\lambda}}\,, \tag{15.4}$$

where $\mathbf{k}_x = (K(x_1, x), \ldots, K(x_m, x))^\top$. If $\mathbf{x}$ is one of the data points, i.e., $\mathbf{x} = \mathbf{x}_i$ for $1 \leq i \leq m$, then $\mathbf{k}_x$ is the ith column of $\mathbf{K}$ and (15.4) can be simplified as follows:

$$\mathbf{x}^\top \mathbf{u} = \frac{\mathbf{k}_x^\top \mathbf{v}}{\sqrt{\lambda}} = \frac{\lambda v_i}{\sqrt{\lambda}} = \sqrt{\lambda} v_i\,, \tag{15.5}$$

where v_i is the ith component of $\mathbf{v}$. More generally, the PCA solution of theorem 15.1 can be fully defined by the top k singular vectors (or eigenvectors) of $\mathbf{K}$, $\mathbf{v}_1, \ldots, \mathbf{v}_k$, and the corresponding singular values (or eigenvalues). This alternative derivation of the PCA solution in terms of $\mathbf{K}$ precisely defines the KPCA solution, providing a generalization of PCA via the use of PDS kernels (see chapter 6 for more details on kernel methods).

15.3 KPCA and manifold learning

Several manifold learning techniques have been proposed as non-linear methods for dimensionality reduction. These algorithms implicitly assume that high-dimensional data lie on or near a low-dimensional non-linear manifold embedded in the input space. They aim to learn this manifold structure by finding a low-dimensional space that in some way preserves the local structure of high-dimensional input data. For instance, the Isomap algorithm aims to preserve approximate geodesic distances, or distances along the manifold, between all pairs of data points. Other algorithms, such as Laplacian eigenmaps and locally linear embedding, focus only on preserving local neighborhood relationships in the high-dimensional space. We will next describe these classical manifold learning algorithms and then interpret them as specific instances of KPCA.

15.3.1 Isomap

Isomap aims to extract a low-dimensional data representation that best preserves all pairwise distances between input points, as measured by their geodesic distances along the underlying manifold. It approximates geodesic distance assuming that L_2 distance provides good approximations for nearby points, and for faraway points it estimates distance as a series of hops between neighboring points. The Isomap algorithm works as follows:

1. Find the t nearest neighbors for each data point based on L_2 distance and construct an undirected neighborhood graph, denoted by $\mathcal{G}$, with points as nodes and links between neighbors as edges.
2. Compute the approximate geodesic distances, Δ_{ij}, between all pairs of nodes (i, j) by computing all-pairs shortest distances in $\mathcal{G}$ using, for instance, the Floyd-Warshall algorithm.
3. Convert the squared distance matrix into a $m \times m$ similarity matrix by performing double centering, i.e., compute $\mathbf{K}_{\text{Iso}} = -\frac{1}{2}\mathbf{H}\Delta\mathbf{H}$, where Δ is the squared distance matrix, $\mathbf{H} = \mathbf{I}_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$ is the centering matrix, $\mathbf{I}_m$ is the $m \times m$ identity matrix and $\mathbf{1}$ is a column vector of all ones (for more details on double centering see exercise 15.2).
4. Find the optimal k-dimensional representation, $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n$, such that $\mathbf{Y} = \text{argmin}_{\mathbf{Y}'} \sum_{i,j} \left(\|\mathbf{y}'_i - \mathbf{y}'_j\|_2^2 - \Delta_{ij}^2\right)$. The solution is given by,

$$\mathbf{Y} = (\mathbf{\Sigma}_{\text{Iso},k})^{1/2}\mathbf{U}_{\text{Iso},k}^\top \tag{15.6}$$

   where $\mathbf{\Sigma}_{\text{Iso},k}$ is the diagonal matrix of the top k singular values of $\mathbf{K}_{\text{Iso}}$ and $\mathbf{U}_{\text{Iso},k}$ are the associated singular vectors.

$\mathbf{K}_{\text{Iso}}$ can naturally be viewed as a kernel matrix, thus providing a simple connection between Isomap and KPCA. Note, however, that this interpretation is valid only when $\mathbf{K}_{\text{Iso}}$ is in fact positive semidefinite, which is indeed the case in the continuum limit for a smooth manifold.

15.3.2 Laplacian eigenmaps

The *Laplacian eigenmaps* algorithm aims to find a low-dimensional representation that best preserves neighborhood relations as measured by a weight matrix $\mathbf{W}$. The algorithm works as follows:

1. Find t nearest neighbors for each point.
2. Construct $\mathbf{W}$, a sparse, symmetric $m \times m$ matrix, where $\mathbf{W}_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma^2\right)$ if $(\mathbf{x}_i, \mathbf{x}_j)$ are neighbors, 0 otherwise, and σ is a scaling parameter.
3. Construct the diagonal matrix $\mathbf{D}$, such that $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$.
4. Find the k-dimensional representation by minimizing the weighted distance between neighbors as,

$$\mathbf{Y} = \underset{\mathbf{Y}'}{\text{argmin}} \sum_{i,j} \mathbf{W}_{ij}\|\mathbf{y}'_i - \mathbf{y}'_j\|_2^2. \tag{15.7}$$

   This objective function penalizes nearby inputs for being mapped to faraway outputs, with "nearness" measured by the weight matrix $\mathbf{W}$. The solution to the minimization in (15.7) is $\mathbf{Y} = \mathbf{U}_{\mathbf{L},k}^\top$, where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian

and $\mathbf{U}_{\mathbf{L},k}^{\top}$ are the bottom k singular vectors of $\mathbf{L}$, excluding the last singular vector corresponding to the singular value 0 (assuming that the underlying neighborhood graph is connected).

The solution to (15.7) can also be interpreted as finding the largest singular vectors of $\mathbf{L}^{\dagger}$, the pseudo-inverse of $\mathbf{L}$. Defining $\mathbf{K}_{\mathbf{L}} = \mathbf{L}^{\dagger}$ we can thus view Laplacian Eigenmaps as an instance of KPCA in which the output dimensions are normalized to have unit variance, which corresponds to setting $\lambda = 1$ in (15.5). Moreover, it can be shown that $\mathbf{K}_{\mathbf{L}}$ is the kernel matrix associated with the commute times of diffusion on the underlying neighborhood graph, where the commute time between nodes i and j in a graph is the expected time taken for a random walk to start at node i, reach node j and then return to i.

15.3.3 Locally linear embedding (LLE)

The *locally linear embedding* (LLE) algorithm also aims to find a low-dimensional representation that preserves neighborhood relations as measured by a weight matrix $\mathbf{W}$. The algorithm works as follows:

1. Find t nearest neighbors for each point.
2. Construct $\mathbf{W}$, a sparse, symmetric $m \times m$ matrix, whose ith row sums to one and contains the linear coefficients that optimally reconstruct $\mathbf{x}_i$ from its t neighbors. More specifically, if we assume that the ith row of $\mathbf{W}$ sums to one, then the reconstruction error is
$$\Big(\mathbf{x}_i - \sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij}\mathbf{x}_j\Big)^2 = \Big(\sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij}(\mathbf{x}_i - \mathbf{x}_j)\Big)^2 = \sum_{j,k \in \mathcal{N}_i} \mathbf{W}_{ij}\mathbf{W}_{ik}\mathbf{C}'_{jk} \tag{15.8}$$
where $\mathcal{N}_i$ is the set of indices of the neighbors of point $\mathbf{x}_i$ and $\mathbf{C}'_{jk} = (\mathbf{x}_i - \mathbf{x}_j)^{\top}(\mathbf{x}_i - \mathbf{x}_k)$ the local covariance matrix. Minimizing this expression with the constraint $\sum_j \mathbf{W}_{ij} = 1$ gives the solution
$$\mathbf{W}_{ij} = \frac{\sum_k (\mathbf{C}'^{-1})_{jk}}{\sum_{st} (\mathbf{C}'^{-1})_{st}}\,. \tag{15.9}$$
Note that the solution can be equivalently obtained by first solving the system of linear equations $\sum_j \mathbf{C}'_{kj}\mathbf{W}_{ij} = 1$, for $k \in \mathcal{N}_i$, and then normalizing so that the weights sum to one.
3. Find the k-dimensional representation that best obeys neighborhood relations as specified by $\mathbf{W}$, i.e.,
$$\mathbf{Y} = \operatorname*{argmin}_{\mathbf{Y}'} \sum_i \Big(\mathbf{y}'_i - \sum_j \mathbf{W}_{ij}\mathbf{y}'_j\Big)^2. \tag{15.10}$$

The solution to the minimization in (15.10) is $\mathbf{Y} = \mathbf{U}_{\mathbf{M},k}^\top$, where $\mathbf{M} = (\mathbf{I} - \mathbf{W}^\top)(\mathbf{I} - \mathbf{W}^\top)$ and $\mathbf{U}_{\mathbf{M},k}^\top$ are the bottom k singular vectors of $\mathbf{M}$, excluding the last singular vector corresponding to the singular value 0.

As discussed in exercise 15.5, LLE coincides with KPCA used with a particular kernel matrix $\mathbf{K}_{LLE}$ whereby the output dimensions are normalized to have unit variance (as in the case of Laplacian Eigenmaps).

15.4 Johnson-Lindenstrauss lemma

The Johnson-Lindenstrauss lemma is a fundamental result in dimensionality reduction that states that any m points in high-dimensional space can be mapped to a much lower dimension, $k \geq O(\frac{\log m}{\epsilon^2})$, without distorting pairwise distance between any two points by more than a factor of $(1 \pm \epsilon)$. In fact, such a mapping can be found in randomized polynomial time by projecting the high-dimensional points onto randomly chosen k-dimensional linear subspaces. The Johnson-Lindenstrauss lemma is formally presented in lemma 15.4. The proof of this lemma hinges on lemma 15.2 and lemma 15.3, and it is an example of the "probabilistic method", in which probabilistic arguments lead to a deterministic statement. Moreover, as we will see, the Johnson-Lindenstrauss lemma follows by showing that the squared norm of a random vector is sharply concentrated around its mean when the vector is projected onto a k-dimensional random subspace.

First, we prove the following property of the χ^2 distribution (see definition C.7 in appendix), which will be used in lemma 15.3.

Lemma 15.2 *Let Q be a random variable following a χ^2 distribution with k degrees of freedom. Then, for any $0 < \epsilon < 1/2$, the following inequality holds:*

$$\mathbb{P}[(1-\epsilon)k \leq Q \leq (1+\epsilon)k] \geq 1 - 2e^{-(\epsilon^2-\epsilon^3)k/4}. \tag{15.11}$$

Proof: By Markov's inequality, we can write

$$\begin{aligned}\mathbb{P}[Q \geq (1+\epsilon)k] = \mathbb{P}[\exp(\lambda Q) \geq \exp(\lambda(1+\epsilon)k)] &\leq \frac{\mathbb{E}[\exp(\lambda Q)]}{\exp(\lambda(1+\epsilon)k)} \\ &= \frac{(1-2\lambda)^{-k/2}}{\exp(\lambda(1+\epsilon)k)},\end{aligned}$$

where we used for the final equality the expression of the moment-generating function of a χ^2 distribution, $\mathbb{E}[\exp(\lambda Q)]$, for $\lambda < 1/2$ (equation (C.25)). Choosing $\lambda = \frac{\epsilon}{2(1+\epsilon)} < 1/2$, which minimizes the right-hand side of the final equality, and using the inequality $1 + \epsilon \leq \exp(\epsilon - (\epsilon^2 - \epsilon^3)/2)$ yield

$$\mathbb{P}[Q \geq (1+\epsilon)k] \leq \left(\frac{1+\epsilon}{\exp(\epsilon)}\right)^{k/2} \leq \left(\frac{\exp\left(\epsilon - \frac{\epsilon^2-\epsilon^3}{2}\right)}{\exp(\epsilon)}\right)^{k/2} = \exp\Big(-\frac{k}{4}(\epsilon^2-\epsilon^3)\Big).$$

The statement of the lemma follows by using similar techniques to bound $\mathbb{P}[Q \leq (1-\epsilon)k]$ and by applying the union bound. □

Lemma 15.3 *Let $\mathbf{x} \in \mathbb{R}^N$, define $k < N$ and assume that entries in $\mathbf{A} \in \mathbb{R}^{k \times N}$ are sampled independently from the standard normal distribution, $N(0,1)$. Then, for any $0 < \epsilon < 1/2$,*

$$\mathbb{P}\left[(1-\epsilon)\|\mathbf{x}\|^2 \leq \|\frac{1}{\sqrt{k}}\mathbf{A}\mathbf{x}\|^2 \leq (1+\epsilon)\|\mathbf{x}\|^2\right] \geq 1 - 2e^{-(\epsilon^2-\epsilon^3)k/4}\,. \tag{15.12}$$

Proof: Let $\widehat{\mathbf{x}} = \mathbf{A}\mathbf{x}$ and observe that

$$\mathbb{E}[\widehat{x}_j^2] = \mathbb{E}\left[\Big(\sum_{i=1}^N A_{ji}x_i\Big)^2\right] = \mathbb{E}\left[\sum_{i=1}^N A_{ji}^2 x_i^2\right] = \sum_{i=1}^N x_i^2 = \|\mathbf{x}\|^2\,.$$

The second and third equalities follow from the independence and unit variance, respectively, of the A_{ij}. Now, define $T_j = \widehat{x}_j/\|\mathbf{x}\|$ and note that the T_js are independent standard normal random variables since the A_{ij} are i.i.d. standard normal random variables and $\mathbb{E}[\widehat{x}_j^2] = \|\mathbf{x}\|^2$. Thus, the variable Q defined by $Q = \sum_{j=1}^k T_j^2$ follows a χ^2 distribution with k degrees of freedom and we have

$$\begin{aligned}\mathbb{P}\left[(1-\epsilon)\|\mathbf{x}\|^2 \leq \frac{\|\widehat{\mathbf{x}}\|^2}{k} \leq (1+\epsilon)\|\mathbf{x}\|^2\right] &= \mathbb{P}\left[(1-\epsilon)k \leq \sum_{j=1}^k T_j^2 \leq (1+\epsilon)k\right] \\ &= \mathbb{P}\left[(1-\epsilon)k \leq Q \leq (1+\epsilon)k\right] \\ &\geq 1 - 2e^{-(\epsilon^2-\epsilon^3)k/4}\,,\end{aligned}$$

where the final inequality holds by lemma 15.2, thus proving the statement of the lemma. □

Lemma 15.4 (Johnson-Lindenstrauss) *For any $0 < \epsilon < 1/2$ and any integer $m > 4$, let $k = \frac{20\log m}{\epsilon^2}$. Then for any set V of m points in $\mathbb{R}^N$, there exists a map $f\colon \mathbb{R}^N \to \mathbb{R}^k$ such that for all $\mathbf{u}, \mathbf{v} \in V$,*

$$(1-\epsilon)\|\mathbf{u}-\mathbf{v}\|^2 \leq \|f(\mathbf{u}) - f(\mathbf{v})\|^2 \leq (1+\epsilon)\|\mathbf{u}-\mathbf{v}\|^2. \tag{15.13}$$

Proof: Let $f = \frac{1}{\sqrt{k}}\mathbf{A}$ where $k < N$ and entries in $\mathbf{A} \in \mathbb{R}^{k\times N}$ are sampled independently from the standard normal distribution, $N(0,1)$. For fixed $\mathbf{u}, \mathbf{v} \in V$, we can apply lemma 15.3, with $\mathbf{x} = \mathbf{u} - \mathbf{v}$, to lower bound the success probability by $1 - 2e^{-(\epsilon^2-\epsilon^3)k/4}$. Applying the union bound over the $\mathrm{O}(m^2)$ pairs in V, setting $k = \frac{20}{\epsilon^2}\log m$ and upper bounding ϵ by $1/2$, we have

$$\mathbb{P}[\mathit{success}] \geq 1 - 2m^2 e^{-(\epsilon^2-\epsilon^3)k/4} = 1 - 2m^{5\epsilon-3} > 1 - 2m^{-1/2} > 0\,.$$

Since the success probability is strictly greater than zero, a map that satisfies the desired conditions must exist, thus proving the statement of the lemma. □

15.5 Chapter notes

PCA was introduced in the early 1900s by Pearson [1901]. KPCA was introduced roughly a century later, and our presentation of KPCA is a more concise derivation of results given by Mika et al. [1999]. Isomap and LLE were pioneering works on non-linear dimensionality reduction introduced by Tenenbaum et al. [2000], Roweis and Saul [2000]. Isomap itself is a generalization of a standard linear dimensionality reduction technique called Multidimensional Scaling [Cox and Cox, 2000]. Isomap and LLE led to the development of several related algorithms for manifold learning, e.g., Laplacian Eigenmaps and Maximum Variance Unfolding [Belkin and Niyogi, 2001, Weinberger and Saul, 2006]. As shown in this chapter, classical manifold learning algorithms are special instances of KPCA [Ham et al., 2004]. The Johnson-Lindenstrauss lemma was introduced by Johnson and Lindenstrauss [1984], though our proof of the lemma follows Vempala [2004]. Other simplified proofs of this lemma have also been presented, including Dasgupta and Gupta [2003].

15.6 Exercises

15.1 PCA and maximal variance. Let $\mathbf{X}$ be an *uncentered* data matrix and let $\bar{\mathbf{x}} = \frac{1}{m}\sum_i \mathbf{x}_i$ be the sample mean of the columns of $\mathbf{X}$.

(a) Show that the variance of one-dimensional projections of the data onto an arbitrary vector $\mathbf{u}$ equals $\mathbf{u}^\top \mathbf{C}\mathbf{u}$, where $\mathbf{C} = \frac{1}{m}\sum_i (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$ is the sample covariance matrix.

(b) Show that PCA with $k = 1$ projects the data onto the direction (i.e., $\mathbf{u}^\top \mathbf{u} = 1$) of maximal variance.

15.2 Double centering. In this problem we will prove the correctness of the double centering step in Isomap when working with Euclidean distances. Define $\mathbf{X}$ and $\bar{\mathbf{x}}$ as in exercise 15.1, and define $\mathbf{X}^*$ as the centered version of $\mathbf{X}$, that is, let $\mathbf{x}_i^* = \mathbf{x}_i - \bar{\mathbf{x}}$ be the ith column of $\mathbf{X}^*$. Let $\mathbf{K} = \mathbf{X}^\top \mathbf{X}$, and let $\mathbf{D}$ denote the Euclidean distance matrix, i.e., $\mathbf{D}_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$.

(a) Show that $\mathbf{K}_{ij} = \frac{1}{2}(\mathbf{K}_{ii} + \mathbf{K}_{jj} + \mathbf{D}_{ij}^2)$.

(b) Show that $\mathbf{K}^* = {\mathbf{X}^*}^\top \mathbf{X}^* = \mathbf{K} - \frac{1}{m}\mathbf{K}\mathbf{1}\mathbf{1}^\top - \frac{1}{m}\mathbf{1}\mathbf{1}^\top\mathbf{K} + \frac{1}{m^2}\mathbf{1}\mathbf{1}^\top\mathbf{K}\mathbf{1}\mathbf{1}^\top$.

(c) Using the results from (a) and (b) show that

$$\mathbf{K}_{ij}^* = -\frac{1}{2}\left[\mathbf{D}_{ij}^2 - \frac{1}{m}\sum_{k=1}^{m}\mathbf{D}_{ik}^2 - \frac{1}{m}\sum_{k=1}^{m}\mathbf{D}_{kj}^2 + \bar{\mathbf{D}}\right],$$

where $\bar{\mathbf{D}} = \frac{1}{m^2}\sum_u \sum_v \mathbf{D}^2_{u,v}$ is the mean of the m^2 entries in $\mathbf{D}$.

(d) Show that $\mathbf{K}^* = -\frac{1}{2}\mathbf{HDH}$, where $\mathbf{H} = \mathbf{I}_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$.

15.3 Laplacian eigenmaps. Assume $k = 1$ and we seek a one-dimensional representation $\mathbf{y}$. Show that (15.7) is equivalent to $\mathbf{y} = \text{argmin}_{\mathbf{y}'}\, \mathbf{y}'^\top \mathbf{L}\mathbf{y}'$, where $\mathbf{L}$ is the graph Laplacian.

15.4 Nyström method. Define the following block representation of a kernel matrix:

$$\mathbf{K} = \begin{bmatrix} \mathbf{W} & \mathbf{K}_{21}^\top \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} \mathbf{W} \\ \mathbf{K}_{21} \end{bmatrix}.$$

The Nyström method uses $\mathbf{W} \in \mathbb{R}^{l \times l}$ and $\mathbf{C} \in \mathbb{R}^{m \times l}$ to generate the approximation $\widetilde{\mathbf{K}} = \mathbf{C}\mathbf{W}^\dagger\mathbf{C}^\top \approx \mathbf{K}$.

(a) Show that $\mathbf{W}$ is SPSD and that $\|\mathbf{K} - \widetilde{\mathbf{K}}\|_F = \|\mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{W}^\dagger\mathbf{K}_{21}^\top\|_F$.

(b) Let $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$ for some $\mathbf{X} \in \mathbb{R}^{N \times m}$, and let $\mathbf{X}' \in \mathbb{R}^{N \times l}$ be the first l columns of $\mathbf{X}$. Show that $\widetilde{\mathbf{K}} = \mathbf{X}^\top\mathbf{P}_{U_{X'}}\mathbf{X}$, where $\mathbf{P}_{U_{X'}}$ is the orthogonal projection onto the span of the left singular vectors of $\mathbf{X}'$.

(c) Is $\widetilde{\mathbf{K}}$ SPSD?

(d) If $\text{rank}(\mathbf{K}) = \text{rank}(\mathbf{W}) = r \ll m$, show that $\widetilde{\mathbf{K}} = \mathbf{K}$. Note: this statement holds whenever $\text{rank}(\mathbf{K}) = \text{rank}(\mathbf{W})$, but is of interest mainly in the low-rank setting.

(e) If $m = 20\text{M}$ and $\mathbf{K}$ is a dense matrix, how much space is required to store $\mathbf{K}$ if each entry is stored as a double? How much space is required by the Nyström method if $l = 10\text{K}$?

15.5 Expression for $\mathbf{K}_{LLE}$. Show the connection between LLE and KPCA by deriving the expression for $\mathbf{K}_{LLE}$.

15.6 Random projection, PCA, and nearest neighbors.

(a) Download the MNIST test set of handwritten digits at:

`http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz.`

Create a data matrix $\mathbf{X} \in \mathbb{R}^{N \times m}$ from the first $m = 2{,}000$ instances of this dataset (the dimension of each instance should be $N = 784$).

(b) Find the ten nearest neighbors for each point in $\mathbf{X}$, that is, compute $\mathcal{N}_{i,10}$ for $1 \le i \le m$, where $\mathcal{N}_{i,t}$ denotes the set of the t nearest neighbors for the

ith datapoint and nearest neighbors are defined with respect to the L_2 norm. Also compute $\mathcal{N}_{i,50}$ for all i.

(c) Generate $\tilde{\mathbf{X}} = \mathbf{A}\mathbf{X}$, where $\mathbf{A} \in \mathbb{R}^{k \times N}$, $k = 100$ and entries of $\mathbf{A}$ are sampled independently from the standard normal distribution. Find the ten nearest neighbors for each point in $\tilde{\mathbf{X}}$, that is, compute $\tilde{\mathcal{N}}_{i,10}$ for $1 \leq i \leq m$.

(d) Report the quality of approximation by computing $\text{score}_{10} = \frac{1}{m}\sum_{i=1}^{m} |\mathcal{N}_{i,10} \cap \tilde{\mathcal{N}}_{i,10}|$. Similarly, compute $\text{score}_{50} = \frac{1}{m}\sum_{i=1}^{m} |\mathcal{N}_{i,50} \cap \tilde{\mathcal{N}}_{i,10}|$.

(e) Generate two plots that show score_{10} and score_{50} as functions of k (i.e., perform steps (c) and (d) for $k = \{1, 10, 50, 100, 250, 500\}$). Provide a one- or two-sentence explanation of these plots.

(f) Generate similar plots as in (e) using PCA (with various values of k) to generate $\tilde{\mathbf{X}}$ and subsequently compute nearest neighbors. Are the nearest neighbor approximations generated via PCA better or worse than those generated via random projections? Explain why.

16 Learning Automata and Languages

This chapter presents an introduction to the problem of learning languages. This is a classical problem explored since the early days of formal language theory and computer science, and there is a very large body of literature dealing with related mathematical questions. In this chapter, we present a brief introduction to this problem and concentrate specifically on the question of learning finite automata, which, by itself, has been a topic investigated in multiple forms by thousands of technical papers. We will examine two broad frameworks for learning automata, and for each, we will present an algorithm. In particular, we describe an algorithm for learning automata in which the learner has access to several types of query, and we discuss an algorithm for identifying a sub-class of the family of automata in the limit.

16.1 Introduction

Learning languages is one of the earliest problems discussed in linguistics and computer science. It has been prompted by the remarkable faculty of humans to learn natural languages. Humans are capable of uttering well-formed new sentences at an early age, after having been exposed only to finitely many sentences. Moreover, even at an early age, they can make accurate judgments of grammaticality for new sentences.

In computer science, the problem of learning languages is directly related to that of learning the representation of the computational device generating a language. Thus, for example, learning regular languages is equivalent to learning finite automata, or learning context-free languages or context-free grammars is equivalent to learning pushdown automata.

There are several reasons for examining specifically the problem of learning finite automata. Automata provide natural modeling representations in a variety of different domains including systems, networking, image processing, text and speech

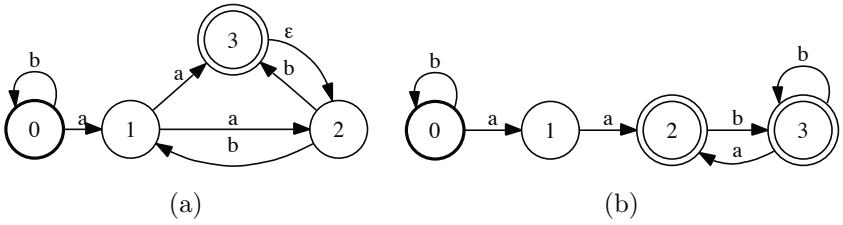


Figure 16.1
(a) A graphical representation of a finite automaton. (b) Equivalent (minimal) deterministic automaton.

processing, logic and many others. Automata can also serve as simple or efficient approximations for more complex devices. For example, in natural language processing, they can be used to approximate context-free languages. When it is possible, learning automata is often efficient, though, as we shall see, the problem is hard in a number of natural scenarios. Thus, learning more complex devices or languages is even harder.

We consider two general learning frameworks: the model of *efficient exact learning* and the model of *identification in the limit*. For each of these models, we briefly discuss the problem of learning automata and describe an algorithm.

We first give a brief review of some basic automata definitions and algorithms, then discuss the problem of efficient exact learning of automata and that of the identification in the limit.

16.2 Finite automata

We will denote by Σ a finite alphabet. The length of a string $x \in \Sigma^*$ over that alphabet is denoted by $|x|$. The *empty string* is denoted by ϵ, thus $|\epsilon| = 0$. For any string $x = x_1 \cdots x_k \in \Sigma^*$ of length $k \geq 0$, we denote by $x[j] = x_1 \cdots x_j$ its prefix of length $j \leq k$ and define $x[0]$ as ϵ.

Finite automata are labeled directed graphs equipped with initial and final states. The following gives a formal definition of these devices.

Definition 16.1 (Finite automata) *A* finite automaton *A is a 5-tuple (Σ, Q, I, F, E) where Σ is a finite alphabet, Q a finite set of states, $I \subseteq Q$ a set of initial states, $F \subseteq Q$ a set of final states, and $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ a finite set of* transitions.

Figure 16.1a shows a simple example of a finite automaton. States are represented by circles. A bold circle indicates an initial state, a double circle a final state. Each transition is represented by an arrow from its origin state to its destination state with its label in $\Sigma \cup \{\epsilon\}$.

A path from an initial state to a final state is said to be an *accepting path*. An automaton is said to be *trim* if all of its states are accessible from an initial state and admit a path to a final state, that is, if all of its states lie on an accepting path. A string $x \in \Sigma^*$ is *accepted* by an automaton A iff x labels an accepting path. For convenience, we will say that $x \in \Sigma^*$ is *rejected* by A when it is not accepted. The set of all strings accepted by A defines the *language accepted by A* denoted by $L(A)$. The class of languages accepted by finite automata coincides with the family of *regular languages*, that is, languages that can be described by *regular expressions*.

Any finite automaton admits an equivalent automaton with no ϵ-*transition*, that is, no transition labeled with the empty string: there exists a general ϵ-removal algorithm that takes as input an automaton and returns an equivalent automaton with no ϵ-transition.

An automaton with no ϵ-*transition* is said to be *deterministic* if it admits a unique initial state and if no two transitions sharing the same label leave any given state. A deterministic finite automaton is often referred to by the acronym *DFA*, while the acronym *NFA* is used for arbitrary automata, that is, non-deterministic finite automata. Any NFA admits an equivalent DFA: there exists a general (exponential-time) *determinization* algorithm that takes as input an NFA with no ϵ-transition and returns an equivalent DFA. Thus, the class of languages accepted by DFAs coincides with that of the languages accepted by NFAs, that is regular languages. For any string $x \in \Sigma^*$ and DFA A, we denote by $A(x)$ the state reached in A when reading x from its unique initial state.

A DFA is said to be *minimal* if it admits no equivalent deterministic automaton with a smaller number of states. There exists a general *minimization* algorithm taking as input a deterministic automaton and returning a minimal one that runs in $O(|E| \log |Q|)$. When the input DFA is *acyclic*, that is when it admits no path forming a cycle, it can be minimized in linear time $O(|Q| + |E|)$. Figure 16.1b shows the minimal DFA equivalent to the NFA of figure 16.1a.

16.3 Efficient exact learning

In the *efficient exact learning* framework, the problem consists of identifying a target concept c from a finite set of examples in time polynomial in the size of the representation of the concept and in an upper bound on the size of the representation of an example. Unlike the PAC-learning framework, in this model, there is no stochastic assumption, instances are not assumed to be drawn according to some unknown distribution. Furthermore, the objective is to identify the target concept

exactly, without any approximation. A concept class $\mathcal{C}$ is said to be efficiently exactly learnable if there is an algorithm for efficient exact learning of any $c \in \mathcal{C}$.

We will consider two different scenarios within the framework of efficiently exact learning: a *passive* and an *active learning* scenario. The passive learning scenario is similar to the standard supervised learning scenario discussed in previous chapters but without any stochastic assumption: the learning algorithm *passively* receives data instances as in the PAC model and returns a hypothesis, but here, instances are not assumed to be drawn from any distribution. In the active learning scenario, the learner *actively* participates in the selection of the training samples by using various types of queries that we will describe. In both cases, we will focus more specifically on the problem of learning automata.

16.3.1 Passive learning

The problem of learning finite automata in this scenario is known as the *minimum consistent DFA learning problem* . It can be formulated as follows: the learner receives a finite sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ with $x_i \in \Sigma^*$ and $y_i \in \{-1, +1\}$ for any $i \in [m]$. If $y_i = +1$, then x_i is an accepted string, otherwise it is rejected. The problem consists of using this sample to learn the smallest DFA A *consistent* with S, that is the automaton with the smallest number of states that accepts the strings of S with label $+1$ and rejects those with label -1. Note that seeking the smallest DFA consistent with S can be viewed as following Occam's razor principle.

The problem just described is distinct from the standard minimization of DFAs. A minimal DFA accepting exactly the strings of S labeled positively may not have the smallest number of states: in general there may be DFAs with fewer states accepting a superset of these strings and rejecting the negatively labeled sample strings. For example, in the simple case $S = ((a, +1), (b, -1))$, a minimal deterministic automaton accepting the unique positively labeled string a or the unique negatively labeled string b admits two states. However, the deterministic automaton accepting the language a^* accepts a and rejects b and has only one state.

Passive learning of finite automata turns out to be a computationally hard problem. The following theorems present several negative results known for this problem.

Theorem 16.2 *The problem of finding the smallest deterministic automaton consistent with a set of accepted or rejected strings is NP-complete.*

Hardness results are known even for a polynomial approximation, as stated by the following theorem.

Theorem 16.3 *If $P \neq NP$, then, no polynomial-time algorithm can be guaranteed to find a DFA consistent with a set of accepted or rejected strings of size smaller than*

a polynomial function of the smallest consistent DFA, even when the alphabet is reduced to just two elements.

Other strong negative results are known for passive learning of finite automata under various cryptographic assumptions.

These negative results for passive learning invite us to consider alternative learning scenarios for finite automata. The next section describes a scenario leading to more positive results where the learner can actively participate in the data selection process using various types of queries.

16.3.2 Learning with queries

The model of *learning with queries* corresponds to that of a (minimal) teacher or oracle and an active learner. In this model, the learner can make the following two types of queries to which an oracle responds:

- *membership queries*: the learner requests the target label $f(x) \in \{-1, +1\}$ of an instance x and receives that label;
- *equivalence queries*: the learner conjectures hypothesis h; it receives the response yes if $h = f$, a counter-example otherwise.

We will say that a concept class $\mathcal{C}$ is *efficiently exactly learnable with membership and equivalence queries* when it is efficiently exactly learnable within this model.

This model is not realistic, since no such oracle is typically available in practice. Nevertheless, it provides a natural framework, which, as we shall see, leads to positive results. Note also that for this model to be significant, equivalence must be computationally testable. This would not be the case for some concept classes such as that of *context-free grammars*, for example, for which the equivalence problem is undecidable. In fact, equivalence must be further efficiently testable, otherwise the response to the learner cannot be supplied in a reasonable amount of time.[21]

Efficient exact learning within this model of learning with queries implies the following variant of PAC-learning: we will say that a concept class $\mathcal{C}$ is *PAC-learnable with membership queries* if it is PAC-learnable by an algorithm that has access to a polynomial number of membership queries.

Theorem 16.4 *Let $\mathcal{C}$ be a concept class that is efficiently exactly learnable with membership and equivalence queries, then $\mathcal{C}$ is PAC-learnable using membership queries.*

Proof: Let $\mathcal{A}$ be an algorithm for efficiently exactly learning $\mathcal{C}$ using membership and equivalence queries. Fix $\epsilon, \delta > 0$. We replace in the execution of $\mathcal{A}$ for learning

[21] For a human oracle, answering membership queries may also become very hard in some cases when the queries are near the class boundaries. This may also make the model difficult to adopt in practice.

target $c \in \mathcal{C}$, each equivalence query by a test of the current hypothesis on a polynomial number of labeled examples. Let $\mathcal{D}$ be the distribution according to which points are drawn. To simulate the tth equivalence query, we draw $m_t = \frac{1}{\epsilon}(\log \frac{1}{\delta} + t \log 2)$ points i.i.d. according to $\mathcal{D}$ to test the current hypothesis h_t. If h_t is consistent with all of these points, then the algorithm stops and returns h_t. Otherwise, one of the points drawn does not belong to h_t, which provides a counter-example.

Since $\mathcal{A}$ learns c exactly, it makes at most T equivalence queries, where T is polynomial in the size of the representation of the target concept and in an upper bound on the size of the representation of an example. Thus, if no equivalence query is positively responded by the simulation, the algorithm will terminate after T equivalence queries and return the correct concept c. Otherwise, the algorithm stops at the first equivalence query positively responded by the simulation. The hypothesis it returns is not an ϵ-approximation only if the equivalence query stopping the algorithm is incorrectly responded positively. By the union bound, since for any fixed $t \in [T]$, $\mathbb{P}[R(h_t) > \epsilon] \leq (1-\epsilon)^{m_t}$, the probability that for some $t \in [T]$, $R(h_t) > \epsilon$ can be bounded as follows:

$$\begin{aligned}\mathbb{P}[\exists t \in [T] \colon R(h_t) > \epsilon] &\leq \sum_{t=1}^{T} \mathbb{P}[R(h_t) > \epsilon] \\ &\leq \sum_{t=1}^{T} (1-\epsilon)^{m_t} \leq \sum_{t=1}^{T} e^{-m_t \epsilon} \leq \sum_{t=1}^{T} \frac{\delta}{2^t} \leq \sum_{t=1}^{+\infty} \frac{\delta}{2^t} = \delta.\end{aligned}$$

Thus, with probability at least $1-\delta$, the hypothesis returned by the algorithm is an ϵ-approximation. Finally, the maximum number of points drawn is $\sum_{t=1}^{T} m_t = \frac{1}{\epsilon}(T \log \frac{1}{\delta} + \frac{T(T+1)}{2} \log 2)$, which is polynomial in $1/\epsilon$, $1/\delta$, and T. Since the rest of the computational cost of $\mathcal{A}$ is also polynomial by assumption, this proves the PAC-learning of $\mathcal{C}$. □

16.3.3 Learning automata with queries

In this section, we describe an algorithm for efficient exact learning of DFAs with membership and equivalence queries. We will denote by A the target DFA and by $\widehat{A}$ the DFA that is the current hypothesis of the algorithm. For the discussion of the algorithm, we assume without loss of generality that A is a minimal DFA.

The algorithm uses two sets of strings, U and V. U is a set of *access strings*: reading an access string $u \in U$ from the initial state of A leads to a state $A(u)$. The algorithm ensures that the states $A(u)$, $u \in U$, are all distinct. To do so, it uses a set V of *distinguishing strings*. Since A is minimal, for two distinct states q and q' of A, there must exist at least one string that leads to a final state from q and not from q', or vice versa. That string helps *distinguish* q and q'. The set of strings V

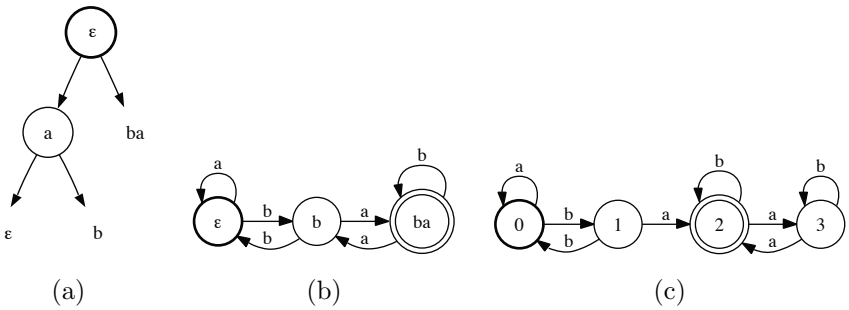


Figure 16.2
(a) Classification tree T, with $U = \{\epsilon, b, ba\}$ and $V = \{\epsilon, a\}$. (b) Current automaton $\widehat{A}$ constructed using T. (c) Target automaton A.

help distinguish any pair of access strings in U. They define in fact a partition of all strings of Σ^*.

The objective of the algorithm is to find at each iteration a new access string distinguished from all previous ones, ultimately obtaining a number of access strings equal to the number of states of A. It can then identify each state $A(u)$ of A with its access string u. To find the destination state of the transition labeled with $a \in \Sigma$ leaving state u, it suffices to determine, using the partition induced by V the access string u' that belongs to the same equivalence class as ua. The finality of each state can be determined in a similar way.

Both sets U and V are maintained by the algorithm via a binary decision tree T similar to those presented in chapter 9. Figure 16.2a shows an example. T defines the partition of all strings induced by the distinguishing strings V. The leaves of T are each labeled with a distinct $u \in U$ and its internal nodes with a string $v \in V$. The decision tree question defined by $v \in V$, given a string $x \in \Sigma^*$, is whether xv is accepted by A, which is determined via a membership query. If accepted, x is assigned to right sub-tree, otherwise to the left sub-tree, and the same is applied recursively with the sub-trees until a leaf is reached. We denote by $T(x)$ the label of the leaf reached. For example, for the tree T of figure 16.2a and target automaton A of figure 16.2c, $T(baa) = b$ since baa is not accepted by A (root question) and $baaa$ is (question at node a). At its initialization step, the algorithm ensures that the root node is labeled with ϵ, which is convenient to check the finality of the strings.

The tentative hypothesis DFA $\widehat{A}$ can be constructed from T as follows. We denote by ConstructAutomaton() the corresponding function. A distinct state $\widehat{A}(u)$ is created for each leaf $u \in U$. The finality of a state $\widehat{A}(u)$ is determined based on the sub-tree of the root node that u belongs to: $\widehat{A}(u)$ is made final iff u belongs

```
QueryLearnAutomata()
 1  t ← MembershipQuery(ε)
 2  T ← T_0
 3  Â ← A_0
 4  while (EquivalenceQuery(Â) ≠ true) do
 5      x ← CounterExample()
 6      if (T = T_0) then
 7          T ← T_1 ▷ nil replaced with x.
 8      else j ← argmin_k A(x[k]) ≠_T Â(x[k])
 9          Split(Â(x[j − 1]))
10      Â ← ConstructAutomaton(T)
11  return Â
```

Figure 16.3
Algorithm for learning automata with membership and equivalence queries. A_0 is a single-state automaton with self-loops labeled with all $a \in \Sigma$. That state is initial. It is final iff $t = \text{true}$. T_0 is a tree with root node labeled with ϵ and two leaves, one labeled with ϵ, the other with nil. the right leaf is labeled with ϵ labels iff $t = \text{true}$. T_1 is the tree obtained from T_0 by replacing nil with x.

to the right sub-tree that is iff $u = \epsilon u$ is accepted by A. The destination of the transition labeled with $a \in \Sigma$ leaving state $\widehat{A}(u)$ is the state $\widehat{A}(v)$ where $v = T(ua)$. Figure 16.2b shows the DFA $\widehat{A}$ constructed from the decision tree of figure 16.2a. For convenience, for any $x \in \Sigma^*$, we denote by $U(\widehat{A}(x))$ the access string identifying state $\widehat{A}(x)$.

Figure 16.3 shows the pseudocode of the algorithm. The initialization steps at lines 1–3 construct a tree T with a single internal node labeled with ϵ and one leaf string labeled with ϵ, the other left undetermined and labeled with nil. They also define a tentative DFA $\widehat{A}$ with a single state with self-loops labeled with all elements of the alphabet. That single state is an initial state. It is made a final state only if ϵ is accepted by the target DFA A, which is determined via the membership query of line 1.

At each iteration of the loop of lines 4–11, an equivalence query is used. If $\widehat{A}$ is not equivalent to A, then a counter-example string x is received (line 5). If T is the tree constructed in the initialization step, then the leaf labeled with nil is replaced with x (lines 6–7). Otherwise, since x is a counter-example, states $A(x)$

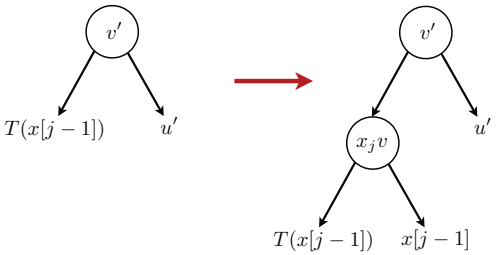


Figure 16.4
Illustration of the splitting procedure SPLIT($\widehat{A}(x[j-1])$).

and $\widehat{A}(x)$ have a different finality; thus, the string x defining $A(x)$ and the access string $U(\widehat{A}(x))$ are assigned to different equivalence classes by T. Thus, there exists a smallest j such that $A(x[j])$ and $\widehat{A}(x[j])$ are not equivalent, that is, such that the prefix $x[j]$ of x and the access string $U(\widehat{A}(x[j]))$ are assigned to different leaves by T. j cannot be 0 since the initialization ensures that $\widehat{A}(\epsilon)$ is an initial state and has the same finality as the initial state $A(\epsilon)$ of A. The equivalence of $A(x[j])$ and $\widehat{A}(x[j])$ is tested by checking the equality of $T(x[j])$ and $T(U(\widehat{A}(x[j])))$, which can be both determined using the tree T and membership queries (line 8).

Now, by definition, $A(x[j-1])$ and $\widehat{A}(x[j-1])$ are equivalent, that is T assigns $x[j-1]$ to the leaf labeled with $U(\widehat{A}(x[j-1]))$. But, $x[j-1]$ and $U(\widehat{A}(x[j-1]))$ must be distinguished since $A(x[j-1])$ and $\widehat{A}(x[j-1])$ admit transitions labeled with the same label x_j to two non-equivalent states. Let v be a distinguishing string for $A(x[j])$ and $\widehat{A}(x[j])$. v can be obtained as the least common ancestor of the leaves labeled with $x[j]$ and $U(\widehat{A}(x[j]))$. To distinguish $x[j-1]$ and $U(\widehat{A}(x[j-1]))$, it suffices to split the leaf of T labeled with $T(x[j-1])$ to create an internal node $x_j v$ dominating a leaf labeled with $x[j-1]$ and another one labeled with $T(x[j-1])$ (line 9). Figure 16.4 illustrates this construction. Thus, this provides a new access string $x[j-1]$ which, by construction, is distinguished from $U(\widehat{A}(x[j-1]))$ and all other access strings.

Thus, the number of access strings (or states of $\widehat{A}$) increases by one at each iteration of the loop. When it reaches the number of states of A, all states of A are of the form $A(u)$ for a distinct $u \in U$. A and $\widehat{A}$ have then the same number of states and in fact $A = \widehat{A}$. Indeed, let $(A(u), a, A(u'))$ be a transition in A, then by definition the equality $A(ua) = A(u')$ holds. The tree T defines a partition of all strings in terms of their distinguishing strings in A. Since in A, ua and u' lead to the same state, they are assigned to the same leaf by T, that is, the leaf labeled with u'. The destination of the transition from $\widehat{A}(u)$ with label a is found by CONSTRUCTAUTOMATON() by determining the leaf in T assigned to ua, that is, u'. Thus, by construction, the same transition $(\widehat{A}(u), a, \widehat{A}(u'))$ is created in $\widehat{A}$.

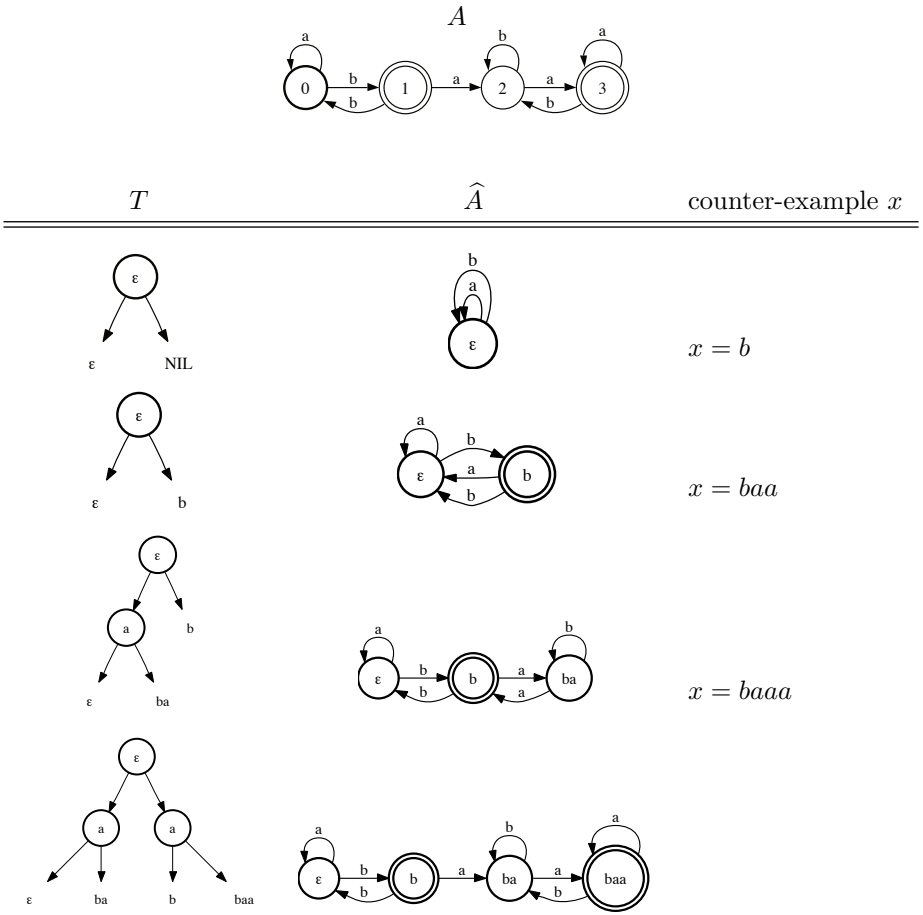


Figure 16.5
Illustration of the execution of Algorithm QUERYLEARNAUTOMATA() for the target automaton A. Each line shows the current decision tree T and the tentative DFA $\widehat{A}$ constructed using T. When $\widehat{A}$ is not equivalent to A, the learner receives a counter-example x indicated in the third column.

Also, a state $A(u)$ of A is final iff u accepted by A that is iff u is assigned to the right sub-tree of the root node by T, which is the criterion determining the finality of $\widehat{A}(u)$. Thus, the automata A and $\widehat{A}$ coincide.

The following is the analysis of the running-time complexity of the algorithm. At each iteration, one new distinguished access string is found associated to a distinct state of A, thus, at most $|A|$ states are created. For each counter-example x, at most $|x|$ tree operations are performed. Constructing $\widehat{A}$ requires $O(|\Sigma||A|)$ tree operations. The cost of a tree operation is $O(|A|)$ since it consists of at most $|A|$

membership queries. Thus, the overall complexity of the algorithm is in $O(|\Sigma||A|^2 + n|A|)$, where n is the maximum length of a counter-example. Note that this analysis assumes that equivalence and membership queries are made in constant time.

Our analysis shows the following result.

Theorem 16.5 (Learning DFAs with queries) *The class of all DFAs is efficiently exactly learnable using membership and equivalence queries.*

Figure 16.5 illustrates a full execution of the algorithm in a specific case. In the next section, we examine a different learning scenario for automata.

16.4 Identification in the limit

In the *identification in the limit framework*, the problem consists of identifying a target concept c exactly after receiving a finite set of examples. A class of languages is said to be *identifiable in the limit* if there exists an algorithm that identifies any language L in that class after examining a finite number of examples and its hypothesis remains unchanged thereafter.

This framework is perhaps less realistic from a computational point of view since it requires no upper bound on the number of instances or the efficiency of the algorithm. Nevertheless, it has been argued by some to be similar to the scenario of humans learning languages. In this framework as well, negative results hold for the general problem of learning DFAs.

Theorem 16.6 *Deterministic automata are not identifiable in the limit from positive examples.*

Some sub-classes of finite automata can however be successfully identified in the limit. Most algorithms for inference of automata are based on a *state-partitioning paradigm*. They start with an initial DFA, typically a tree accepting the finite set of sample strings available and the trivial partition: each block is reduced to one state of the tree. At each iteration, they merge partition blocks while preserving some congruence property. The iteration ends when no other merging is possible and the final partition defines the automaton inferred. Thus, the choice of the congruence fully determines the algorithm and a variety of different algorithms can be defined by varying that choice. A *state-splitting paradigm* can be similarly defined starting from the single-state automaton accepting Σ^*. In this section, we present an algorithm for learning reversible automata, which is a special instance of the general state-partitioning algorithmic paradigm just described.

Let $A = (\Sigma, Q, I, F, E)$ be a DFA and let π be a partition of Q. The DFA defined by the partition π is called the *automaton quotient of A and π*. It is denoted by

A/π and defined as follows: $A/\pi = (\Sigma, \pi, I_\pi, F_\pi, E_\pi)$ with

$$I_\pi = \{B \in \pi \colon I \cap B \neq \emptyset\}$$
$$F_\pi = \{B \in \pi \colon F \cap B \neq \emptyset\}$$
$$E_\pi = \{(B, a, B') \colon \exists (q, a, q') \in E \mid q \in B, q' \in B', B \in \pi, B' \in \pi\}.$$

Let S be a finite set of strings and let $\mathrm{Pref}(S)$ denote the set of prefixes of all strings of S. A *prefix-tree automaton* accepting exactly the set of strings S is a particular DFA denoted by $PT(S) = (\Sigma, \mathrm{Pref}(S), \{\epsilon\}, S, E_S)$ where Σ is the set of alphabet symbols used in S and E_S defined as follows:

$$E_S = \{(x, a, xa) \colon x \in \mathrm{Pref}(S), xa \in \mathrm{Pref}(S)\}.$$

Figure 16.7a shows the prefix-tree automaton of a particular set of strings S.

16.4.1 Learning reversible automata

In this section, we show that the sub-class of *reversible automata* or *reversible languages* can be identified in the limit. In particular, we show that the language can be identified given a *positive presentation*.

A positive presentation of a language L is an infinite sequence $(x_n)_{n \in \mathbb{N}}$ such that $\{x_n \colon n \in \mathbb{N}\} = L$. Thus, in particular, for any $x \in L$ there exists $n \in \mathbb{N}$ such that $x = x_n$. An algorithm identifies L in the limit from a positive presentation if there exists $N \in \mathbb{N}$ such that for $n \geq N$ the hypothesis it returns is L.

Given a DFA A, we define its *reverse* A^R as the automaton derived from A by making the initial state final, the final states initial, and by reversing the direction of every transition. The language accepted by the reverse of A is precisely the language of the reverse (or mirror image) of the strings accepted by A.

Definition 16.7 (Reversible automata) *A finite automaton A is said to be* reversible *iff both A and A^R are deterministic. A language L is said to be* reversible *if it is the language accepted by some reversible automaton.*

Some direct consequences of this definition are that a reversible automaton A has a unique final state and that its reverse A^R is also reversible. Note also that a trim reversible automaton A is minimal. Indeed, if states q and q' in A are equivalent, then, they admit a common string x leading both from q and from q' to a final state. But, by the reverse determinism of A, reading the reverse of x from the final state must lead to a unique state, which implies that $q = q'$.

For any $u \in \Sigma^*$ and any language $L \subseteq \Sigma^*$, let $\mathrm{Suff}_L(u)$ denote the set of all possible suffixes in L for u:

$$\mathrm{Suff}_L(u) = \{v \in \Sigma^* \colon uv \in L\}. \tag{16.1}$$

$\mathrm{Suff}_L(u)$ is also often denoted by $u^{-1}L$. Observe that if L is a reversible language, then the following implication holds for any two strings $u, u' \in \Sigma^*$:

$$\mathrm{Suff}_L(u) \cap \mathrm{Suff}_L(u') \neq \emptyset \implies \mathrm{Suff}_L(u) = \mathrm{Suff}_L(u'). \tag{16.2}$$

Indeed, let A be a reversible automaton accepting L. Let q be the state of A reached from the initial state when reading u and q' the one reached reading u'. If $v \in \mathrm{Suff}_L(u) \cap \mathrm{Suff}_L(u')$, then v can be read both from q and q' to reach the final state. Since A^R is deterministic, reading back the reverse of v from the final state must lead to a unique state, therefore $q = q'$, that is $\mathrm{Suff}_L(u) = \mathrm{Suff}_L(u')$.

Let $A = (\Sigma, Q, \{i_0\}, \{f_0\}, E)$ be a reversible automaton accepting a reversible language L. We define a set of strings S_L as follows:

$$S_L = \{d[q]f[q] : q \in Q\} \cup \{d[q], a, f[q'] : q, q' \in Q, a \in \Sigma\},$$

where $d[q]$ is a string of minimum length from i_0 to q, and $f[q]$ a string of minimum length from q to f_0. As shown by the following proposition, S_L characterizes the language L in the sense that any reversible language containing S_L must contain L.

Proposition 16.8 *Let L be a reversible language. Then, L is the smallest reversible language containing S_L.*

Proof: Let L' be a reversible language containing S_L and let $x = x_1 \cdots x_n$ be a string accepted by L, with $x_k \in \Sigma$ for $k \in [n]$ and $n \geq 1$. For convenience, we also define x_0 as ϵ. Let $(q_0, x_1, q_1) \cdots (q_{n-1}, x_n, q_n)$ be the accepting path in A labeled with x. We show by recurrence that $\mathrm{Suff}_{L'}(x_0 \cdots x_k) = \mathrm{Suff}_{L'}(d[q_k])$ for all $k \in \{0, \ldots, n\}$. Since $d[q_0] = d[i_0] = \epsilon$, this clearly holds for $k = 0$. Now assume that $\mathrm{Suff}_{L'}(x_0 \cdots x_k) = \mathrm{Suff}_{L'}(d[q_k])$ for some $k \in \{0, \ldots, n-1\}$. This implies immediately that $\mathrm{Suff}_{L'}(x_0 \cdots x_k x_{k+1}) = \mathrm{Suff}_{L'}(d[q_k]x_{k+1})$. By definition, S_L contains both $d[q_{k+1}]f[q_{k+1}]$ and $d[q_k]x_{k+1}f[q_{k+1}]$. Since L' includes S_L, the same holds for L'. Thus, $f[q_{k+1}]$ belongs to $\mathrm{Suff}_{L'}(d[q_{k+1}) \cap \mathrm{Suff}_{L'}(d[q_k]x_{k+1})$. In view of (16.2), this implies that $\mathrm{Suff}_{L'}(d[q_k]x_{k+1}) = \mathrm{Suff}_{L'}(d[q_{k+1}])$. Thus, we have $\mathrm{Suff}_{L'}(x_0 \cdots x_k x_{k+1}) = \mathrm{Suff}_{L'}(d[q_{k+1}])$. This shows that $\mathrm{Suff}_{L'}(x_0 \cdots x_k) = \mathrm{Suff}_{L'}(d[q_k])$ holds for all $k \in \{0, \ldots, n\}$, in particular, for $k = n$. Note that since $q_n = f_0$, we have $f[q_n] = \epsilon$, therefore $d[q_n] = d[q_n]f[q_n]$ is in $S_L \subseteq L'$, which implies that $\mathrm{Suff}_{L'}(d[q_n])$ contains ϵ and thus that $\mathrm{Suff}_{L'}(x_0 \cdots x_n)$ contains ϵ. This is equivalent to $x = x_0 \cdots x_n \in L'$. □

Figure 16.6 shows the pseudocode of an algorithm for inferring a reversible automaton from a sample S of m strings $x_1, \ldots, x_m$. The algorithm starts by creating a prefix-tree automaton A for S (line 1) and then iteratively defines a partition π of the states of A, starting with the trivial partition π_0 with one block per state (line 2). The automaton returned is the quotient of A and the final partition π defined.

```
LearnReversibleAutomata(S = (x_1, ..., x_m))
 1  A = (Σ, Q, {i_0}, F, E) ← PT(S)
 2  π ← π_0 ▷ trivial partition.
 3  list ← {(f, f') : f' ∈ F} ▷ f arbitrarily chosen in F.
 4  while list ≠ ∅ do
 5      Remove(list, (q_1, q_2))
 6      if B(q_1, π) ≠ B(q_2, π) then
 7          B_1 ← B(q_1, π)
 8          B_2 ← B(q_2, π)
 9          for all a ∈ Σ do
10              if (succ(B_1, a) ≠ ∅) ∧ (succ(B_2, a) ≠ ∅) then
11                  Add(list, (succ(B_1, a), succ(B_2, a)))
12              if (pred(B_1, a) ≠ ∅ ∧ (pred(B_2, a) ≠ ∅)) then
13                  Add(list, (pred(B_1, a), pred(B_2, a)))
14              Update(succ, pred, B_1, B_2)
15              π ← Merge(π, B_1, B_2)
16  return A/π
```

Figure 16.6
Algorithm for learning reversible automata from a set of positive strings S.

The algorithm maintains a list LIST of pairs of states whose corresponding blocks are to be merged, starting with all pairs of final states (f, f') for an arbitrarily chosen final state $f \in F$ (line 3). We denote by $B(q, \pi)$ the block containing q based on the partition π.

For each block B and alphabet symbol $a \in \Sigma$, the algorithm also maintains a successor $succ(B, a)$, that is, a state that can be reached by reading a from a state of B; $succ(B, a) = \emptyset$ if no such state exists. It maintains similarly the predecessor $pred(B, a)$, which is a state that admits a transition labeled with a leading to a state in B; $pred(B, a) = \emptyset$ if no such state exists.

Then, while LIST is not empty, a pair is removed from LIST and processed as follows. If the pair (q_1, q_1') has not been already merged, the pairs formed by the successors and predecessors of $B_1 = B(q_1, \pi)$ and $B_2 = B(q_2, \pi)$ are added to LIST

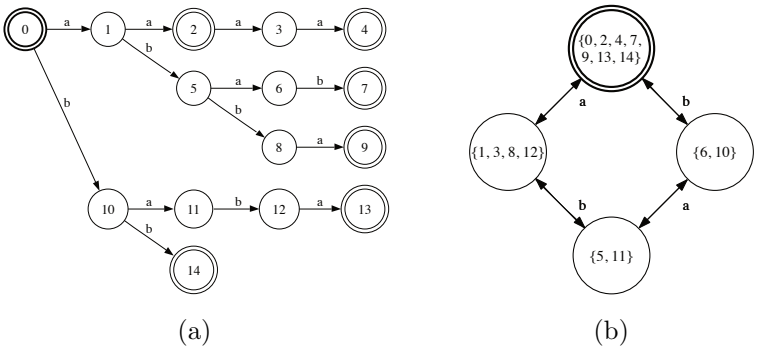


Figure 16.7
Example of inference of a reversible automaton. (a) Prefix-tree $PT(S)$ representing $S = (\epsilon, aa, bb, aaaa, abab, abba, baba)$. (b) Automaton $\widehat{A}$ returned by LearnReversibleAutomata() for the input S. A double-direction arrow represents two transitions with the same label with opposite directions. The language accepted by $\widehat{A}$ is that of strings with an even number of as and bs.

(lines 10–13). Before merging blocks B_1 and B_2 into a new block B' that defines a new partition π (line 15), the successor and predecessor values for the new block B' are defined as follows (line 14). For each symbol $a \in \Sigma$, $succ(B', a) = \emptyset$ if $succ(B_1, a) = succ(B_2, a) = \emptyset$, otherwise $succ(B', a)$ is set to one of $succ(B_1, a)$ if it is non-empty, $succ(B_2, a)$ otherwise. The predecessor values are defined in a similar way. Figure 16.7 illustrates the application of the algorithm in the case of a sample with $m = 7$ strings.

Proposition 16.9 *Let S be a finite set of strings and let $A = PT(S)$ be the prefix-tree automaton defined from S. Then, the final partition defined by* LearnReversibleAutomata() *used with input S is the finest partition π for which A/π is reversible.*

Proof: Let T be the number of iterations of the algorithm for the input sample S. We denote by π_t the partition defined by the algorithm after $t \geq 1$ iterations of the loop, with π_T the final partition.

A/π_T is a reversible automaton since all final states are guaranteed to be merged into the same block as a consequence of the initialization step of line 3 and, for any block B, by definition of the algorithm, states reachable by $a \in \Sigma$ from B are contained in the same block, and similarly for those admitting a transition labeled with a to a state of B.

Let π' be a partition of the states of A for which A/π' is reversible. We show by recurrence that π_T refines π'. Clearly, the trivial partition π_0 refines π'. Assume that π_s refines π' for all $s \leq t$. π_{t+1} is obtained from π by merging two blocks $B(q_1, \pi_t)$ and $B(q_2, \pi_t)$. Since π_t refines π', we must have $B(q_1, \pi_t) \subseteq B(q_1, \pi')$

and $B(q_2, \pi_t) \subseteq B(q_2, \pi')$. To show that π_{t+1} refines π', it suffices to prove that $B(q_1, \pi') = B(q_2, \pi')$.

A reversible automaton has only one final state, therefore, for the partition π', all final states of A must be placed in the same block. Thus, if the pair (q_1, q_2) processed at the $(t+1)$th iteration is a pair of final states placed in LIST at the initialization step (line 3), then we must have $B(q_1, \pi') = B(q_2, \pi')$. Otherwise, (q_1, q_2) was placed in LIST as a pair of successor or predecessor states of two states q_1' and q_2' merged at a previous iteration $s \leq t$. Since π_s refines π', q_1' and q_2' are in the same block of π' and since A/π' is reversible, q_1 and q_2 must also be in the same block as successors or predecessors of the same block for the same label $a \in \Sigma$, thus $B(q_1, \pi') = B(q_2, \pi')$. □

Theorem 16.10 *Let S be a finite set of strings and let A be the automaton returned by* LearnReversibleAutomata() *when used with input S. Then, $L(A)$ is the smallest reversible language containing S.*

Proof: Let L be a reversible language containing S, and let A' be a reversible automaton with $L(A') = L$. Since every string of S is accepted by A', any $u \in \mathrm{Pref}(S)$ can be read from the initial state of A' to reach some state $q(u)$ of A'. Consider the automaton A'' derived from A' by keeping only states of the form $q(u)$ and transitions between such states. A'' has the unique final state of A' since $q(u)$ is final for $u \in S$, and it has the initial state of A', since ϵ is a prefix of strings of S. Furthermore, A'' directly inherits from A' the property of being deterministic and reverse deterministic. Thus, A'' is reversible.

The states of A'' define a partition of $\mathrm{Pref}(S)$: $u, v \in \mathrm{Pref}(S)$ are in the same block iff $q(u) = q(v)$. Since by definition of the prefix-tree $PT(S)$, its states can be identified with $\mathrm{Pref}(S)$, the states of A'' also define a partition π' of the states of $PT(S)$ and thus $A'' = PT(S)/\pi'$. By proposition 16.9, the partition π defined by algorithm LearnReversibleAutomata() run with input S is the finest such that $PT(S)/\pi$ is reversible. Therefore, we must have $L(PT(S)/\pi) \subseteq L(PT(S)/\pi') = L(A'')$. Since A'' is a sub-automaton of A', L contains $L(A'')$ and therefore $L(PT(S)/\pi) = L(A)$, which concludes the proof. □

Theorem 16.11 (Identification in the limit of reversible languages) *Let L be a reversible language, then algorithm* LearnReversibleAutomata() *identifies L in the limit from a positive presentation.*

Proof: Let L be a reversible language. By proposition 16.8, L admits a finite characteristic sample S_L. Let $(x_n)_{n \in \mathbb{N}}$ be a positive presentation of L and let $\mathcal{X}_n$ denote the union of the first n elements of the sequence. Since S_L is finite, there exists $N \geq 1$ such that $S_L \subseteq \mathcal{X}_N$. By theorem 16.10, for any $n \geq N$, LearnReversibleAutomata() run on the finite sample $\mathcal{X}_n$ returns the smallest

reversible language L' containing $\mathcal{X}_n$ a fortiori S_L, which, by definition of S_L, implies that $L' = L$. □

The main operations needed for the implementation of the algorithm for learning reversible automata are the standard FIND and UNION to determine the block a state belongs to and to merge two blocks into a single one. Using a disjoint-set data structure for these operations, the time complexity of the algorithm can be shown to be in $O(n\alpha(n))$, where n denotes the sum of the lengths of all strings in the input sample S and $\alpha(n)$ the inverse of the Ackermann function, which is essentially constant ($\alpha(n) \leq 4$ for $n \leq 10^{80}$).

16.5 Chapter notes

For an overview of finite automata and some related results, see Hopcroft and Ullman [1979] or the more recent Handbook chapter by Perrin [1990], as well as the series of books by M. Lothaire [Lothaire, 1982, 1990, 2005] and the even more recent book by De la Higuera [2010].

Theorem 16.2, stating that the problem of finding a minimum consistent DFA is NP-hard, is due to Gold [1978]. This result was later extended by Angluin [1978]. Pitt and Warmuth [1993] further strengthened these results by showing that even an approximation within a polynomial function of the size of the smallest automaton is NP-hard (theorem 16.3). Their hardness results apply also to the case where prediction is made using NFAs. Kearns and Valiant [1994] presented hardness results of a different nature relying on cryptographic assumptions. Their results imply that no polynomial-time algorithm can learn consistent NFAs polynomial in the size of the smallest DFA from a finite sample of accepted and rejected strings if any of the generally accepted cryptographic assumptions holds: if factoring Blum integers is hard; or if the RSA public key cryptosystem is secure; or if deciding quadratic residuosity is hard. Most recently, Chalermsook et al. [2014] improved the non-approximation guarantee of Pitt and Warmuth [1993] to a tight bound.

On the positive side, Trakhtenbrot and Barzdin [1973] showed that the smallest finite automaton consistent with the input data can be learned exactly from a uniform complete sample, whose size is exponential in the size of the automaton. The worst-case complexity of their algorithm is exponential, but a better average-case complexity can be obtained assuming that the topology and the labeling are selected randomly [Trakhtenbrot and Barzdin, 1973] or even that the topology is selected adversarially [Freund et al., 1993].

Cortes, Kontorovich, and Mohri [2007a] study an approach to the problem of learning automata based on linear separation in some appropriate high-dimensional feature space; see also Kontorovich et al. [2006, 2008]. The mapping of strings to

that feature space can be defined implicitly using the rational kernels presented in chapter 6, which are themselves defined via weighted automata and transducers.

The model of learning with queries was introduced by Angluin [1978], who also proved that finite automata can be learned in time polynomial in the size of the minimal automaton and that of the longest counter-example. Bergadano and Varricchio [1995] further extended this result to the problem of learning weighted automata defined over any field (see also an optimal algorithm by Bisht et al. [2006]). Using the relationship between the size of a minimal weighted automaton over a field and the rank of the corresponding Hankel matrix, the learnability of many other concepts classes such as disjoint DNF can be shown [Beimel et al., 2000]. Our description of an efficient implementation of the algorithm of Angluin [1982] using decision trees is adapted from Kearns and Vazirani [1994].

The model of identification in the limit of automata was introduced and analyzed by Gold [1967]. Deterministic finite automata were shown not to be identifiable in the limit from positive examples [Gold, 1967]. But, positive results were given for the identification in the limit of a number of sub-classes, such as the family of k-reversible languages Angluin [1982] considered in this chapter. Positive results also hold for learning subsequential transducers Oncina et al. [1993]. Some restricted classes of probabilistic automata such as acyclic probabilistic automata were also shown by Ron et al. [1995] to be efficiently learnable.

There is a vast literature dealing with the problem of learning automata. In particular, positive results have been shown for a variety of sub-families of finite automata in the scenario of learning with queries and learning scenarios of different kinds have been introduced and analyzed for this problem. The results presented in this chapter should therefore be viewed only as an introduction to that material.

16.6 Exercises

16.1 Minimal DFA. Show that a minimal DFA A also has the minimal number of transitions among all other DFAs equivalent to A. Prove that a language L is regular iff $Q = \{\text{Suff}_L(u) \colon u \in \Sigma^*\}$ is finite. Show that the number of states of a minimal DFA A with $L(A) = L$ is precisely the cardinality of Q.

16.2 VC-dimension of finite automata.

(a) What is the VC-dimension of the family of all finite automata? What does that imply for PAC-learning of finite automata? Does this result change if we restrict ourselves to learning acyclic automata (automata with no cycles)?

(b) Show that the VC-dimension of the family of DFAs with at most n states is bounded by $O(|\Sigma| n \log n)$.

16.3 PAC learning with membership queries. Give an example of a concept class $\mathcal{C}$ that is efficiently PAC-learnable with membership queries but that is not efficiently exactly learnable.

16.4 Learning monotone DNF formulae with queries. Show that the class of monotone DNF formulae over n variables is efficiently exactly learnable using membership and equivalence queries. (*Hint*: a *prime implicant* t of a formula f is a product of literals such that t implies f but no proper sub-term of t implies f. Use the fact that for monotone DNF, the number of prime implicants is at the most the number of terms of the formula.)

16.5 Learning with unreliable query responses. Consider the problem where the learner must find an integer x selected by the oracle within $[n]$, where $n \geq 1$ is given. To do so, the learner can ask questions of the form $(x \leq m?)$ or $(x > m?)$ for $m \in [n]$. The oracle responds to these questions but may give an incorrect response to k questions. How many questions should the learner ask to determine x? (*Hint*: observe that the learner can repeat each question $2k+1$ times and use the majority vote.)

16.6 Algorithm for learning reversible languages. What is the DFA A returned by the algorithm for learning reversible languages when applied to the sample $S = \{ab, aaabb, aabbb, aabbbb\}$? Suppose we add a new string to the sample, say $x = abab$. How should A be updated to compute the result of the algorithm for $S \cup \{x\}$? More generally, describe a method for updating the result of the algorithm incrementally.

16.7 k-reversible languages. A finite automaton A' is said to be *k-deterministic* if it is deterministic modulo a lookahead k: if two distinct states p and q are both initial, or are both reached from another state r by reading $a \in \Sigma$, then no string u of length k can be read in A' both from p and q. A finite automaton A is said to be *k-reversible* if it is deterministic and if A^R is k-deterministic. A language L is *k-reversible* if it is accepted by some k-reversible automaton.

(a) Prove that L is k-reversible iff for any strings $u, u', v \in \Sigma^*$ with $|v| = k$,

$$\mathrm{Suff}_L(uv) \cap \mathrm{Suff}_L(u'v) \neq \emptyset \implies \mathrm{Suff}_L(uv) = \mathrm{Suff}_L(u'v).$$

(b) Show that a k-reversible language admits a characteristic language.

(c) Show that the following defines an algorithm for learning k-reversible automata. Proceed as in the algorithm for learning reversible automata but with the following merging rule instead: merge blocks B_1 and B_2 if they can be reached by the same string u of length k from some other block and if B_1 and B_2 are both final or have a common successor.

17 Reinforcement Learning

This chapter presents an introduction to reinforcement learning, a rich area of machine learning with connections to control theory, optimization, and cognitive sciences. Reinforcement learning is the study of planning and learning in a scenario where a learner actively interacts with the environment to achieve a certain goal. This active interaction justifies the terminology of *agent* used to refer to the learner. The achievement of the agent's goal is typically measured by the reward it receives from the environment and which it seeks to maximize.

We first introduce the general scenario of reinforcement learning and then introduce the model of Markov decision processes (MDPs), which is widely adopted in this area, as well as essential concepts such as that of *policy* or *policy value* related to this model. The rest of the chapter presents several algorithms for the *planning* problem, which corresponds to the case where the environment model is known to the agent, and then a series of *learning* algorithms for the more general case of an unknown model.

17.1 Learning scenario

The general scenario of reinforcement learning is illustrated by figure 17.1. Unlike the supervised learning scenario considered in previous chapters, here, the learner does not passively receive a labeled data set. Instead, it collects information through a course of *actions* by interacting with the *environment*. In response to an action, the learner or *agent*, receives two types of information: its current *state* in the environment, and a real-valued *reward*, which is specific to the task and its corresponding goal.

The objective of the agent is to maximize its reward and thus to determine the best course of actions, or *policy*, to achieve that objective. However, the information he receives from the environment is only the immediate reward related to the action just taken. No future or long-term reward feedback is provided by the environment. An important aspect of reinforcement learning is to consider delayed rewards or

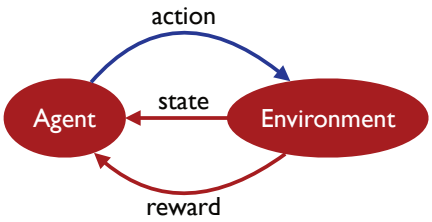


Figure 17.1
Representation of the general scenario of reinforcement learning.

penalties. The agent is faced with the dilemma between exploring unknown states and actions to gain more information about the environment and the rewards, and exploiting the information already collected to optimize its reward. This is known as the *exploration versus exploitation trade-off* inherent to reinforcement learning.

Note that there are several differences between the learning scenario of reinforcement learning and that of supervised learning examined in most of the previous chapters. Unlike supervised learning, in reinforcement learning there is no fixed distribution according to which instances are drawn; it is the choice of a policy that defines the distribution over observations. In fact, slight changes to the policy may have dramatic effects on the rewards received. Furthermore, in general, the environment may not be fixed and could vary as a result of the actions selected by the agent. This may be a more realistic model for some learning problems than the standard supervised learning. Finally, note that, unlike supervised learning, in reinforcement learning, training and testing phases are intermixed.

Two main settings can be distinguished here: the one where the environment model is known to the agent, in which case its objective of maximizing the reward received is reduced to a *planning problem*; and the one where the environment model is unknown, in which case the agent faces a *learning problem*. In the latter setting, the agent must learn from the state and reward information gathered to both gain information about the environment and determine the best action policy. This chapter presents algorithmic solutions for both of these settings.

17.2 Markov decision process model

We first introduce the model of Markov decision processes (MDPs), a model of the environment and interactions with the environment widely adopted in reinforcement learning. An MDP is a Markovian process defined as follows.

Definition 17.1 (MDPs) *A Markov decision process (MDP) is defined by:*

- *a set of* states S*, possibly infinite.*

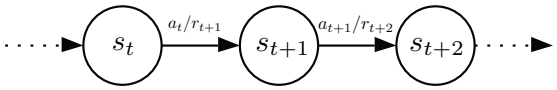


Figure 17.2
Illustration of the states and transitions of an MDP at different times.

- *a* start state *or* initial state $s_0 \in S$.
- *a set of* actions A, *possibly infinite.*
- *a* transition probability $\mathbb{P}[s'|s,a]$*: distribution over destination states* $s' = \delta(s,a)$.
- *a* reward probability $\mathbb{P}[r'|s,a]$*: distribution over rewards returned* $r' = r(s,a)$.

The model is Markovian because the transition and reward probabilities depend only on the current state s and not the entire history of states and actions taken. This definition of MDP can be further generalized to the case of non-discrete state and action sets.

In a discrete-time model, actions are taken at a set of *decision epochs* $\{0, \ldots, T\}$, and this is the model we will adopt in what follows. This model can also be straightforwardly generalized to a continuous-time one where actions are taken at arbitrary points in time.

When T is finite, the MDP is said to have a *finite horizon*. Independently of the finiteness of the time horizon, an MDP is said to be *finite* when both S and A are finite sets. Here, we are considering the general case where the reward $r(s,a)$ at state s when taking action a is a random variable. However, in many cases, the reward is assumed to be a deterministic function the state and action pair (s,a).

Figure 17.2 illustrates the model corresponding to an MDP. At time $t \in \{0, \ldots, T\}$ the state observed by the agent is s_t and it takes action $a_t \in A$. The state reached is s_{t+1} (with probability $\mathbb{P}[s_{t+1}|s_t, a_t]$) and the reward received $r_{t+1} \in \mathbb{R}$ (with probability $\mathbb{P}[r_{t+1}|s_t, a_t]$).

Many real-world tasks can be represented by MDPs. Figure 17.3 gives the example of a simple MDP for a robot picking up balls on a tennis court.

17.3 Policy

The main problem for an agent in an MDP environment is to determine the action to take at each state, that is, an action *policy*.

17.3.1 Definition

Definition 17.2 (Policy) *A* policy *is a mapping* $\pi\colon S \to \Delta(A)$*, where* $\Delta(A)$ *is the set of probability distributions over* A*. A policy* π *is* deterministic *if for any* s*, there exists a unique* $a \in A$ *such that* $\pi(s)(a) = 1$*. In that case, we can identify* π *with a mapping from* S *to* A *and use* $\pi(s)$ *to denote that action.*

More precisely, this is the definition of a *stationary policy* since the choice of the distribution of actions does not depend on time. More generally, we could define a *non-stationary policy* as a sequence of mappings $\pi_t \colon S \to \Delta(A)$ indexed by t. In particular, in the finite horizon case, a non-stationary policy is typically necessary for optimizing rewards.

The agent's objective is to find a policy that maximizes its expected (reward) *return*. The return it receives following a deterministic policy π along a specific sequence of states $s_0, \ldots, s_T$ is defined as follows:

- for a finite horizon ($T < \infty$): $\sum_{t=0}^{T} r\big(s_t, \pi(s_t)\big)$.
- for an infinite horizon ($T = \infty$): $\sum_{t=0}^{+\infty} \gamma^t r\big(s_t, \pi(s_t)\big)$, where $\gamma \in [0, 1)$ is a constant factor less than one used to discount future rewards.

Note that the return is a single scalar summarizing a possibly infinite sequence of immediate rewards. In the discounted case, early rewards are viewed as more valuable than later ones.

17.3.2 Policy value

This leads to the following definition of the value of a policy at each state.

Definition 17.3 (Policy value) *The* value $V_\pi(s)$ *of a policy* π *at state* $s \in S$ *is defined as the expected reward returned when starting at* s *and following policy* π*:*

- *finite horizon:* $V_\pi(s) = \mathbb{E}_{a_t \sim \pi(s_t)} \left[\sum_{t=0}^{T} r\big(s_t, a_t\big) \;\middle|\; s_0 = s \right]$;
- *infinite discounted horizon:* $V_\pi(s) = \mathbb{E}_{a_t \sim \pi(s_t)} \left[\sum_{t=0}^{+\infty} \gamma^t r\big(s_t, a_t\big) \;\middle|\; s_0 = s \right]$,

where the expectations are over the random selection of an action a_t *according to the distribution* $\pi(s_t)$*, which is explicitly indicated, and over the random states* s_t *reached and the reward values* $r\big(s_t, a_t\big)$.[22] *An infinite undiscounted horizon is also often considered based on the limit of the average reward, when it exists.*

17.3.3 Optimal policies

Starting from a state $s \in S$, to maximize its reward, an agent naturally seeks a policy π with the largest value $V_\pi(s)$. In this section, we will show that, remarkably, for any finite MDP in the infinite horizon setting, there exists a policy that is *optimal* for *any* start state, that is one with the following definition.

Definition 17.4 (Optimal policy) *A policy* π^* *is* optimal *if its value is maximal for every state* $s \in S$*, that is, for any policy* π *and any state* $s \in S$*,* $V_{\pi^*}(s) \geq V_\pi(s)$.

[22] More generally, in all that follows, the randomization with respect to the reward function and the next state will not be explicitly indicated to simplify the notation.

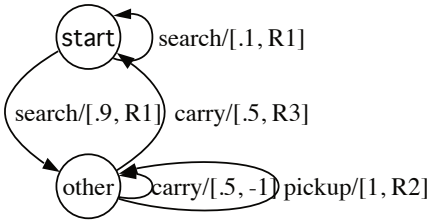


Figure 17.3
Example of a simple MDP for a robot picking up balls on a tennis court. The set of actions is $A = \{search, carry, pickup\}$ and the set of states reduced to $S = \{start, other\}$. Each transition is labeled with the action followed by the probability of the transition probability and the reward received after taking that action. R_1, R_2, and R_3 are real numbers indicating the reward associated to each transition (case of deterministic reward).

Moreover, we will show that for any MDP there exists a deterministic optimal policy. To do so, it is convenient to introduce the notion of *state-action value function*.

Definition 17.5 (State-action value function) *The* state-action value function Q *associated to a policy* π *is defined for all* $(s, a) \in S \times A$ *as the expected return for taking action* $a \in A$ *at state* $s \in S$ *and then following policy* π*:*

$$\begin{aligned} Q_\pi(s,a) &= \mathbb{E}[r(s,a)] + \mathop{\mathbb{E}}_{a_t \sim \pi(s_t)} \left[\sum_{t=1}^{+\infty} \gamma^t r\big(s_t, a_t\big) \;\middle|\; s_0 = s, a_0 = a \right] \\ &= \mathbb{E}\left[r(s,a) + \gamma V_\pi(s_1) \;\middle|\; s_0 = s, a_0 = a \right]. \end{aligned} \tag{17.1}$$

Observe that $\mathbb{E}_{a \sim \pi(s)} \big[Q_\pi(s,a) \big] = V_\pi(s)$ (see also proposition 17.9)

Theorem 17.6 (Policy improvement theorem) *For any two policies* π *and* π' *the following holds:*

$$\Big(\forall s \in S, \mathop{\mathbb{E}}_{a \sim \pi'(s)} \big[Q_\pi(s,a) \big] \geq \mathop{\mathbb{E}}_{a \sim \pi(s)} \big[Q_\pi(s,a) \big] \Big) \Rightarrow \Big(\forall s \in S, V_{\pi'}(s) \geq V_\pi(s) \Big).$$

Furthermore, a strict inequality for at least one state s *in the left-hand side implies a strict inequality for at least one* s *in the right-hand side.*

Proof: Assume that π and π' verify the left-hand side. For any $s \in S$, we have

$$\begin{aligned}
V_\pi(s) &= \mathop{\mathbb{E}}_{a\sim\pi(s)} \big[Q_\pi(s,a)\big] \\
&\leq \mathop{\mathbb{E}}_{a\sim\pi'(s)} \big[Q_\pi(s,a)\big] \\
&= \mathop{\mathbb{E}}_{a\sim\pi'(s)} \Big[r(s,a) + \gamma V_\pi(s_1) \;\Big|\; s_0 = s\Big] \\
&= \mathop{\mathbb{E}}_{a\sim\pi'(s)} \Big[r(s,a) + \gamma \mathop{\mathbb{E}}_{a_1\sim\pi(s_1)} \big[Q_\pi(s_1,a_1)\big] \;\Big|\; s_0 = s\Big] \\
&\leq \mathop{\mathbb{E}}_{a\sim\pi'(s)} \Big[r(s,a) + \gamma \mathop{\mathbb{E}}_{a_1\sim\pi'(s_1)} \big[Q_\pi(s_1,a_1)\big] \;\Big|\; s_0 = s\Big] \\
&= \mathop{\mathbb{E}}_{\substack{a\sim\pi'(s)\\ a_1\sim\pi'(s_1)}} \Big[r(s,a) + \gamma r(s_1,a_1) + \gamma^2 V_\pi(s_2) \;\Big|\; s_0 = s\Big].
\end{aligned}$$

Proceeding in this way shows that for any $T \geq 1$:

$$V_\pi(s) \leq \mathop{\mathbb{E}}_{a_t\sim\pi'(s_t)} \left[\sum_{t=0}^{T} \gamma^t \,\mathbb{E}[r(s_t,a_t)] + \gamma^{T+1} V_\pi(s_{T+1}) \;\middle|\; s_0 = s\right].$$

Since $V_\pi(s_{T+1})$ is bounded, taking the limit $T \to +\infty$ gives

$$V_\pi(s) \leq \mathop{\mathbb{E}}_{a_t\sim\pi'(s_t)} \left[\sum_{t=0}^{+\infty} \gamma^t \,\mathbb{E}[r(s_t,a_t)] \;\middle|\; s_0 = s\right] = V_{\pi'}(s).$$

Finally, any strict inequality in the left-hand side property results in a strict inequality in the chain of inequalities above. □

Theorem 17.7 (Bellman's optimality condition) *A policy π is optimal iff for any pair $(s,a) \in S \times A$ with $\pi(s)(a) > 0$ the following holds:*

$$a \in \operatorname*{argmax}_{a' \in A} Q_\pi(s,a'). \tag{17.2}$$

Proof: By Theorem 17.6, if the condition (17.2) does not hold for some (s,a) with $\pi(s)(a) > 0$, then the policy π is not optimal. This is because π can then be improved by defining π' such that $\pi'(s') = \pi(s)$ for $s' \neq s$ and $\pi'(s)$ concentrated on any element of $\operatorname{argmax}_{a'\in A} Q_\pi(s,a')$. π' verifies $\mathbb{E}_{a\sim\pi'(s)}\big[Q_\pi(s',a)\big] = \mathbb{E}_{a\sim\pi(s)}\big[Q_\pi(s',a)\big]$ for $s' \neq s$ and $\mathbb{E}_{a\sim\pi'(s)}\big[Q_\pi(s,a)\big] > \mathbb{E}_{a\sim\pi(s)}\big[Q_\pi(s,a)\big]$. Thus, by Theorem 17.6, $V_{\pi'}(s) > V_\pi(s)$ for at least one s and π is not optimal.

Conversely, let π' be a non-optimal policy. Then there exists a policy π and at least one state s for which $V_{\pi'}(s) < V_\pi(s)$. By Theorem 17.6, this implies that there exists some state $s \in S$ with $\mathbb{E}_{a\sim\pi'(s)}\big[Q_\pi(s,a)\big] < \mathbb{E}_{a\sim\pi(s)}\big[Q_\pi(s,a)\big]$. Thus, π' cannot satisfy the condition (17.2). □

Theorem 17.8 (Existence of an optimal deterministic policy) *Any finite MDP admits an optimal deterministic policy.*

Proof: Let π^* be a deterministic policy maximizing $\sum_{s\in S} V_\pi(s)$. π^* exists since there are only finitely many deterministic policies. If π^* were not optimal, by Theorem 17.7, there would exist a state s with $\pi(s) \notin \operatorname{argmax}_{a'\in A} Q_\pi(s,a')$. By theorem 17.6, π^* could then be improved by choosing a policy π with $\pi(s) \in \operatorname{argmax}_{a'\in A} Q_\pi(s,a')$ and π coinciding with π^* for all other states. But then π would verify $V_{\pi*}(s) \leq V_\pi(s)$ with a strict inequality at least for one state. This would contradict the fact that π^* maximizes $\sum_{s\in S} V_\pi(s)$. □

In view of the existence of a deterministic optimal policy, in what follows, to simplify the discussion, we will consider only deterministic policies. Let π^* denote a (deterministic) optimal policy, and let Q^* and V^* denote its corresponding state-action value function and value function. By Theorem 17.7, we can write

$$\forall s \in S,\ \pi^*(s) = \operatorname*{argmax}_{a\in A} Q^*(s,a). \tag{17.3}$$

Thus, the knowledge of the state-action value function Q^* is sufficient for the agent to determine the optimal policy, without any direct knowledge of the reward or transition probabilities. Replacing Q^* by its definition gives the following system of equations for the optimal policy values $V^*(s) = Q^*(s,\pi^*(s))$:

$$\forall s \in S,\ V^*(s) = \max_{a\in A} \Big\{ \mathbb{E}[r(s,a)] + \gamma \sum_{s'\in S} \mathbb{P}[s'|s,a] V^*(s') \Big\}, \tag{17.4}$$

also known as *Bellman equations*. Note that this system of equations is not linear due to the presence of the max operator.

17.3.4 Policy evaluation

The value of a policy at state s can be expressed in terms of its values at other states, forming a system of linear equations.

Proposition 17.9 (*Bellman equations***)** *The values $V_\pi(s)$ of policy π at states $s \in S$ for an infinite horizon MDP obey the following system of linear equations:*

$$\forall s \in S,\ V_\pi(s) = \mathop{\mathbb{E}}_{a_1\sim\pi(s)}[r(s,a_1)] + \gamma \sum_{s'} \mathbb{P}[s'|s,\pi(s)] V_\pi(s'). \tag{17.5}$$

Proof: We can decompose the expression of the policy value as a sum of the first term and the rest of the terms, which admit γ as a multiplier:

$$\begin{aligned} V_\pi(s) &= \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t r\big(s_t, \pi(s_t)\big) \;\middle|\; s_0 = s\right]. \\ &= \mathbb{E}[r(s,\pi(s))] + \gamma\, \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t r\big(s_{t+1}, \pi(s_{t+1})\big) \;\middle|\; s_0 = s\right] \\ &= \mathbb{E}[r(s,\pi(s))] + \gamma\, \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t r\big(s_{t+1}, \pi(s_{t+1})\big) \;\middle|\; s_1 = \delta(s,\pi(s))\right] \\ &= \mathbb{E}[r(s,\pi(s)] + \gamma\, \mathbb{E}[V_\pi(\delta(s,\pi(s)))]. \end{aligned}$$

This completes the proof. □

This a linear system of equations, also known as Bellman equations, that is distinct from the non-linear system (17.4). The system can be rewritten as

$$\mathbf{V} = \mathbf{R} + \gamma \mathbf{P}\mathbf{V}, \tag{17.6}$$

using the following notation: $\mathbf{P}$ denotes the transition probability matrix defined by $\mathbf{P}_{s,s'} = \mathbb{P}[s'|s,\pi(s)]$ for all $s, s' \in S$; $\mathbf{V}$ is the value column matrix whose sth component is $\mathbf{V}_s = V_\pi(s)$; and $\mathbf{R}$ the reward column matrix whose sth component is $\mathbf{R}_s = \mathbb{E}[r(s,\pi(s)]$. $\mathbf{V}$ is typically the unknown variable in the Bellman equations and is determined by solving for it.

The following theorem shows that, for a finite MDP, this system of linear equations admits a unique solution.

Theorem 17.10 *For a finite MDP, Bellman's equations admit a unique solution given by*

$$\mathbf{V}_0 = (\mathbf{I} - \gamma \mathbf{P})^{-1}\mathbf{R}. \tag{17.7}$$

Proof: The Bellman equations (17.6) can be equivalently written as

$$(\mathbf{I} - \gamma \mathbf{P})\mathbf{V} = \mathbf{R}.$$

Thus, to prove the theorem it suffices to show that $(\mathbf{I} - \gamma\mathbf{P})$ is invertible. To do so, note that the infinity of $\mathbf{P}$ can be computed using its stochasticity properties:

$$\|\mathbf{P}\|_\infty = \max_s \sum_{s'} |\mathbf{P}_{ss'}| = \max_s \sum_{s'} \mathbb{P}[s'|s,\pi(s)] = 1.$$

This implies that $\|\gamma\mathbf{P}\|_\infty = \gamma < 1$. The eigenvalues of $\gamma\mathbf{P}$ are thus all less than one, and $(\mathbf{I} - \gamma\mathbf{P})$ is invertible. □

Thus, for a finite MDP, when the transition probability matrix $\mathbf{P}$ and the reward expectations $\mathbf{R}$ are known, the value of policy π at all states can be determined by inverting a matrix.

17.4 Planning algorithms

In this section, we assume that the environment model is known. That is, the transition probability $\mathbb{P}[s'|s,a]$ and the expected reward $\mathbb{E}[r(s,a)]$ for all $s, s' \in S$ and $a \in A$ are assumed to be given. The problem of finding the optimal policy then does not require learning the parameters of the environment model or estimating other quantities helpful in determining the best course of actions, it is purely a *planning* problem.

This section discusses three algorithms for this planning problem: the value iteration algorithm, the policy iteration algorithm, and a linear programming formulation of the problem.

17.4.1 Value iteration

The *value iteration algorithm* seeks to determine the optimal policy values $V^*(s)$ at each state $s \in S$, and thereby the optimal policy. The algorithm is based on the Bellman equations (17.4). As already indicated, these equations do not form a system of linear equations and require a different technique to determine the solution. The main idea behind the design of the algorithm is to use an iterative method to solve them: the new values of $V(s)$ are determined using the Bellman equations and the current values. This process is repeated until a convergence condition is met.

For a vector $\mathbf{V}$ in $\mathbb{R}^{|S|}$, we denote by $V(s)$ its sth coordinate, for any $s \in S$. Let $\mathbf{\Phi}\colon \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ be the mapping defined based on Bellman's equations (17.4):

$$\forall s \in S, [\mathbf{\Phi}(\mathbf{V})](s) = \max_{a \in A} \Big\{ \mathbb{E}[r(s,a)] + \gamma \sum_{s' \in S} \mathbb{P}[s'|s,a] V(s') \Big\}. \tag{17.8}$$

The maximizing actions $a \in A$ in these equations define an action to take at each state $s \in S$, that is a policy π. We can thus rewrite these equations in matrix terms as follows:

$$\mathbf{\Phi}(\mathbf{V}) = \max_{\pi} \{ \mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V} \}, \tag{17.9}$$

where $\mathbf{P}_\pi$ is the transition probability matrix defined by $(\mathbf{P}_\pi)_{ss'} = \mathbb{P}[s'|s,\pi(s)]$ for all $s, s' \in S$, and $\mathbf{R}_\pi$ the reward vector defined by $(\mathbf{R}_\pi)_s = \mathbb{E}[r(s,\pi(s)]$, for all $s \in S$.

The algorithm is directly based on (17.9). The pseudocode is given above. Starting from an arbitrary policy value vector $\mathbf{V}_0 \in \mathbb{R}^{|S|}$, the algorithm iteratively applies

VALUEITERATION($\mathbf{V}_0$)

1 $\mathbf{V} \leftarrow \mathbf{V}_0$ ▷ $\mathbf{V}_0$ arbitrary value
2 **while** $\|\mathbf{V} - \mathbf{\Phi}(\mathbf{V})\| \geq \frac{(1-\gamma)\epsilon}{\gamma}$ **do**
3 $\mathbf{V} \leftarrow \mathbf{\Phi}(\mathbf{V})$
4 **return** $\mathbf{\Phi}(\mathbf{V})$

Figure 17.4
Value iteration algorithm.

$\mathbf{\Phi}$ to the current $\mathbf{V}$ to obtain a new policy value vector until $\|\mathbf{V} - \mathbf{\Phi}(\mathbf{V})\| < \frac{(1-\gamma)\epsilon}{\gamma}$, where $\epsilon > 0$ is a desired approximation. The following theorem proves the convergence of the algorithm to the optimal policy values.

Theorem 17.11 *For any initial value $\mathbf{V}_0$, the sequence defined by $\mathbf{V}_{n+1} = \mathbf{\Phi}(\mathbf{V}_n)$ converges to $\mathbf{V}^*$.*

Proof: We first show that $\mathbf{\Phi}$ is γ-Lipschitz for the $\|\cdot\|_\infty$.[23] For any $s \in S$ and $\mathbf{V} \in \mathbb{R}^{|S|}$, let $a^*(s)$ be the maximizing action defining $\mathbf{\Phi}(\mathbf{V})(s)$ in (17.8). Then, for any $s \in S$ and any $\mathbf{U} \in \mathbb{R}^{|S|}$,

$$\begin{aligned}\mathbf{\Phi}(\mathbf{V})(s) - \mathbf{\Phi}(\mathbf{U})(s) &\leq \mathbf{\Phi}(\mathbf{V})(s) - \Big(\mathbb{E}[r(s, a^*(s))] + \gamma \sum_{s' \in S} \mathbb{P}[s' \mid s, a^*(s)]\mathbf{U}(s')\Big) \\ &= \gamma \sum_{s' \in S} \mathbb{P}[s'|s, a^*(s)][\mathbf{V}(s') - \mathbf{U}(s')] \\ &\leq \gamma \sum_{s' \in S} \mathbb{P}[s'|s, a^*(s)]\|\mathbf{V} - \mathbf{U}\|_\infty = \gamma\|\mathbf{V} - \mathbf{U}\|_\infty.\end{aligned}$$

Proceeding similarly with $\mathbf{\Phi}(\mathbf{U})(s) - \mathbf{\Phi}(\mathbf{V})(s)$, we obtain $\mathbf{\Phi}(\mathbf{U})(s) - \mathbf{\Phi}(\mathbf{V})(s) \leq \gamma\|\mathbf{V} - \mathbf{U}\|_\infty$. Thus, $|\mathbf{\Phi}(\mathbf{V})(s) - \mathbf{\Phi}(\mathbf{U})(s)| \leq \gamma\|\mathbf{V} - \mathbf{U}\|_\infty$ for all s, which implies

$$\|\mathbf{\Phi}(\mathbf{V}) - \mathbf{\Phi}(\mathbf{U})\|_\infty \leq \gamma\|\mathbf{V} - \mathbf{U}\|_\infty,$$

[23] A β-Lipschitz function with $\beta < 1$ is also called β-*contracting*. In a *complete metric space*, that is a metric space where any Cauchy sequence converges to a point of that space, a β-contracting function f admits a *fixed point*: any sequence $(f(x_n))_{n\in\mathbb{N}}$ converges to some x with $f(x) = x$. $\mathbb{R}^N$, $N \geq 1$, or, more generally, any finite-dimensional vector space, is a complete metric space.

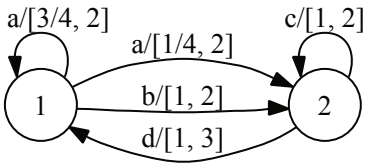


Figure 17.5
Example of MDP with two states. The state set is reduced to $S = \{1, 2\}$ and the action set to $A = \{a, b, c, d\}$. Only transitions with non-zero probabilities are represented. Each transition is labeled with the action taken followed by a pair $[p, r]$ after a slash separator, where p is the probability of the transition and r the expected reward for taking that transition.

that is the γ-Lipschitz property of $\mathbf{\Phi}$. Now, by Bellman equations (17.4), $\mathbf{V}^* = \mathbf{\Phi}(\mathbf{V}^*)$, thus for any $n \in \mathbb{N}$,

$$\|\mathbf{V}^* - \mathbf{V}_{n+1}\|_\infty = \|\mathbf{\Phi}(\mathbf{V}^*) - \mathbf{\Phi}(\mathbf{V}_n)\|_\infty \leq \gamma \|\mathbf{V}^* - \mathbf{V}_n\|_\infty \leq \gamma^{n+1} \|\mathbf{V}^* - \mathbf{V}_0\|_\infty,$$

which proves the convergence of the sequence to $\mathbf{V}^*$ since $\gamma \in (0, 1)$. □

The ϵ-optimality of the value returned by the algorithm can be shown as follows. By the triangle inequality and the γ-Lipschitz property of $\mathbf{\Phi}$, for any $n \in \mathbb{N}$,

$$\begin{aligned}\|\mathbf{V}^* - \mathbf{V}_{n+1}\|_\infty &\leq \|\mathbf{V}^* - \mathbf{\Phi}(\mathbf{V}_{n+1})\|_\infty + \|\mathbf{\Phi}(\mathbf{V}_{n+1}) - \mathbf{V}_{n+1}\|_\infty \\ &= \|\mathbf{\Phi}(\mathbf{V}^*) - \mathbf{\Phi}(\mathbf{V}_{n+1})\|_\infty + \|\mathbf{\Phi}(\mathbf{V}_{n+1}) - \mathbf{\Phi}(\mathbf{V}_n)\|_\infty \\ &\leq \gamma \|\mathbf{V}^* - \mathbf{V}_{n+1}\|_\infty + \gamma \|\mathbf{V}_{n+1} - \mathbf{V}_n\|_\infty.\end{aligned}$$

Thus, if $\mathbf{V}_{n+1}$ is the policy value returned by the algorithm, we have

$$\|\mathbf{V}^* - \mathbf{V}_{n+1}\|_\infty \leq \frac{\gamma}{1-\gamma} \|\mathbf{V}_{n+1} - \mathbf{V}_n\|_\infty \leq \epsilon.$$

The convergence of the algorithm is in $O(\log \frac{1}{\epsilon})$ number of iterations. Indeed, observe that

$$\|\mathbf{V}_{n+1} - \mathbf{V}_n\|_\infty = \|\mathbf{\Phi}(\mathbf{V}_n) - \mathbf{\Phi}(\mathbf{V}_{n-1})\|_\infty \leq \gamma \|\mathbf{V}_n - \mathbf{V}_{n-1}\|_\infty \leq \gamma^n \|\mathbf{\Phi}(\mathbf{V}_0) - \mathbf{V}_0\|_\infty.$$

Thus, if n is the largest integer such that $\frac{(1-\gamma)\epsilon}{\gamma} \leq \|\mathbf{V}_{n+1} - \mathbf{V}_n\|_\infty$, it must verify $\frac{(1-\gamma)\epsilon}{\gamma} \leq \gamma^n \|\mathbf{\Phi}(\mathbf{V}_0) - \mathbf{V}_0\|_\infty$ and therefore $n \leq O\big(\log \frac{1}{\epsilon}\big)$.[24]

[24] Here, the O-notation hides the dependency on the discount factor γ. As a function of γ, the running time is not polynomial.

PolicyIteration(π_0)

1 $\pi \leftarrow \pi_0$ ▷ π_0 arbitrary policy
2 $\pi' \leftarrow$ NIL
3 **while** $(\pi \neq \pi')$ **do**
4 $\quad \mathbf{V} \leftarrow \mathbf{V}_\pi$ ▷ policy evaluation: solve $(\mathbf{I} - \gamma \mathbf{P}_\pi)\mathbf{V} = \mathbf{R}_\pi$.
5 $\quad \pi' \leftarrow \pi$
6 $\quad \pi \leftarrow \text{argmax}_\pi\{\mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V}\}$ ▷ greedy policy improvement.
7 **return** π

Figure 17.6
Policy iteration algorithm.

Figure 17.5 shows a simple example of MDP with two states. The iterated values of these states calculated by the algorithm for that MDP are given by

$$\mathbf{V}_{n+1}(1) = \max\left\{2 + \gamma\Big(\frac{3}{4}\mathbf{V}_n(1) + \frac{1}{4}\mathbf{V}_n(2)\Big), 2 + \gamma \mathbf{V}_n(2)\right\}$$
$$\mathbf{V}_{n+1}(2) = \max\left\{3 + \gamma \mathbf{V}_n(1), 2 + \gamma \mathbf{V}_n(2)\right\}.$$

For $\mathbf{V}_0(1) = -1$, $\mathbf{V}_0(2) = 1$, and $\gamma = 1/2$, we obtain $\mathbf{V}_1(1) = \mathbf{V}_1(2) = 5/2$. Thus, both states seem to have the same policy value initially. However, by the fifth iteration, $\mathbf{V}_5(1) = 4.53125$, $\mathbf{V}_5(2) = 5.15625$ and the algorithm quickly converges to the optimal values $\mathbf{V}^*(1) = 14/3$ and $\mathbf{V}^*(2) = 16/3$ showing that state 2 has a higher optimal value.

17.4.2 Policy iteration

An alternative algorithm for determining the best policy consists of using policy evaluations, which can be achieved via a matrix inversion, as shown by theorem 17.10. The pseudocode of the algorithm known as *policy iteration algorithm* is given in figure 17.6. Starting with an arbitrary action policy π_0, the algorithm repeatedly computes the value of the current policy π via that matrix inversion and greedily selects the new policy as the one maximizing the right-hand side of the Bellman equations (17.9).

The following theorem proves the convergence of the policy iteration algorithm.

Theorem 17.12 *Let $(\mathbf{V}_n)_{n\in\mathbb{N}}$ be the sequence of policy values computed by the algorithm, then, for any $n \in \mathbb{N}$, the following inequalities hold:*

$$\mathbf{V}_n \leq \mathbf{V}_{n+1} \leq \mathbf{V}^*. \tag{17.10}$$

Proof: Let π_{n+1} be the policy improvement at the nth iteration of the algorithm. We first show that $(\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}$ preserves ordering, that is, for any column matrices $\mathbf{X}$ and $\mathbf{Y}$ in $\mathbb{R}^{|S|}$, if $(\mathbf{Y} - \mathbf{X}) \geq \mathbf{0}$, then $(\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}(\mathbf{Y} - \mathbf{X}) \geq \mathbf{0}$. As shown in the proof of theorem 17.10, $\|\gamma\mathbf{P}\|_\infty = \gamma < 1$. Since the radius of convergence of the power series $(1-x)^{-1}$ is one, we can use its expansion and write

$$(\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1} = \sum_{k=0}^{\infty}(\gamma\mathbf{P}_{\pi_{n+1}})^k.$$

Thus, if $\mathbf{Z} = (\mathbf{Y} - \mathbf{X}) \geq \mathbf{0}$, then $(\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}\mathbf{Z} = \sum_{k=0}^{\infty}(\gamma\mathbf{P}_{\pi_{n+1}})^k\mathbf{Z} \geq \mathbf{0}$, since the entries of matrix $\mathbf{P}_{\pi_{n+1}}$ and its powers are all non-negative as well as those of $\mathbf{Z}$.

Now, by definition of π_{n+1}, we have

$$\mathbf{R}_{\pi_{n+1}} + \gamma\mathbf{P}_{\pi_{n+1}}\mathbf{V}_n \geq \mathbf{R}_{\pi_n} + \gamma\mathbf{P}_{\pi_n}\mathbf{V}_n = \mathbf{V}_n,$$

which shows that $\mathbf{R}_{\pi_{n+1}} \geq (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})\mathbf{V}_n$. Since $(\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}$ preserves ordering, this implies that $\mathbf{V}_{n+1} = (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}\mathbf{R}_{\pi_{n+1}} \geq \mathbf{V}_n$, which concludes the proof of the theorem. □

Note that two consecutive policy values can be equal only at the last iteration of the algorithm. The total number of possible policies is $|A|^{|S|}$, thus this constitutes a straightforward upper bound on the maximal number of iterations. Better upper bounds of the form $O\big(\frac{|A|^{|S|}}{|S|}\big)$ are known for this algorithm.

For the simple MDP shown by figure 17.5, let the initial policy π_0 be defined by $\pi_0(1) = b, \pi_0(2) = c$. Then, the system of linear equations for evaluating this policy is

$$\begin{cases} V_{\pi_0}(1) = 1 + \gamma V_{\pi_0}(2) \\ V_{\pi_0}(2) = 2 + \gamma V_{\pi_0}(2), \end{cases}$$

which gives $V_{\pi_0}(1) = \frac{1+\gamma}{1-\gamma}$ and $V_{\pi_0}(2) = \frac{2}{1-\gamma}$.

Theorem 17.13 *Let $(\mathbf{U}_n)_{n\in\mathbb{N}}$ be the sequence of policy values generated by the value iteration algorithm, and $(\mathbf{V}_n)_{n\in\mathbb{N}}$ the one generated by the policy iteration algorithm. If $\mathbf{U}_0 = \mathbf{V}_0$, then,*

$$\forall n \in \mathbb{N},\ \mathbf{U}_n \leq \mathbf{V}_n \leq \mathbf{V}^*. \tag{17.11}$$

Proof: We first show that the function $\mathbf{\Phi}$ previously introduced is monotonic. Let $\mathbf{U}$ and $\mathbf{V}$ be such that $\mathbf{U} \leq \mathbf{V}$ and let π be the policy such that $\mathbf{\Phi}(\mathbf{U}) = \mathbf{R}_\pi + \gamma\mathbf{P}_\pi\mathbf{U}$.

Then,

$$\mathbf{\Phi}(\mathbf{U}) \leq \mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V} \leq \max_{\pi'} \{\mathbf{R}_{\pi'} + \gamma \mathbf{P}_{\pi'} \mathbf{V}\} = \mathbf{\Phi}(\mathbf{V}).$$

The proof is by induction on n. Assume that $\mathbf{U}_n \leq \mathbf{V}_n$, then by the monotonicity of $\mathbf{\Phi}$, we have

$$\mathbf{U}_{n+1} = \mathbf{\Phi}(\mathbf{U}_n) \leq \mathbf{\Phi}(\mathbf{V}_n) = \max_{\pi} \{\mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V}_n\}.$$

Let π_{n+1} be the maximizing policy, that is, $\pi_{n+1} = \text{argmax}_\pi \{\mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V}_n\}$. Then,

$$\mathbf{\Phi}(\mathbf{V}_n) = \mathbf{R}_{\pi_{n+1}} + \gamma \mathbf{P}_{\pi_{n+1}} \mathbf{V}_n \leq \mathbf{R}_{\pi_{n+1}} + \gamma \mathbf{P}_{\pi_{n+1}} \mathbf{V}_{n+1} = \mathbf{V}_{n+1},$$

and thus $\mathbf{U}_{n+1} \leq \mathbf{V}_{n+1}$. □

The theorem shows that the policy iteration algorithm converges in a smaller number of iterations than the value iteration algorithm due to the optimal policy. But, each iteration of the policy iteration algorithm requires computing a policy value, that is, solving a system of linear equations, which is more expensive to compute than an iteration of the value iteration algorithm.

17.4.3 Linear programming

An alternative formulation of the optimization problem defined by the Bellman equations (17.4) or the proof of Theorem 17.8 is via linear programming (LP), that is an optimization problem with a linear objective function and linear constraints. LPs admit (weakly) polynomial-time algorithmic solutions. There exist a variety of different methods for solving relatively large LPs in practice, using the simplex method, interior-point methods, or a variety of special-purpose solutions. All of these methods could be applied in this context.

By definition, the equations (17.4) are each based on a maximization. These maximizations are equivalent to seeking to minimize all elements of $\{V(s) \colon s \in S\}$ under the constraints $V(s) \geq \mathbb{E}[r(s,a)] + \gamma \sum_{s' \in S} \mathbb{P}[s'|s,a] V(s')$, $(s \in S)$. Thus, this can be written as the following LP for any set of fixed positive weights $\alpha(s) > 0$, $(s \in S)$:

$$\min_{\mathbf{V}} \quad \sum_{s \in S} \alpha(s) V(s) \tag{17.12}$$

$$\text{subject to} \quad \forall s \in S, \forall a \in A, V(s) \geq \mathbb{E}[r(s,a)] + \gamma \sum_{s' \in S} \mathbb{P}[s'|s,a] V(s'),$$

where $\boldsymbol{\alpha} > \mathbf{0}$ is the vector with the sth component equal to $\alpha(s)$.[25] To make each coefficient $\alpha(s)$ interpretable as a probability, we can further add the constraints that $\sum_{s \in S} \alpha(s) = 1$. The number of rows of this LP is $|S||A|$ and its number of columns $|S|$. The complexity of the solution techniques for LPs is typically more favorable in terms of the number of rows than the number of columns. This motivates a solution based on the equivalent dual formulation of this LP which can be written as

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{s \in S, a \in A} \mathbb{E}[r(s,a)]\, x(s,a) \\ \text{subject to} \quad & \forall s \in S, \sum_{a \in A} x(s',a) = \alpha(s') + \gamma \sum_{s \in S, a \in A} \mathbb{P}[s'|s,a]\, x(s',a) \\ & \forall s \in S, \forall a \in A, x(s,a) \geq 0, \end{aligned} \tag{17.13}$$

and for which the number of rows is only $|S|$ and the number of columns $|S||A|$. Here $x(s,a)$ can be interpreted as the probability of being in state s and taking action a.

17.5 Learning algorithms

This section considers the more general scenario where the environment model of an MDP, that is the transition and reward probabilities, is unknown. This matches many realistic applications of reinforcement learning where, for example, a robot is placed in an environment that it needs to explore in order to reach a specific goal.

How can an agent determine the best policy in this context? Since the environment models are not known, it may seek to learn them by estimating transition or reward probabilities. To do so, as in the standard case of supervised learning, the agent needs some amount of training information. In the context of reinforcement learning with MDPs, the training information is the sequence of immediate rewards the agent receives based on the actions it has taken.

There are two main learning approaches that can be adopted. One known as the *model-free approach* consists of learning an action policy directly. Another one, a *model-based* approach, consists of first learning the environment model, and then of using that to learn a policy. The Q-learning algorithm we present for this problem is widely adopted in reinforcement learning and belongs to the family of model-free approaches.

[25] Let us emphasize that the LP is only in terms of the variables $V(s)$, as indicated by the subscript of the minimization operator, and not in terms of $V(s)$ and $\alpha(s)$.

The estimation and algorithmic methods adopted for learning in reinforcement learning are closely related to the concepts and techniques in *stochastic approximation*. Thus, we start by introducing several useful results of this field that will be needed for the proofs of convergence of the reinforcement learning algorithms presented.

17.5.1 Stochastic approximation

Stochastic approximation methods are iterative algorithms for solving optimization problems whose objective function is defined as the expectation of some random variable, or to find the fixed point of a function H that is accessible only through noisy observations. These are precisely the type of optimization problems found in reinforcement learning. For example, for the Q-learning algorithm we will describe, the optimal state-action value function Q^* is the fixed point of some function H that is defined as an expectation and thus not directly accessible.

We start with a basic result whose proof and related algorithm show the flavor of more complex ones found in stochastic approximation. The theorem is a generalization of a result known as the *strong law of large numbers*. It shows that under some conditions on the coefficients, an iterative sequence of estimates μ_m converges almost surely (a.s.) to the mean of a bounded random variable.

Theorem 17.14 (Mean estimation) *Let X be a random variable taking values in $[0,1]$ and let $x_0, \ldots, x_m$ be i.i.d. values of X. Define the sequence $(\mu_m)_{m\in\mathbb{N}}$ by*

$$\mu_{m+1} = (1-\alpha_m)\mu_m + \alpha_m x_m, \tag{17.14}$$

with $\mu_0 = x_0$, $\alpha_m \in [0,1]$, $\sum_{m\geq 0} \alpha_m = +\infty$ and $\sum_{m\geq 0} \alpha_m^2 < +\infty$. Then,

$$\mu_m \xrightarrow{a.s.} \mathbb{E}[X]. \tag{17.15}$$

Proof: We give the proof of the L_2 convergence. The a.s. convergence is shown later for a more general theorem. By the independence assumption, for $m \geq 0$,

$$\operatorname{Var}[\mu_{m+1}] = (1-\alpha_m)^2 \operatorname{Var}[\mu_m] + \alpha_m^2 \operatorname{Var}[x_m] \leq (1-\alpha_m)\operatorname{Var}[\mu_m] + \alpha_m^2. \tag{17.16}$$

Let $\epsilon > 0$ and suppose that there exists $N \in \mathbb{N}$ such that for all $m \geq N$, $\operatorname{Var}[\mu_m] \geq \epsilon$. Then, for $m \geq N$,

$$\operatorname{Var}[\mu_{m+1}] \leq \operatorname{Var}[\mu_m] - \alpha_m \operatorname{Var}[\mu_m] + \alpha_m^2 \leq \operatorname{Var}[\mu_m] - \alpha_m \epsilon + \alpha_m^2,$$

which implies, by reapplying this inequality, that

$$\operatorname{Var}[\mu_{m+N}] \leq \underbrace{\operatorname{Var}[\mu_N] - \epsilon \sum_{n=N}^{m+N} \alpha_n + \sum_{n=N}^{m+N} \alpha_n^2}_{\to -\infty \text{ when } m \to \infty},$$

contradicting $\text{Var}[\mu_{m+N}] \geq 0$. Thus, this contradicts the existence of such an integer N. Therefore, for all $N \in \mathbb{N}$, there exists $m_0 \geq N$ such that $\text{Var}[\mu_{m_0}] \leq \epsilon$.

Choose N large enough so that for all $m \geq N$, the inequality $\alpha_m \leq \epsilon$ holds. This is possible since the sequence $(\alpha_m^2)_{m\in\mathbb{N}}$ and thus $(\alpha_m)_{m\in\mathbb{N}}$ converges to zero in view of $\sum_{m\geq 0} \alpha_m^2 < +\infty$. We will show by induction that for any $m \geq m_0$, $\text{Var}[\mu_m] \leq \epsilon$, which implies the statement of the theorem.

Assume that $\text{Var}[\mu_m] \leq \epsilon$ for some $m \geq m_0$. Then, using this assumption, inequality 17.16, and the fact that $\alpha_m \leq \epsilon$, the following inequality holds:

$$\text{Var}[\mu_{m+1}] \leq (1 - \alpha_m)\epsilon + \epsilon\alpha_m = \epsilon.$$

Thus, this proves that $\lim_{m\to+\infty} \text{Var}[\mu_m] = 0$, that is the L_2 convergence of μ_m to $\mathbb{E}[X]$. □

Note that the hypotheses of the theorem related to the sequence $(\alpha_m)_{m\in\mathbb{N}}$ hold in particular when $\alpha_m = \frac{1}{m}$. The special case of the theorem with this choice of α_m coincides with the strong law of large numbers. This result has tight connections with the general problem of stochastic optimization.

Stochastic optimization is the general problem of finding the solution to the equation

$$\mathbf{x} = H(\mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^N$, when

- $H(x)$ cannot be computed, for example, because H is not accessible or because the cost of its computation is prohibitive;
- but an i.i.d. sample of m noisy observations $H(\mathbf{x}_i) + \mathbf{w}_i$ are available, $i \in [m]$, where the noise random variable $\mathbf{w}$ has expectation zero: $\mathbb{E}[\mathbf{w}] = \mathbf{0}$.

This problem arises in a variety of different contexts and applications. As we shall see, it is directly related to the learning problem for MDPs.

One general idea for solving this problem is to use an iterative method and define a sequence $(\mathbf{x}_t)_{t\in\mathbb{N}}$ in a way similar to what is suggested by theorem 17.14:

$$\mathbf{x}_{t+1} = (1 - \alpha_t)\mathbf{x}_t + \alpha_t[H(\mathbf{x}_t) + \mathbf{w}_t] \tag{17.17}$$

$$= \mathbf{x}_t + \alpha_t[H(\mathbf{x}_t) + \mathbf{w}_t - \mathbf{x}_t], \tag{17.18}$$

where $(\alpha_t)_{t\in\mathbb{N}}$ follow conditions similar to those assumed in theorem 17.14. More generally, we consider sequences defined via

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha_t D(\mathbf{x}_t, \mathbf{w}_t), \tag{17.19}$$

where D is a function mapping $\mathbb{R}^N \times \mathbb{R}^N$ to $\mathbb{R}^N$. There are many different theorems guaranteeing the convergence of this sequence under various assumptions. We will present one of the most general forms of such theorems, which relies on the following result.

Theorem 17.15 (Supermartingale convergence) *Let $(X_t)_{t\in\mathbb{N}}$, $(Y_t)_{t\in\mathbb{N}}$, and $(Z_t)_{t\in\mathbb{N}}$ be sequences of non-negative random variables such that $\sum_{t=0}^{+\infty} Y_t < +\infty$. Let $\mathcal{F}_t$ denote all the information for $t' \leq t$: $\mathcal{F}_t = \{(X_{t'})_{t'\leq t}, (Y_{t'})_{t'\leq t}, (Z_{t'})_{t'\leq t}\}$. Then, if $\mathbb{E}\left[X_{t+1}\middle|\mathcal{F}_t\right] \leq X_t + Y_t - Z_t$, the following holds:*

- *X_t converges to a limit (with probability one).*
- *$\sum_{t=0}^{+\infty} Z_t < +\infty$.*

The following is one of the most general forms of such theorems.

Theorem 17.16 *Let D be a function mapping $\mathbb{R}^N \times \mathbb{R}^N$ to $\mathbb{R}^N$, $(\mathbf{w}_t)_{t\in\mathbb{N}}$ a sequence of random variables in $\mathbb{R}^N$, $(\alpha_t)_{t\in\mathbb{N}}$ a sequence of real numbers, and $(\mathbf{x}_t)_{t\in\mathbb{N}}$ a sequence defined by $\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha_t D(\mathbf{x}_t, \mathbf{w}_t)$ with $\mathbf{x}_0 \in \mathbb{R}^N$. Let $\mathcal{F}_t$ denote the entire history up to t, that is: $\mathcal{F}_t = \{(\mathbf{x}_{t'})_{t'\leq t}, (\mathbf{w}_{t'})_{t'\leq t-1}, (\alpha_{t'})_{t'\leq t}\}$, and let Ψ denote the function $\mathbf{x} \to \frac{1}{2}\|\mathbf{x}-\mathbf{x}^*\|_2^2$ for some $\mathbf{x}^* \in \mathbb{R}^N$. Assume that D and $(\alpha)_{t\in\mathbb{N}}$ verify the following conditions:*

- *$\exists K_1, K_2 \in \mathbb{R}\colon\ \mathbb{E}\left[\|D(\mathbf{x}_t, \mathbf{w}_t)\|_2^2 \,\middle|\, \mathcal{F}_t\right] \leq K_1 + K_2\,\Psi(\mathbf{x}_t)$;*
- *$\exists c \geq 0\colon \nabla\Psi(\mathbf{x}_t)^\top \mathbb{E}\left[D(\mathbf{x}_t, \mathbf{w}_t) \,\middle|\, \mathcal{F}_t\right] \leq -c\,\Psi(\mathbf{x}_t)$;*
- *$\alpha_t > 0, \sum_{t=0}^{+\infty} \alpha_t = +\infty, \sum_{t=0}^{+\infty} \alpha_t^2 < +\infty$.*

Then, the sequence $\mathbf{x}_t$ converges almost surely to $\mathbf{x}^*$:

$$\mathbf{x}_t \xrightarrow{a.s.} \mathbf{x}^*. \tag{17.20}$$

Proof: Since function Ψ is quadratic, a Taylor expansion gives

$$\Psi(\mathbf{x}_{t+1}) = \Psi(\mathbf{x}_t) + \nabla\Psi(\mathbf{x}_t)^\top(\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{1}{2}(\mathbf{x}_{t+1} - \mathbf{x}_t)^\top \nabla^2\Psi(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t).$$

Thus,

$$\begin{aligned}
\mathbb{E}\left[\Psi(\mathbf{x}_{t+1})\middle|\mathcal{F}_t\right] &= \Psi(\mathbf{x}_t) + \alpha_t \nabla\Psi(\mathbf{x}_t)^\top \mathbb{E}\left[D(\mathbf{x}_t, \mathbf{w}_t)\middle|\mathcal{F}_t\right] + \frac{\alpha_t^2}{2}\,\mathbb{E}\left[\|D(\mathbf{x}_t, \mathbf{w}_t)\|^2\middle|\mathcal{F}_t\right] \\
&\leq \Psi(\mathbf{x}_t) - \alpha_t c \Psi(\mathbf{x}_t) + \frac{\alpha_t^2}{2}(K_1 + K_2\Psi(\mathbf{x}_t)) \\
&= \Psi(\mathbf{x}_t) + \frac{\alpha_t^2 K_1}{2} - \Big(\alpha_t c - \frac{\alpha_t^2 K_2}{2}\Big)\Psi(\mathbf{x}_t).
\end{aligned}$$

Since by assumption the series $\sum_{t=0}^{+\infty} \alpha_t^2$ is convergent, $(\alpha_t^2)_t$ and thus $(\alpha_t)_t$ converges to zero. Therefore, for t sufficiently large, the term $\big(\alpha_t c - \frac{\alpha_t^2 K_2}{2}\big)\Psi(\mathbf{x}_t)$ has the sign of $\alpha_t c \Psi(\mathbf{x}_t)$ and is non-negative, since $\alpha_t > 0$, $\Psi(\mathbf{x}_t) \geq 0$, and $c > 0$. Thus, by the supermartingale convergence theorem 17.15, $\Psi(\mathbf{x}_t)$ converges and $\sum_{t=0}^{+\infty} \big(\alpha_t c - \frac{\alpha_t^2 K_2}{2}\big)\Psi(\mathbf{x}_t) < +\infty$. Since $\Psi(\mathbf{x}_t)$ converges and $\sum_{t=0}^{+\infty} \alpha_t^2 < +\infty$, we have $\sum_{t=0}^{+\infty} \frac{\alpha_t^2 K_2}{2}\Psi(\mathbf{x}_t) < +\infty$. But, since $\sum_{t=0}^{+\infty} \alpha_t = +\infty$, if the limit of $\Psi(\mathbf{x}_t)$ were non-zero, we would have $\sum_{t=0}^{+\infty} \alpha_t c \Psi(\mathbf{x}_t) = +\infty$. This implies

that the limit of $\Psi(\mathbf{x}_t)$ is zero, that is $\lim_{t\to\infty} \|\mathbf{x}_t - \mathbf{x}^*\|_2 \to 0$, which implies $\mathbf{x}_t \xrightarrow{\text{a.s.}} \mathbf{x}^*$. □

The following is another related result for which we do not present the full proof.

Theorem 17.17 *Let $\mathbf{H}$ be a function mapping $\mathbb{R}^N$ to $\mathbb{R}^N$, $(\mathbf{w}_t)_{t\in\mathbb{N}}$ a sequence of random variables in $\mathbb{R}^N$, $(\alpha_t)_{t\in\mathbb{N}}$ a sequence of real numbers, and $(\mathbf{x}_t)_{t\in\mathbb{N}}$ a sequence defined by*

$$\forall s \in [N], \quad \mathbf{x}_{t+1}(s) = \mathbf{x}_t(s) + \alpha_t(s)\big[\mathbf{H}(\mathbf{x}_t)(s) - \mathbf{x}_t(s) + \mathbf{w}_t(s)\big],$$

for some $\mathbf{x}_0 \in \mathbb{R}^N$. Define $\mathcal{F}_t$ by $\mathcal{F}_t = \{(\mathbf{x}_{t'})_{t'\leq t}, (\mathbf{w}_{t'})_{t'\leq t-1}(\alpha_{t'})_{t'\leq t}\}$, that is the entire history up to t, and assume that the following conditions are met:

- $\exists K_1, K_2 \in \mathbb{R}\colon \mathbb{E}\left[\|\mathbf{w}_t\|^2(s) \,\middle|\, \mathcal{F}_t\right] \leq K_1 + K_2 \|\mathbf{x}_t\|^2$ *for some norm $\|\cdot\|$;*
- $\mathbb{E}\left[\mathbf{w}_t \,\middle|\, \mathcal{F}_t\right] = 0$*;*
- $\forall s \in [N], \sum_{t=0}^{+\infty} \alpha_t(s) = +\infty, \sum_{t=0}^{+\infty} \alpha_t^2(s) < +\infty$*; and*
- $\mathbf{H}$ *is a $\|\cdot\|_\infty$-contraction with fixed point $\mathbf{x}^*$.*

Then, the sequence $\mathbf{x}_t$ converges almost surely to $\mathbf{x}^*$:

$$\mathbf{x}_t \xrightarrow{a.s.} \mathbf{x}^*. \tag{17.21}$$

The next sections present several learning algorithms for MDPs with an unknown model.

17.5.2 TD(0) algorithm

This section presents an algorithm, TD(0) algorithm, for evaluating a policy in the case where the environment model is unknown. The algorithm is based on Bellman's linear equations giving the value of a policy π (see proposition 17.9):

$$\begin{aligned} V_\pi(s) &= \mathbb{E}[r(s,\pi(s)] + \gamma \sum_{s'} \mathbb{P}[s'|s,\pi(s)] V_\pi(s') \\ &= \mathop{\mathbb{E}}_{s'}\big[r(s,\pi(s)) + \gamma V_\pi(s')|s\big]. \end{aligned}$$

However, here the probability distribution according to which this last expectation is defined is not known. Instead, the TD(0) algorithm consists of

- sampling a new state s'; and
- updating the policy values according to the following, which justifies the name of the algorithm:

$$\begin{aligned} V(s) &\leftarrow (1-\alpha)V(s) + \alpha[r(s,\pi(s)) + \gamma V(s')] \\ &= V(s) + \alpha[\underbrace{r(s,\pi(s)) + \gamma V(s') - V(s)}_{\text{temporal difference of } V \text{ values}}]. \end{aligned} \tag{17.22}$$

Here, the parameter α is a function of the number of visits to the state s.

```
TD(0)()
 1  V ← V_0 ▷ initialization.
 2  for t ← 0 to T do
 3       s ← SELECTSTATE()
 4       for each step of epoch t do
 5            r' ← REWARD(s, π(s))
 6            s' ← NEXTSTATE(π, s)
 7            V(s) ← (1 − α)V(s) + α[r' + γV(s')]
 8            s ← s'
 9  return V
```

The pseudocode of the algorithm is given above. The algorithm starts with an arbitrary policy value vector $\mathbf{V}_0$. An initial state is returned by SELECTSTATE at the beginning of each epoch. Within each epoch, the iteration continues until a final state is found. Within each iteration, action $\pi(s)$ is taken from the current state s following policy π. The new state s' reached and the reward r' received are observed. The policy value of state s is then updated according to the rule (17.22) and current state set to be s'.

The convergence of the algorithm can be proven using theorem 17.17. We will give instead the full proof of the convergence of the Q-learning algorithm, for which that of TD(0) can be viewed as a special case.

17.5.3 Q-learning algorithm

This section presents an algorithm for estimating the optimal state-action value function Q^* in the case of an unknown model. Note that the optimal policy or policy value can be straightforwardly derived from Q^* via: $\pi^*(s) = \operatorname{argmax}_{a\in A} Q^*(s,a)$ and $V^*(s) = \max_{a\in A} Q^*(s,a)$. To simplify the presentation, we will assume a deterministic reward function.

The Q-learning algorithm is based on the equations giving the optimal state-action value function Q^* (17.1):

$$\begin{aligned} Q^*(s,a) &= \mathbb{E}[r(s,a)] + \gamma \sum_{s'\in S} \mathbb{P}[s' \mid s,a] V^*(s') \\ &= \mathop{\mathbb{E}}_{s'}[r(s,a) + \gamma \max_{a\in A} Q^*(s',a)]. \end{aligned}$$

Q-LEARNING(π)

```
1  Q ← Q_0    ▷ initialization, e.g., Q_0 = 0.
2  for t ← 0 to T do
3      s ← SELECTSTATE()
4      for each step of epoch t do
5          a ← SELECTACTION(π, s) ▷   policy π derived from Q, e.g., ε-greedy.
6          r' ← REWARD(s, a)
7          s' ← NEXTSTATE(s, a)
8          Q(s, a) ← Q(s, a) + α[r' + γ max_{a'} Q(s', a') − Q(s, a)]
9          s ← s'
10 return Q
```

As for the policy values in the previous section, the distribution model is not known. Thus, the Q-learning algorithm consists of the following main steps:

- sampling a new state s'; and
- updating the policy values according to the following:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r(s,a) + \gamma \max_{a' \in A} Q(s',a')]. \tag{17.23}$$

where the parameter α is a function of the number of visits to the state s.

The algorithm can be viewed as a stochastic formulation of the value iteration algorithm presented in the previous section. The pseudocode is given above. Within each epoch, an action is selected from the current state s using a policy π derived from Q. The choice of the policy π is arbitrary so long as it guarantees that every pair (s,a) is visited infinitely many times. The reward received and the state s' observed are then used to update Q following (17.23).

Theorem 17.18 *Consider a finite MDP. Assume that for all $s \in S$ and $a \in A$, $\sum_{t=0}^{+\infty} \alpha_t(s,a) = +\infty$, and $\sum_{t=0}^{+\infty} \alpha_t^2(s,a) < +\infty$ with $\alpha_t(s,a) \in [0,1]$. Then, the Q-learning algorithm converges to the optimal value Q^* (with probability one).*

Note that the conditions on $\alpha_t(s,a)$ impose that each state-action pair is visited infinitely many times.

Proof: Let $(Q_t(s,a))_{t\geq 0}$ denote the sequence of state-action value functions at $(s,a) \in S \times A$ generated by the algorithm. By definition of the Q-learning updates,

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha\big[r(s_t,a_t) + \gamma \max_{a'} Q_t(s_{t+1},a') - Q_t(s_t,a_t)\big].$$

This can be rewritten as the following for all $s \in S$ and $a \in A$:

$$\begin{aligned} Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t(s,a)\left[r(s,a) + \gamma \mathop{\mathbb{E}}_{u\sim\mathbb{P}[\cdot|s,a]}\left[\max_{a'} Q_t(u,a')\right] - Q_t(s,a)\right] \\ + \gamma\alpha_t(s,a)\left[\max_{a'} Q_t(s',a') - \mathop{\mathbb{E}}_{u\sim\mathbb{P}[\cdot|s,a]}\left[\max_{a'} Q_t(u,a')\right]\right], \end{aligned} \tag{17.24}$$

if we define $s' = \textsc{NextState}(s,a)$ and $\alpha_t(s,a)$ as 0 if $(s,a) \neq (s_t,a_t)$ and $\alpha_t(s_t,a_t)$ otherwise. Now, let $\mathbf{Q}_t$ denote the vector with components $Q_t(s,a)$, $\mathbf{w}_t$ the vector whose s'th entry is

$$w_t(s) = \max_{a'} Q_t(s',a') - \mathop{\mathbb{E}}_{u\sim\mathbb{P}[\cdot|s,a]}\left[\max_{a'} Q_t(u,a')\right],$$

and $\mathbf{H}(\mathbf{Q}_t)$ the vector with components $\mathbf{H}(\mathbf{Q}_t)(s,a)$ defined by

$$\mathbf{H}(\mathbf{Q}_t)(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{u\sim\mathbb{P}[\cdot|s,a]}\left[\max_{a'} Q_t(u,a')\right].$$

Then, in view of (17.24),

$$\forall (s,a) \in S\times A, \quad \mathbf{Q}_{t+1}(s,a) = \mathbf{Q}_t(s,a) + \alpha_t(s,a)\big[\mathbf{H}(\mathbf{Q}_t)(s,a) - \mathbf{Q}_t(s,a) + \gamma \mathbf{w}_t(s)\big].$$

We now show that the hypotheses of theorem 17.17 hold for $\mathbf{Q}_t$ and $\mathbf{w}_t$, which will imply the convergence of $\mathbf{Q}_t$ to $\mathbf{Q}^*$. The conditions on α_t hold by assumption. By definition of $\mathbf{w}_t$, $\mathbb{E}[\mathbf{w}_t \,\big|\, \mathcal{F}_t] = 0$. Also, for any $s' \in S$,

$$\begin{aligned} |\mathbf{w}_t(s)| &\leq \max_{a'} |Q_t(s',a')| + \left|\mathop{\mathbb{E}}_{u\sim\mathbb{P}[\cdot|s,a]}\left[\max_{a'} Q_t(u,a')\right]\right| \\ &\leq 2 \max_{s'} |\max_{a'} Q_t(s',a')| = 2\|\mathbf{Q}_t\|_\infty. \end{aligned}$$

Thus, $\mathbb{E}\left[\mathbf{w}_t^2(s) \,\middle|\, \mathcal{F}_t\right] \leq 4\|\mathbf{Q}_t\|_\infty^2$. Finally, $\mathbf{H}$ is a γ-contraction for $\|\cdot\|_\infty$ since for any $\mathbf{Q}_1, \mathbf{Q}_2 \in \mathbb{R}^{|S|\times|A|}$, and $(s,a) \in S \times A$, we can write

$$\begin{aligned}
|\mathbf{H}(\mathbf{Q}_2)(x,a) - \mathbf{H}(\mathbf{Q}_1)(x,a)| &= \left|\gamma \mathop{\mathbb{E}}_{u\sim\mathbb{P}[\cdot|s,a]}\left[\max_{a'} Q_2(u,a') - \max_{a'} Q_1(u,a')\right]\right| \\
&\leq \gamma \mathop{\mathbb{E}}_{u\sim\mathbb{P}[\cdot|s,a]}\left[\left|\max_{a'} Q_2(u,a') - \max_{a'} Q_1(u,a')\right|\right] \\
&\leq \gamma \mathop{\mathbb{E}}_{u\sim\mathbb{P}[\cdot|s,a]} \max_{a'}\left[|Q_2(u,a') - Q_1(u,a')|\right] \\
&\leq \gamma \max_u \max_{a'}\left[|Q_2(u,a') - Q_1(u,a')|\right] \\
&= \gamma\|\mathbf{Q}_2 - \mathbf{Q}_1\|_\infty.
\end{aligned}$$

Since $\mathbf{H}$ is a contraction, it admits a fixed point $\mathbf{Q}^*$: $\mathbf{H}(\mathbf{Q}^*) = \mathbf{Q}^*$. □

The choice of the policy π according to which an action a is selected (line 5) is not specified by the algorithm and, as already indicated, the theorem guarantees the convergence of the algorithm for an arbitrary policy so long as it ensures that every pair (s,a) is visited infinitely many times. In practice, several natural choices are considered for π. One possible choice is the policy determined by the state-action value at time t, Q_t. Thus, the action selected from state s is $\operatorname{argmax}_{a\in A} Q_t(s,a)$. But this choice typically does not guarantee that all actions are taken or that all states are visited. Instead, a standard choice in reinforcement learning is the so-called *ϵ-greedy policy*, which consists of selecting with probability $(1-\epsilon)$ the greedy action from state s, that is, $\operatorname{argmax}_{a\in A} Q_t(s,a)$, and with probability ϵ a random action from s, for some $\epsilon \in (0,1)$. Another possible choice is the so-called *Boltzmann exploration*, which, given the current state-action value Q, epoch $t \in \{0,\ldots,T\}$, and current state s, consists of selecting action a with the following probability:

$$p_t(a|s,Q) = \frac{e^{\frac{Q(s,a)}{\tau_t}}}{\sum_{a'\in A} e^{\frac{Q(s,a')}{\tau_t}}},$$

where τ_t is the *temperature*. τ_t must be defined so that $\tau_t \to 0$ as $t \to +\infty$, which ensures that for large values of t, the greedy action based on Q is selected. This is natural, since as t increases, we can expect Q to be close to the optimal function. On the other hand, τ_t must be chosen so that it does not tend to 0 too fast to ensure that all actions are visited infinitely often. It can be chosen, for instance, as $1/\log(n_t(s))$, where $n_t(s)$ is the number of times s has been visited up to epoch t.

Reinforcement learning algorithms include two components: a *learning policy*, which determines the action to take, and an *update rule*, which defines the new estimate of the optimal value function. For an *off-policy algorithm*, the update rule does not necessarily depend on the learning policy. Q-learning is an off-policy algorithm since its update rule (line 8 of the pseudocode) is based on the max

operator and the comparison of all possible actions a', that is the greedy action, which may not coincide with the action recommended by the current the policy π. More generally, an off-policy algorithm evaluates or improves one policy, while acting based on another policy.

In contrast, the algorithm presented in the next section, SARSA, is an *on-policy algorithm*. An on-policy algorithm evaluates and improves the current policy used for control. It evaluates the return based on the algorithm's policy.

17.5.4 SARSA

SARSA is also an algorithm for estimating the optimal state-action value function in the case of an unknown model. The pseudocode is given in figure 17.7. The algorithm is in fact very similar to Q-learning, except that its update rule (line 9 of the pseudocode) is based on the action a' selected by the learning policy. Thus, SARSA is an on-policy algorithm, and its convergence therefore crucially depends on the learning policy. In particular, the convergence of the algorithm requires, in addition to all actions being selected infinitely often, that the learning policy becomes greedy in the limit. The proof of the convergence of the algorithm is nevertheless close to that of Q-learning.

The name of the algorithm derives from the sequence of instructions defining successively s, a, r', s', and a', and the fact that the update to the function Q depends on the quintuple (s, a, r', s', a).

17.5.5 TD(λ) algorithm

Both TD(0) and Q-learning algorithms are only based on immediate rewards. The idea of TD(λ) consists instead of using multiple steps ahead. Thus, for $n > 1$ steps, we would have the update

$$V(s) \leftarrow V(s) + \alpha\,(R_t^n - V(s)),$$

where R_t^n is defined by

$$R_t^n = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n V(s_{t+n}).$$

How should n be chosen? Instead of selecting a specific n, TD(λ) is based on a geometric distribution over all rewards R_t^n, that is, it uses $R_t^\lambda = (1-\lambda)\sum_{n=0}^{+\infty} \lambda^n R_t^n$ instead of R_t^n where $\lambda \in [0, 1]$. Thus, the main update becomes

$$V(s) \leftarrow V(s) + \alpha\,(R_t^\lambda - V(s)).$$

The pseudocode of the algorithm is given above. For $\lambda = 0$, the algorithm coincides with TD(0). $\lambda = 1$ corresponds to the total future reward.

SARSA(π)

$Q \leftarrow Q_0$ ▷ initialization, e.g., $Q_0 = 0$.
for $t \leftarrow 0$ **to** T **do**
    $s \leftarrow$ SELECTSTATE()
    $a \leftarrow$ SELECTACTION($\pi(Q), s$) ▷ policy π derived from Q, e.g., ϵ-greedy.
    for each step of epoch t **do**
        $r' \leftarrow$ REWARD(s, a)
        $s' \leftarrow$ NEXTSTATE(s, a)
        $a' \leftarrow$ SELECTACTION($\pi(Q), s'$) ▷ policy π derived from Q, e.g., ϵ-greedy.
        $Q(s,a) \leftarrow Q(s,a) + \alpha_t(s,a)\big[r' + \gamma Q(s',a') - Q(s,a)\big]$
        $s \leftarrow s'$
        $a \leftarrow a'$
return Q

Figure 17.7
The SARSA algorithm.

In the previous sections, we presented learning algorithms for an agent navigating in an unknown environment. The scenario faced in many practical applications is more challenging; often, the information the agent receives about the environment is uncertain or unreliable. Such problems can be modeled as partially observable Markov decision processes (POMDPs). POMDPs are defined by augmenting the definition of MDPs with an observation probability distribution depending on the action taken, the state reached, and the observation. The presentation of their model and solution techniques are beyond the scope of this material.

17.5.6 Large state space

In some cases in practice, the number of states or actions to consider for the environment may be very large. For example, the number of states in the game of backgammon is estimated to be over 10^{20}. Thus, the algorithms presented in the previous section can become computationally impractical for such applications. More importantly, generalization becomes extremely difficult.

Suppose we wish to estimate the policy value $V_\pi(s)$ at each state s using experience obtained using policy π. To cope with the case of large state spaces, we

```
TD(λ)()
 1  V ← V_0 ▷ initialization.
 2  e ← 0
 3  for t ← 0 to T do
 4       s ← SELECTSTATE()
 5       for each step of epoch t do
 6            s' ← NEXTSTATE(π, s)
 7            δ ← r(s, π(s)) + λV(s') − V(s)
 8            e(s) ← λe(s) + 1
 9            for u ∈ S do
10                 if u ≠ s then
11                      e(u) ← γλe(u)
12                 V(u) ← V(u) + αδe(u)
13            s ← s'
14  return V
```

can map each state of the environment to $\mathbb{R}^N$ via a mapping $\mathbf{\Phi}\colon S \to \mathbb{R}^N$, with N relatively small ($N \approx 200$ has been used for backgammon) and approximate $V_\pi(s)$ by a function $f_{\mathbf{w}}(s)$ parameterized by some vector $\mathbf{w}$. For example, $f_{\mathbf{w}}$ could be a linear function defined by $f_{\mathbf{w}}(s) = \mathbf{w} \cdot \mathbf{\Phi}(s)$ for all $s \in S$, or some more complex non-linear function of $\mathbf{w}$. The problem then consists of approximating V_π with $f_{\mathbf{w}}$ and can be formulated as a regression problem. Note, however, that the empirical data available is not i.i.d.

Suppose that at each time step t the agent receives the exact policy value $V_\pi(s_t)$. Then, if the family of functions $f_{\mathbf{w}}$ is differentiable, a gradient descent method applied to the empirical squared loss can be used to sequentially update the weight vector $\mathbf{w}$ via:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla_{\mathbf{w}_t} \frac{1}{2}[V_\pi(s_t) - f_{\mathbf{w}_t}(s_t)]^2 = \mathbf{w}_t + \alpha[V_\pi(s_t) - f_{\mathbf{w}_t}(s_t)]\nabla_{\mathbf{w}_t} f_{\mathbf{w}_t}(s_t).$$

It is worth mentioning, however, that for large action spaces, there are simple cases where the methods used do not converge and instead cycle.

17.6 Chapter notes

Reinforcement learning is an important area of machine learning with a large body of literature. This chapter presents only a brief introduction to this area. For a more detailed study, the reader could consult the book of Sutton and Barto [1998], whose mathematical content is short, or those of Puterman [1994] and Bertsekas [1987], which discuss in more depth several aspects, as well as the more recent book of Szepesvári [2010]. The Ph.D. theses of Singh [1993] and Littman [1996] are also excellent sources.

Some foundational work on MDPs and the introduction of the temporal difference (TD) methods are due to Sutton [1984]. Q-learning was introduced and analyzed by Watkins [1989], though it can be viewed as a special instance of TD methods. The first proof of the convergence of Q-learning was given by Watkins and Dayan [1992].

Many of the techniques used in reinforcement learning are closely related to those of stochastic approximation which originated with the work of Robbins and Monro [1951], followed by a series of results including Dvoretzky [1956], Schmetterer [1960], Kiefer and Wolfowitz [1952], and Kushner and Clark [1978]. For a recent survey of stochastic approximation, including a discussion of powerful proof techniques based on ODE (ordinary differential equations), see Kushner [2010] and the references therein. The connection with stochastic approximation was emphasized by Tsitsiklis [1994] and Jaakkola et al. [1994], who gave a related proof of the convergence of Q-learning. For the convergence rate of Q-learning, consult Even-Dar and Mansour [2003]. For recent results on the convergence of the policy iteration algorithm, see Ye [2011], which shows that the algorithm is strongly polynomial for a fixed discount factor.

Reinforcement learning has been successfully applied to a variety of problems including robot control, board games such as backgammon in which Tesauro's TD-Gammon reached the level of a strong master [Tesauro, 1995] (see also chapter 11 of Sutton and Barto [1998]), chess, elevator scheduling problems [Crites and Barto, 1996], telecommunications, inventory management, dynamic radio channel assignment [Singh and Bertsekas, 1997], and a number of other problems (see chapter 1 of Puterman [1994]).

Conclusion

We described a large variety of machine learning algorithms and techniques and discussed their theoretical foundations as well as their use and applications. While this is not a fully comprehensive presentation, it should nevertheless offer the reader some idea of the breadth of the field and its multiple connections with a variety of other domains, including statistics, information theory, optimization, game theory, and automata and formal language theory.

The fundamental concepts, algorithms, and proof techniques we presented should supply the reader with the necessary tools for analyzing other learning algorithms, including variants of the algorithms analyzed in this book. They are also likely to be helpful for devising new algorithms or for studying new learning schemes. We strongly encourage the reader to explore both and more generally to seek enhanced solutions for all theoretical, algorithmic, and applied learning problems.

The exercises included at the end of each chapter, as well as the full solutions we provide separately, should help the reader become more familiar with the techniques and concepts described. Some of them could also serve as a starting point for research work and the investigation of new questions.

Many of the algorithms we presented as well as their variants can be directly used in applications to derive effective solutions to real-world learning problems. Our detailed description of the algorithms and discussion should help with their implementation or their adaptation to other learning scenarios.

Machine learning is a relatively recent field and yet probably one of the most active ones in computer science. Given the wide accessibility of digitized data and its many applications, we can expect it to continue to grow at a very fast pace over the next few decades. Learning problems of different nature, some arising due to the substantial increase of the scale of the data, which already requires processing billions of records in some applications, others related to the introduction of completely new learning frameworks, are likely to pose new research challenges and require novel algorithmic solutions. In all cases, learning theory, algorithms,

and applications form an exciting area of computer science and mathematics, which we hope this book could at least partly communicate.

A Linear Algebra Review

In this appendix, we introduce some basic notions of linear algebra relevant to the material presented in this book. This appendix does not represent an exhaustive tutorial, and it is assumed that the reader has some prior knowledge of the subject.

A.1 Vectors and norms

We will denote by $\mathbb{H}$ a vector space whose dimension may be infinite.

A.1.1 Norms

Definition A.1 *A mapping $\Phi : \mathbb{H} \to \mathbb{R}_+$ is said to define a* norm *on $\mathbb{H}$ if it verifies the following axioms:*

- *definiteness:* $\forall \mathbf{x} \in \mathbb{H}, \Phi(\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$;
- *homogeneity:* $\forall \mathbf{x} \in \mathbb{H}, \forall \alpha \in \mathbb{R}, \Phi(\alpha \mathbf{x}) = |\alpha| \Phi(\mathbf{x})$;
- *triangle inequality:* $\forall \mathbf{x}, \mathbf{y} \in \mathbb{H}, \Phi(\mathbf{x} + \mathbf{y}) \leq \Phi(\mathbf{x}) + \Phi(\mathbf{y})$.

A norm is typically denoted by $\|\cdot\|$. Examples of vector norms are the absolute value on $\mathbb{R}$ and the Euclidean (or L_2) norm on $\mathbb{R}^N$. More generally, for any $p \geq 1$ the L_p norm is defined on $\mathbb{R}^N$ as

$$\forall \mathbf{x} \in \mathbb{R}^N, \quad \|\mathbf{x}\|_p = \Big(\sum_{j=1}^{N} |x_j|^p \Big)^{1/p}. \tag{A.1}$$

The L_1, L_2, and L_∞ norms are some of the most commonly used norms, where $\|\mathbf{x}\|_\infty = \max_{j \in [N]} |x_j|$. Two norms $\|\cdot\|$ and $\|\cdot\|'$ are said to be *equivalent* iff there exists $\alpha, \beta > 0$ such that for all $\mathbf{x} \in \mathbb{H}$,

$$\alpha \|\mathbf{x}\| \leq \|\mathbf{x}\|' \leq \beta \|\mathbf{x}\|. \tag{A.2}$$

The following general inequalities relating these norms can be proven straightforwardly:

$$\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{N} \|\mathbf{x}\|_2 \tag{A.3}$$

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{N} \|\mathbf{x}\|_\infty \tag{A.4}$$

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_1 \leq N \|\mathbf{x}\|_\infty. \tag{A.5}$$

The second inequality of the first line can be shown using the *Cauchy-Schwarz inequality* presented later while the other inequalities are clear. These inequalities show the equivalence of these three norms. More generally, all norms on a finite-dimensional space are equivalent. The following additional properties hold for the L_∞ norm: for all $\mathbf{x} \in \mathbb{H}$,

$$\forall p \geq 1, \ \|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_p \leq N^{1/p} \|\mathbf{x}\|_\infty \tag{A.6}$$

$$\lim_{p \to +\infty} \|\mathbf{x}\|_p = \|\mathbf{x}\|_\infty. \tag{A.7}$$

The inequalities of the first line are straightforward and imply the limit property of the second line.

Definition A.2 (Hilbert space) *A* Hilbert space *is a vector space equipped with an inner product* $\langle \cdot, \cdot \rangle$ *and that is* complete *(all Cauchy sequences are convergent). The inner product induces a norm defined as follows:*

$$\forall \mathbf{x} \in \mathbb{H}, \quad \|\mathbf{x}\|_{\mathbb{H}} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}. \tag{A.8}$$

A.1.2 Dual norms

Definition A.3 *Let* $\|\cdot\|$ *be a norm on* $\mathbb{R}^N$*. Then, the* dual norm $\|\cdot\|_*$ *associated to* $\|\cdot\|$ *is the norm defined by*

$$\forall \mathbf{y} \in \mathbb{R}^N, \quad \|\mathbf{y}\|_* = \sup_{\|\mathbf{x}\|=1} |\langle \mathbf{y}, \mathbf{x} \rangle|. \tag{A.9}$$

For any $p, q \geq 1$ that are *conjugate* that is such that $\frac{1}{p} + \frac{1}{q} = 1$, the L_p and L_q norms are dual norms of each other. In particular, the dual norm of L_2 is the L_2 norm, and the dual norm of the L_1 norm is the L_∞ norm.

Proposition A.4 (Hölder's inequality) *Let* $p, q \geq 1$ *be conjugate:* $\frac{1}{p} + \frac{1}{q} = 1$*. Then, for all* $x, y \in \mathbb{R}^N$*,*

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_p \|\mathbf{y}\|_q, \tag{A.10}$$

with equality when $|y_i| = |x_i|^{p-1}$ *for all* $i \in [N]$*.*

Proof: The statement holds trivially for $\mathbf{x} = \mathbf{0}$ or $\mathbf{y} = \mathbf{0}$; thus, we can assume $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{y} \neq \mathbf{0}$. Let $a, b > 0$. By the concavity of log (see definition B.7), we can write

$$\log\left(\frac{1}{p}a^p + \frac{1}{q}b^q\right) \geq \frac{1}{p}\log(a^p) + \frac{1}{q}\log(b^q) = \log(a) + \log(b) = \log(ab).$$

Taking the exponential of the left- and right-hand sides gives

$$\frac{1}{p}a^p + \frac{1}{q}b^q \geq ab,$$

which is known as *Young's inequality*. Using this inequality with $a = |x_j|/\|\mathbf{x}\|_p$ and $b = |y_j|/\|\mathbf{y}\|_q$ for $j \in [N]$ and summing up gives

$$\frac{\sum_{j=1}^N |x_j y_j|}{\|\mathbf{x}\|_p \|\mathbf{y}\|_q} \leq \frac{1}{p}\frac{\|\mathbf{x}\|^p}{\|\mathbf{x}\|^p} + \frac{1}{q}\frac{\|\mathbf{y}\|^q}{\|\mathbf{y}\|^q} = \frac{1}{p} + \frac{1}{q} = 1.$$

Since $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \sum_{j=1}^N |x_j y_j|$, the inequality claim follows. The equality case can be verified straightforwardly. □

Taking $p = q = 2$ immediately yields the following result known as the *Cauchy-Schwarz inequality*.

Corollary A.5 (Cauchy-Schwarz inequality) *For all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$*,*

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2, \tag{A.11}$$

with equality iff $\mathbf{x}$ *and* $\mathbf{y}$ *are collinear.*

Let $\mathcal{H}$ be the hyperplane in $\mathbb{R}^N$ whose equation is given by

$$\mathbf{w} \cdot \mathbf{x} + b = 0,$$

for some normal vector $\mathbf{w} \in \mathbb{R}^N$ and offset $b \in \mathbb{R}$. Let $d_p(\mathbf{x}, \mathcal{H})$ denote the distance of $\mathbf{x}$ to the hyperplane $\mathcal{H}$, that is,

$$d_p(\mathbf{x}, \mathcal{H}) = \inf_{\mathbf{x}' \in \mathcal{H}} \|\mathbf{x}' - \mathbf{x}\|_p. \tag{A.12}$$

Then, the following identity holds for all $p \geq 1$:

$$d_p(\mathbf{x}, \mathcal{H}) = \frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|_q}, \tag{A.13}$$

where q is the conjugate of p: $\frac{1}{p} + \frac{1}{q} = 1$. (A.13) can be shown by a straightforward application of the results of appendix B to the constrained optimization problem (A.12).

A.1.3 Relationship between norms

A general form for the inequalities seen in equations (A.3), (A.4) and (A.5), which holds for all L_p norms, is shown in the following proposition.

Proposition A.6 *Let $1 \leq p \leq q$. Then the following inequalities hold for all $\mathbf{x} \in \mathbb{R}^N$:*

$$\|x\|_q \leq \|x\|_p \leq N^{\frac{1}{p}-\frac{1}{q}} \|x\|_q \,. \tag{A.14}$$

Proof: First, assume $\mathbf{x} \neq \mathbf{0}$, otherwise the inequalities hold trivially. Then the first inequality holds using $1 \leq p \leq q$ as follows:

$$\left[\frac{\|\mathbf{x}\|_p}{\|\mathbf{x}\|_q}\right]^p = \sum_{i=1}^N \left[\underbrace{\frac{x_i}{\|\mathbf{x}\|_q}}_{\leq 1}\right]^p \geq \sum_{i=1}^N \left[\frac{x_i}{\|\mathbf{x}\|_q}\right]^q = 1.$$

Finally, the second inequality follows by using Hölder's inequality (proposition A.4)

$$\|\mathbf{x}\|_p = \left[\sum_{i=1}^N |x_i|^p\right]^{\frac{1}{p}} \leq \left[\left(\sum_{i=1}^N (|x_i|^p)^{\frac{q}{p}}\right)^{\frac{p}{q}} \left(\sum_{i=1}^N (1)^{\frac{q}{q-p}}\right)^{1-\frac{p}{q}}\right]^{\frac{1}{p}} = \|\mathbf{x}\|_q N^{\frac{1}{p}-\frac{1}{q}} \,,$$

which completes the proof. □

A.2 Matrices

For a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ with m rows and n columns, we denote by $\mathbf{M}_{ij}$ its ijth entry, for all $i \in [m]$ and $j \in [n]$. For any $m \geq 1$, we denote by $\mathbf{I}_m$ the m-dimensional identity matrix, and refer to it as $\mathbf{I}$ when the dimension is clear from the context.

The *transpose* of $\mathbf{M}$ is denoted by $\mathbf{M}^\top$ and defined by $(\mathbf{M}^\top)_{ij} = \mathbf{M}_{ji}$ for all (i,j). For any two matrices $\mathbf{M} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{n\times p}$, $(\mathbf{MN})^\top = \mathbf{N}^\top\mathbf{M}^\top$. $\mathbf{M}$ is said to be *symmetric* iff $\mathbf{M}_{ij} = \mathbf{M}_{ji}$ for all (i,j), that is, iff $\mathbf{M} = \mathbf{M}^\top$.

The *trace* of a square matrix $\mathbf{M}$ is denoted by $\mathrm{Tr}[\mathbf{M}]$ and defined as $\mathrm{Tr}[\mathbf{M}] = \sum_{i=1}^N \mathbf{M}_{ii}$. For any two matrices $\mathbf{M} \in \mathbb{R}^{m\times n}$ and $\mathbf{N} \in \mathbb{R}^{n\times m}$, the following identity holds: $\mathrm{Tr}[\mathbf{MN}] = \mathrm{Tr}[\mathbf{NM}]$. More generally, the following cyclic property holds with the appropriate dimensions for the matrices $\mathbf{M}$, $\mathbf{N}$, and $\mathbf{P}$:

$$\mathrm{Tr}[\mathbf{MNP}] = \mathrm{Tr}[\mathbf{PMN}] = \mathrm{Tr}[\mathbf{NPM}]. \tag{A.15}$$

The inverse of a square matrix $\mathbf{M}$, which exists when $\mathbf{M}$ has full rank, is denoted by $\mathbf{M}^{-1}$ and is the unique matrix satisfying $\mathbf{MM}^{-1} = \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$.

A.2.1 Matrix norms

A *matrix norm* is a norm defined over $\mathbb{R}^{m\times n}$ where m and n are the dimensions of the matrices considered. Many matrix norms, including those discussed below, satisfy the following *submultiplicative property*:

$$\|\mathbf{MN}\| \leq \|\mathbf{M}\|\|\mathbf{N}\|. \tag{A.16}$$

The *matrix norm induced* by the vector norm $\|\cdot\|_p$ or the *operator norm induced* by that norm is also denoted by $\|\cdot\|_p$ and defined by

$$\|\mathbf{M}\|_p = \sup_{\|\mathbf{x}\|_p \leq 1} \|\mathbf{Mx}\|_p \,. \tag{A.17}$$

The norm induced for $p = 2$ is known as the *spectral norm*, which equals the largest singular value of $\mathbf{M}$ (see section A.2.2), or the square-root of the largest eigenvalue of $\mathbf{M}^\top\mathbf{M}$:

$$\|\mathbf{M}\|_2 = \sigma_1(\mathbf{M}) = \sqrt{\lambda_{\max}(\mathbf{M}^\top\mathbf{M})}. \tag{A.18}$$

Not all matrix norms are induced by vector norms. The *Frobenius norm* denoted by $\|\cdot\|_F$ is the most notable of such norms and is defined by:

$$\|\mathbf{M}\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{M}_{ij}^2\right)^{1/2}.$$

The Frobenius norm can be interpreted as the L_2 norm of a vector when treating $\mathbf{M}$ as a vector of size mn. It also coincides with the norm induced by the *Frobenius product*, which is the inner product defined for all $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{m\times n}$ by

$$\langle \mathbf{M}, \mathbf{N}\rangle_F = \mathrm{Tr}[\mathbf{M}^\top \mathbf{N}]. \tag{A.19}$$

This relates the Frobenius norm to the singular values of $\mathbf{M}$:

$$\|\mathbf{M}\|_F^2 = \mathrm{Tr}[\mathbf{M}^\top\mathbf{M}] = \sum_{i=1}^{r}\sigma_i(\mathbf{M})^2\,,$$

where $r = \mathrm{rank}(\mathbf{M})$. The second equality follows from properties of SPSD matrices (see section A.2.3).

For any $j \in [n]$, let $\mathbf{M}_j$ denote the jth column of $\mathbf{M}$, that is $\mathbf{M} = [\mathbf{M}_1 \cdots \mathbf{M}_n]$. Then, for any $p, r \geq 1$, the $L_{p,r}$ *group norm* of $\mathbf{M}$ is defined by

$$\|\mathbf{M}\|_{p,r} = \left(\sum_{j=1}^{n}\|\mathbf{M}_i\|_p^r\right)^{1/r}.$$

One of the most commonly used group norms is the $L_{2,1}$ norm defined by

$$\|\mathbf{M}\|_{2,1} = \sum_{i=1}^{n}\|\mathbf{M}_i\|_2\,.$$

A.2.2 Singular value decomposition

The compact *singular value decomposition (SVD)* of $\mathbf{M}$, with $r = \mathrm{rank}(\mathbf{M}) \leq \min(m,n)$, can be written as follows:

$$\mathbf{M} = \mathbf{U}_M\mathbf{\Sigma}_M\mathbf{V}_M^\top\,.$$

The $r \times r$ matrix $\mathbf{\Sigma}_M = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ is diagonal and contains the non-zero *singular values* of $\mathbf{M}$ sorted in decreasing order, that is $\sigma_1 \geq \ldots \geq \sigma_r > 0$. The matrices $\mathbf{U}_M \in \mathbb{R}^{m\times r}$ and $\mathbf{V}_M \in \mathbb{R}^{n\times r}$ have orthonormal columns that contain the *left* and *right singular vectors* of $\mathbf{M}$ corresponding to the sorted singular values. We denote by $\mathbf{U}_k \in \mathbb{R}^{m\times k}$ the top $k \leq r$ left singular vectors of $\mathbf{M}$.

The *orthogonal projection* onto the span of $\mathbf{U}_k$ can be written as $\mathbf{P}_{U_k} = \mathbf{U}_k\mathbf{U}_k^\top$, where $\mathbf{P}_{U_k}$ is SPSD and idempotent, i.e., $\mathbf{P}_{U_k}^2 = \mathbf{P}_{U_k}$. Moreover, the orthogonal projection onto the subspace orthogonal to $\mathbf{U}_k$ is defined as $\mathbf{P}_{U_k,\perp}$. Similar definitions, i.e., $\mathbf{V}_k, \mathbf{P}_{V_k}, \mathbf{P}_{V_k,\perp}$, hold for the right singular vectors.

The *generalized inverse*, or *Moore-Penrose pseudo-inverse* of a matrix $\mathbf{M}$ is denoted by $\mathbf{M}^\dagger$ and defined by

$$\mathbf{M}^\dagger = \mathbf{U}_M\mathbf{\Sigma}_M^\dagger\mathbf{V}_M^\top, \tag{A.20}$$

where $\mathbf{\Sigma}_M^\dagger = \mathrm{diag}(\sigma_1^{-1}, \ldots, \sigma_r^{-1})$. For any square $m \times m$ matrix $\mathbf{M}$ with full rank, i.e., $r = m$, the pseudo-inverse coincides with the matrix inverse: $\mathbf{M}^\dagger = \mathbf{M}^{-1}$.

A.2.3 Symmetric positive semidefinite (SPSD) matrices

Definition A.7 *A symmetric matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ is said to be* positive semidefinite *iff*

$$\mathbf{x}^\top\mathbf{M}\mathbf{x} \geq 0 \tag{A.21}$$

for all $\mathbf{x} \in \mathbb{R}^m$. $\mathbf{M}$ is said to be positive definite *if the inequality is strict.*

Kernel matrices (see chapter 6) and orthogonal projection matrices are two examples of SPSD matrices. It is straightforward to show that a matrix $\mathbf{M}$ is SPSD iff its eigenvalues are all non-negative. Furthermore, the following properties hold for any SPSD matrix $\mathbf{M}$:

- $\mathbf{M}$ admits a decomposition $\mathbf{M} = \mathbf{X}^\top\mathbf{X}$ for some matrix $\mathbf{X}$ and the *Cholesky decomposition* provides one such decomposition in which $\mathbf{X}$ is an upper triangular matrix.
- The left and right singular vectors of $\mathbf{M}$ are the same and the SVD of $\mathbf{M}$ is also its eigenvalue decomposition.
- The SVD of an arbitrary matrix $\mathbf{X} = \mathbf{U}_X\mathbf{\Sigma}_X\mathbf{V}_X^\top$ defines the SVD of two related SPSD matrices: the left singular vectors ($\mathbf{U}_X$) are the eigenvectors of $\mathbf{X}\mathbf{X}^\top$, the right singular vectors ($\mathbf{V}_X$) are the eigenvectors of $\mathbf{X}^\top\mathbf{X}$ and the non-zero singular values of $\mathbf{X}$ are the square roots of the non-zero eigenvalues of $\mathbf{X}\mathbf{X}^\top$ and $\mathbf{X}^\top\mathbf{X}$.
- The trace of $\mathbf{M}$ is the sum of its singular values, i.e., $\mathrm{Tr}[\mathbf{M}] = \sum_{i=1}^{r}\sigma_i(\mathbf{M})$, where $\mathrm{rank}(\mathbf{M}) = r$.
- The top singular vector of $\mathbf{M}$, $\mathbf{u}_1$, maximizes the *Rayleigh quotient*, which is defined as

$$r(\mathbf{x}, \mathbf{M}) = \frac{\mathbf{x}^\top\mathbf{M}\mathbf{x}}{\mathbf{x}^\top\mathbf{x}}\,.$$

  In other words, $\mathbf{u}_1 = \mathrm{argmax}_{\mathbf{x}}\, r(\mathbf{x}, \mathbf{M})$ and $r(\mathbf{u}, \mathbf{M}) = \sigma_1(\mathbf{M})$. Similarly, if $\mathbf{M}' = \mathbf{P}_{U_i,\perp}\mathbf{M}$, that is, the projection of $\mathbf{M}$ onto the subspace orthogonal to $\mathbf{U}_i$, then $\mathbf{u}_{i+1} = \mathrm{argmax}_{\mathbf{x}}\, r(\mathbf{x}, \mathbf{M}')$, where $\mathbf{u}_{i+1}$ is the $(i+1)$st singular vector of $\mathbf{M}$.

B Convex Optimization

In this appendix, we introduce the main definitions and results of convex optimization needed for the analysis of the learning algorithms presented in this book.

B.1 Differentiation and unconstrained optimization

We start with some basic definitions for differentiation needed to present Fermat's theorem and to describe some properties of convex functions.

Definition B.1 (Gradient) *Let $f\colon \mathcal{X} \subseteq \mathbb{R}^N \to \mathbb{R}$ be a differentiable function. Then, the* gradient *of f at $\mathbf{x} \in \mathcal{X}$ is the vector in $\mathbb{R}^N$ denoted by $\nabla f(\mathbf{x})$ and defined by*

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{x}_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial \mathbf{x}_N}(\mathbf{x}) \end{bmatrix}.$$

Definition B.2 (Hessian) *Let $f\colon \mathcal{X} \subseteq \mathbb{R}^N \to \mathbb{R}$ be a twice differentiable function. Then, the* Hessian *of f at $\mathbf{x} \in \mathcal{X}$ is the matrix in $\mathbb{R}^{N \times N}$ denoted by $\nabla^2 f(\mathbf{x})$ and defined by*

$$\nabla^2 f(\mathbf{x}) = \Big[\frac{\partial^2 f}{\partial \mathbf{x}_i, \mathbf{x}_j}(\mathbf{x})\Big]_{1 \leq i,j \leq N}.$$

Next, we present a classic result for unconstrained optimization.

Theorem B.3 (Fermat's theorem) *Let $f\colon \mathcal{X} \subseteq \mathbb{R}^N \to \mathbb{R}$ be a differentiable function. If f admits a local extremum at $\mathbf{x}^* \in \mathcal{X}$, then $\nabla f(\mathbf{x}^*) = 0$, that is, $\mathbf{x}^*$ is a* stationary point.

B.2 Convexity

This section introduces the notions of *convex sets* and *convex functions*. Convex functions play an important role in the design and analysis of learning algorithms, in part because a local minimum of a convex function is necessarily also a global minimum. Thus, the properties of a hypothesis that is learned by finding a local minimum of a convex optimization are often well understood, while for some non-convex optimization problems there may be a very large number of local minima for which no clear characterization of the learned hypothesis can be given.

Definition B.4 (Convex set) *A set $\mathcal{X} \subseteq \mathbb{R}^N$ is said to be* convex *if for any two points $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ the segment $[\mathbf{x}, \mathbf{y}]$ lies in $\mathcal{X}$, that is*

$$\{\alpha \mathbf{x} + (1 - \alpha)\mathbf{y}\colon 0 \leq \alpha \leq 1\} \subseteq \mathcal{X}.$$

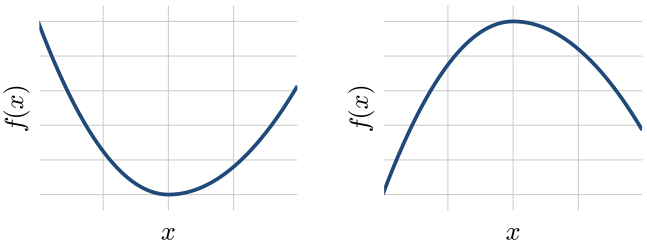


Figure B.1
Examples of a convex (left) and a concave (right) functions. Note that any line segment drawn between two points on the convex function lies entirely above the graph of the function while any line segment drawn between two points on the concave function lies entirely below the graph of the function.

The following lemma illustrates several operations on convex sets that preserve convexity. These will be useful for proving several subsequent results of this section.

Lemma B.5 (Operations that preserve convexity of sets) *The following operations on convex sets preserve convexity:*

- *Let* $\{\mathcal{C}_i\}_{i\in I}$ *be any family of sets where for all* $i \in I$ *the set* $\mathcal{C}_i$ *is convex. Then the intersection of these sets* $\bigcap_{i\in I} \mathcal{C}_i$ *is also convex.*
- *Let* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *be convex sets, then their sum* $\mathcal{C}_1 + \mathcal{C}_2 = \{x_1 + x_2 : x_1 \in \mathcal{C}_1, x_2 \in \mathcal{C}_2\}$*, when defined, is convex.*
- *Let* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *be convex sets, then their cross-product* $(\mathcal{C}_1 \times \mathcal{C}_2)$ *is also convex.*
- *Any projection of a convex set* $\mathcal{C}$ *is also convex.*

Proof: The first property holds since for any $x, y \in \bigcap_{i\in I} \mathcal{C}_i$ and any $\alpha \in [0, 1]$, we have $\alpha x + (1-\alpha)y \in \mathcal{C}_i$ for any $i \in I$ by the convexity of $\mathcal{C}_i$.

The second property holds since for any $(x_1 + x_2), (y_1 + y_2) \in (\mathcal{C}_1 + \mathcal{C}_2)$ we have $\alpha(x_1 + x_2) + (1-\alpha)(y_1 + y_2) = (\alpha x_1 + (1-\alpha)y_1 + \alpha x_2 + (1-\alpha)y_2) \in (\mathcal{C}_1 + \mathcal{C}_2)$, which follows since $\alpha x_1 + (1-\alpha)y_1 \in \mathcal{C}_1$ and $\alpha x_2 + (1-\alpha)y_2 \in \mathcal{C}_2$.

The third property holds since for $(x_1, x_2), (y_1, y_2) \in (\mathcal{C}_1 \times \mathcal{C}_2)$ we have $\alpha(x_1, x_2) + (1-\alpha)(y_1, y_2) = (\alpha x_1 + (1-\alpha)y_1, \alpha x_2 + (1-\alpha)y_2) \in (\mathcal{C}_1 \times \mathcal{C}_2)$, where the membership holds due to the assumption that $\mathcal{C}_1$ and $\mathcal{C}_2$ are convex.

Finally, the fourth property holds by noting that for any decomposition of the convex set $\mathcal{C}$ into projections $\mathcal{C}_1$ and $\mathcal{C}_2$, such that $\mathcal{C} = (\mathcal{C}_1 \times \mathcal{C}_2)$, it must be the case that $\mathcal{C}_1$ is convex. If $\mathcal{C}_2$ is empty, then the result is trivially true. Otherwise, fix an element $x_2 \in \mathcal{C}_2$, then for any $x, y \in \mathcal{C}_1$ and any $\alpha \in [0, 1]$ we have $\alpha(x, x_2) + (1-\alpha)(y, x_2) \in \mathcal{C}$, which implies $\alpha x + (1-\alpha)y \in \mathcal{C}_1$. Since $\mathcal{C}_1$ was chosen arbitrarily, this fact holds for any projection of $\mathcal{C}$. □

Note that many set operations may not preserve convexity. Consider the union of disjoint intervals on $\mathbb{R}$: $[a, b] \cup [c, d]$ where $a < b < c < d$. Clearly $[a, b]$ and $[c, d]$ are convex, however we have $\frac{1}{2}b + (1 - \frac{1}{2})c \notin ([a, b] \cup [c, d])$.

Definition B.6 (Convex hull) *The* convex hull $\text{conv}(\mathcal{X})$ *of a set of points* $\mathcal{X} \subseteq \mathbb{R}^N$ *is the minimal convex set containing* $\mathcal{X}$ *and can be equivalently defined as follows:*

$$\text{conv}(\mathcal{X}) = \Big\{ \sum_{i=1}^{m} \alpha_i \mathbf{x}_i : m \geq 1, \forall i \in [m], \mathbf{x}_i \in \mathcal{X}, \alpha_i \geq 0, \sum_{i=1}^{m} \alpha_i = 1 \Big\}. \tag{B.1}$$

Let $\text{Epi}\, f$ denote the *epigraph* of function $f : \mathcal{X} \to \mathbb{R}$, that is the set of points lying above its graph: $\{(x, y) : x \in \mathcal{X}, y \geq f(x)\}$.

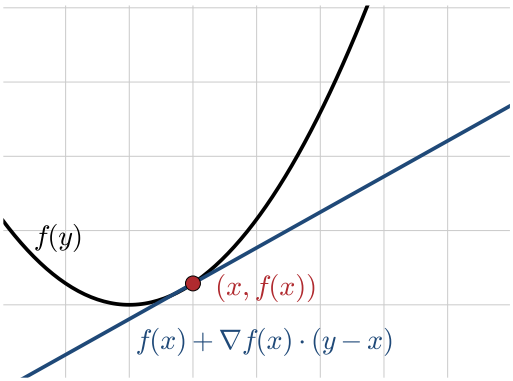


Figure B.2
Illustration of the first-order property satisfied by all convex functions.

Definition B.7 (Convex function) *Let $\mathcal{X}$ be a convex set. A function $f\colon \mathcal{X} \to \mathbb{R}$ is said to be* convex *iff* Epi f *is a convex set, or, equivalently, if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ and $\alpha \in [0, 1]$,*

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y})\,. \tag{B.2}$$

f is said to be *strictly convex* if inequality (B.2) is strict for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ where $\mathbf{x} \neq \mathbf{y}$ and $\alpha \in (0, 1)$. f is said to be (strictly) *concave* when $-f$ is (strictly) convex. Figure B.1 shows simple examples of convex and concave functions. Convex functions can also be characterized in terms of their first- or second-order differential.

Theorem B.8 *Let f be a differentiable function, then f is convex if and only if* $\mathrm{dom}(f)$ *is convex and the following inequalities hold:*

$$\forall \mathbf{x}, \mathbf{y} \in \mathrm{dom}(f),\ \ f(\mathbf{y}) - f(\mathbf{x}) \geq \nabla f(\mathbf{x}) \cdot (\mathbf{y} - \mathbf{x})\,. \tag{B.3}$$

The property (B.3) is illustrated by figure B.2: for a convex function, the hyperplane tangent at $\mathbf{x}$ is always below the graph.

Theorem B.9 *Let f be a twice differentiable function, then f is convex iff* $\mathrm{dom}(f)$ *is convex and its Hessian is positive semidefinite:*

$$\forall \mathbf{x} \in \mathrm{dom}(f),\ \ \nabla^2 f(\mathbf{x}) \succeq 0\,.$$

Recall that a symmetric matrix is positive semidefinite if all of its eigenvalues are non-negative. Further, note that when f is scalar, this theorem states that f is convex if and only if its second derivative is always non-negative, that is, for all $x \in \mathrm{dom}(f)$, $f''(x) \geq 0$.

Example B.10 (Linear functions) Any linear function f is both convex and concave, since equation (B.2) holds with equality for both f and $-f$ by the definition of linearity.

Example B.11 (Quadratic function) The function $f\colon x \mapsto x^2$ defined over $\mathbb{R}$ is convex since it is twice differentiable and for all $x \in \mathbb{R}$, $f''(x) = 2 > 0$.

Example B.12 (Norms) Any norm $\|\cdot\|$ defined over a convex set $\mathcal{X}$ is convex since by the triangle inequality and the homogeneity property of the norm, for all $\alpha \in [0, 1], \mathbf{x}, \mathbf{y} \in \mathcal{X}$, we can write

$$\|\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}\| \leq \|\alpha\mathbf{x}\| + \|(1 - \alpha)\mathbf{y}\| = \alpha\|\mathbf{x}\| + (1 - \alpha)\|\mathbf{y}\|\,.$$

Example B.13 (Maximum function) The max function defined for all $\mathbf{x} \in \mathbb{R}^N$, by $\mathbf{x} \mapsto \max_{j \in [N]} \mathbf{x}_j$ is convex. For all $\alpha \in [0,1], \mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, by the sub-additivity of max, we can write

$$\max_j(\alpha \mathbf{x}_j + (1-\alpha)\mathbf{y}_j) \leq \max_j(\alpha \mathbf{x}_j) + \max_j((1-\alpha)\mathbf{y}_j) = \alpha \max_j(\mathbf{x}_j) + (1-\alpha) \max_j(\mathbf{y}_j)\,.$$

One useful approach for proving convexity or concavity of functions is to make use of composition rules. For simplicity of presentation, we will assume twice differentiability, although the results can also be proven without this assumption.

Lemma B.14 (Composition of convex/concave functions) *Assume $h : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R}^N \to \mathbb{R}$ are twice differentiable functions and for all $\mathbf{x} \in \mathbb{R}^N$, define $f(\mathbf{x}) = h(g(\mathbf{x}))$. Then the following implications are valid:*

- *h is convex and non-decreasing, and g is convex $\implies$ f is convex.*
- *h is convex and non-increasing, and g is concave $\implies$ f is convex.*
- *h is concave and non-decreasing, and g is concave $\implies$ f is concave.*
- *h is concave and non-increasing, and g is convex $\implies$ f is concave.*

Proof: We restrict ourselves to $N = 1$, since it suffices to prove convexity (concavity) along all arbitrary lines that intersect the domain. Now, consider the second derivative of f:

$$f''(x) = h''(g(x))g'(x)^2 + h'(g(x))g''(x)\,. \tag{B.4}$$

Note that if h is convex and non-decreasing, we have $h'' \geq 0$ and $h' \geq 0$. Furthermore, if g is convex we also have $g'' \geq 0$, and it follows that $f''(x) \geq 0$, which proves the first statement. The remainder of the statements are proven in a similar manner. □

Example B.15 (Composition of functions) The previous lemma shows the convexity or concavity of the following composed functions:

- If $f : \mathbb{R}^N \to \mathbb{R}$ is convex, then $\exp(f)$ is convex.
- Any squared norm $\|\cdot\|^2$ is convex.
- For all $\mathbf{x} \in \mathbb{R}^N$ the function $\mathbf{x} \mapsto \log(\sum_{j=1}^N x_j)$ is concave.

The following two lemmas give examples of two other operations preserving convexity.

Lemma B.16 (Pointwise supremum or maximum of convex functions) *Let $(f_i)_{i \in \mathcal{I}}$ be a family of convex functions defined over a convex set $\mathcal{C}$. Then, their* pointwise supremum *f defined for all $x \in \mathcal{C}$ by $f(x) = \sup_{i \in \mathcal{I}} f_i(x)$ (resp. their* pointwise maximum *if $|\mathcal{I}| < +\infty$) is a convex function.*

Proof: Observe that $\operatorname{Epi} f = \cap_{i \in \mathcal{I}} \operatorname{Epi} f_i$ and is therefore convex as an intersection of convex sets. □

Example B.17 (Pointwise supremum of convex functions) The lemma shows in particular the convexity of the following functions:

- A piecewise linear function f defined for all $x \in \mathbb{R}^N$ by $f(x) = \max_{i \in [m]} \mathbf{w}_i^\top \mathbf{x} + b_i$ is convex as a pointwise maximum of affine (and thus convex) functions.
- The maximum eigenvalue $\lambda_{\max}(\mathbf{M})$ is a convex function over the set of symmetric matrices $\mathbf{M}$ since the set of symmetric matrices is convex and since $\lambda_{\max}(\mathbf{M}) = \sup_{\|\mathbf{x}\|_2 \leq 1} \mathbf{x}^\top \mathbf{M}\mathbf{x}$ is defined as the supremum of the linear (and thus convex) functions $\mathbf{M} \mapsto \mathbf{x}^\top \mathbf{M}\mathbf{x}$.
- More generally, let $\lambda_1(\mathbf{M}), \ldots, \lambda_k(\mathbf{M})$ denote the top $k \leq n$ eigenvalues of a symmetric $n \times n$ matrix $\mathbf{M}$. Then, by a similar argument, $\mathbf{M} \mapsto \sum_{i=1}^k \lambda_i(\mathbf{M})$ is a convex function since $\sum_{i=1}^k \lambda_i(\mathbf{M}) = \sup_{\dim(\mathbf{V})=k} \sum_{i=1}^k \mathbf{u}_i^\top \mathbf{M}\mathbf{u}_i$, where $\mathbf{u}_1, \ldots, \mathbf{u}_k$ is an orthonormal basis of $\mathbf{V}$.
- Using the previous property, along with the fact that $\operatorname{Tr}(\mathbf{M})$ is linear in $\mathbf{M}$, also shows that $\mathbf{M} \mapsto \sum_{i=k+1}^n \lambda_i(\mathbf{M}) = \operatorname{Tr}(\mathbf{M}) - \sum_{i=1}^k \lambda_i(\mathbf{M})$ or $\mathbf{M} \mapsto \sum_{i=n-k+1}^n \lambda_i(\mathbf{M}) = -\sum_{i=1}^k \lambda_i(-\mathbf{M})$ are concave functions.

Lemma B.18 (Partial infimum) *Let f be a convex function defined over a convex set $\mathcal{C} \subseteq \mathcal{X} \times \mathcal{Y}$ and let $\mathcal{B} \subseteq \mathcal{Y}$ be a convex set such that $\mathcal{A} = \{x \in \mathcal{X} \colon \exists y \in \mathcal{B} \mid (x, y) \in \mathcal{C}\}$ is non-empty. Then, $\mathcal{A}$ is a convex set and the function g defined for all $x \in \mathcal{A}$ by $g(x) = \inf_{y \in \mathcal{B}} f(x, y)$ is convex.*

Proof: First note that the intersection of the convex sets $\mathcal{C}$ and $(\mathcal{X} \times \mathcal{B})$ is convex. Thus, $\mathcal{A}$ is convex since it is the projection of the convex set $\mathcal{C} \cap (\mathcal{X} \times \mathcal{B})$ onto $\mathcal{X}$.

Let x_1 and x_2 be in $\mathcal{A}$. By definition of g, for any $\epsilon > 0$, there exist $y_1, y_2 \in \mathcal{B}$ with $(x_1, y_1), (x_2, y_2) \in \mathcal{C}$ such that $f(x_1, y_1) \leq g(x_1) + \epsilon$ and $f(x_2, y_2) \leq g(x_2) + \epsilon$. Then, for any $\alpha \in [0, 1]$,

$$\begin{aligned} g(\alpha x_1 + (1-\alpha)x_2) &= \inf_{y \in \mathcal{B}} f(\alpha x_1 + (1-\alpha)x_2, y) \\ &\leq f(\alpha x_1 + (1-\alpha)x_2, \alpha y_1 + (1-\alpha)y_2) \\ &\leq \alpha f(x_1, y_1) + (1-\alpha) f(x_2, y_2) \\ &\leq \alpha g(x_1) + (1-\alpha) g(x_2) + \epsilon. \end{aligned}$$

Since the inequality holds for all $\epsilon > 0$, it implies

$$g(\alpha x_1 + (1-\alpha)x_2) \leq \alpha g(x_1) + (1-\alpha) g(x_2),$$

which completes the proof. □

Example B.19 The lemma shows in particular that the distance to a convex set $\mathcal{B}$, $d(x, \mathcal{B}) = \inf_{y \in \mathcal{B}} \|x - y\|$, is a convex function of x in any normed vector space, since $(x, y) \mapsto \|x - y\|$ is jointly convex in x and y for any norm $\|\cdot\|$.

The following is a useful inequality applied in a variety of contexts. It is in fact a quasi-direct consequence of the definition of convexity.

Theorem B.20 (Jensen's inequality) *Let X be a random variable taking values in a non-empty convex set $\mathcal{C} \subseteq \mathbb{R}^N$ with a finite expectation $\mathbb{E}[X]$, and f a measurable convex function defined over $\mathcal{C}$. Then, $\mathbb{E}[X]$ is in $\mathcal{C}$, $\mathbb{E}[f(X)]$ is finite, and the following inequality holds:*

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)].$$

Proof: We give a sketch of the proof, which essentially follows from the definition of convexity. Note that for any finite set of elements $x_1, \ldots, x_n$ in $\mathcal{C}$ and any positive reals $\alpha_1, \ldots, \alpha_n$ such that $\sum_{i=1}^n \alpha_i = 1$, we have

$$f\Big(\sum_{i=1}^n \alpha_i x_i\Big) \leq \sum_{i=1}^n \alpha_i f(x_i)\,.$$

This follows straightforwardly by induction from the definition of convexity. Since the α_is can be interpreted as probabilities, this immediately proves the inequality for any distribution with a finite support defined by $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$:

$$f(\mathop{\mathbb{E}}_{\boldsymbol{\alpha}}[X]) \leq \mathop{\mathbb{E}}_{\boldsymbol{\alpha}}[f(X)]\,.$$

Extending this to arbitrary distributions can be shown via the continuity of f on any open set, which is guaranteed by the convexity of f, and the weak density of distributions with finite support in the family of all probability measures. □

B.3 Constrained optimization

We now define a general constrained optimization problem and the specific properties associated to convex constrained optimization problems.

Definition B.21 (Constrained optimization problem) *Let* $\mathcal{X} \subseteq \mathbb{R}^N$ *and* $f, g_i \colon \mathcal{X} \to \mathbb{R}$*, for all* $i \in [m]$*. Then, a* constrained optimization problem *has the form:*

$$\begin{aligned} &\min_{\mathbf{x}\in\mathcal{X}} \quad f(\mathbf{x}) \\ &\textit{subject to:} \quad g_i(\mathbf{x}) \leq 0, \ \forall i \in \{1, \ldots, m\}. \end{aligned}$$

This general formulation does not make any convexity assumptions and can be augmented with equality constraints. It is referred to as the *primal problem* in contrast with a related problem introduced later. We will denote by p^* the optimal value of the objective.

For any $\mathbf{x} \in \mathcal{X}$, we will denote by $g(\mathbf{x})$ the vector $(g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))^\top$. Thus, the constraints can be written as $g(\mathbf{x}) \leq \mathbf{0}$. To any constrained optimization problem, we can associate a *Lagrange function* that plays an important role in the analysis of the problem and its relationship with another related optimization problem.

Definition B.22 (Lagrangian) *The* Lagrange function *or the* Lagrangian *associated to the general constrained optimization problem defined in (B.21) is the function defined over* $\mathcal{X} \times \mathbb{R}_+$ *by:*

$$\forall \mathbf{x} \in \mathcal{X}, \forall \boldsymbol{\alpha} \geq 0, \quad \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}) = f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}),$$

where the variables α_i *are known as the* Lagrange *or* dual *variables with* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)^\top$*.*

Any equality constraint of the form $g(\mathbf{x}) = 0$ for a function g can be equivalently expressed by two inequalities: $-g(\mathbf{x}) \leq 0$ and $+g(\mathbf{x}) \leq 0$. Let $\alpha_- \geq 0$ be the Lagrange variable associated to the first constraint and $\alpha_+ \geq 0$ the one associated to the second constraint. The sum of the terms corresponding to these constraints in the definition of the Lagrange function can therefore be written as $\alpha g(\mathbf{x})$ with $\alpha = (\alpha_+ - \alpha_-)$. Thus, in general, for an equality constraint $g(\mathbf{x}) = 0$ the Lagrangian is augmented with a term $\alpha g(\mathbf{x})$ but with $\alpha \in \mathbb{R}$ not constrained to be non-negative. Note that in the case of a convex optimization problem, equality constraints $g(\mathbf{x})$ are required to be affine since both $g(\mathbf{x})$ and $-g(\mathbf{x})$ are required to be convex.

Definition B.23 (Dual function) *The* (Lagrange) dual function *associated to the constrained optimization problem is defined by*

$$\forall \boldsymbol{\alpha} \geq 0, F(\boldsymbol{\alpha}) = \inf_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}) = \inf_{\mathbf{x}\in\mathcal{X}} \left(f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}) \right). \tag{B.5}$$

Note that F is always concave, since the Lagrangian is linear with respect to $\boldsymbol{\alpha}$ and since the infimum preserves concavity. We further observe that

$$\forall \boldsymbol{\alpha} \geq 0, \quad F(\boldsymbol{\alpha}) \leq p^*, \tag{B.6}$$

since for any feasible $\mathbf{x}$, $f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}) \leq f(\mathbf{x})$. The dual function naturally leads to the following optimization problem.

Definition B.24 (Dual problem) *The* dual (optimization) problem *associated to the constrained optimization problem is*

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \quad F(\boldsymbol{\alpha}) \\ &\textit{subject to:} \quad \boldsymbol{\alpha} \geq 0. \end{aligned}$$

The dual problem is always a convex optimization problem (as a maximization of a concave problem). Let d^* denote an optimal value. By (B.6), the following inequality always holds:

$$d^* \leq p^* \qquad (\textit{weak duality}).$$

The difference $(p^* - d^*)$ is known as the *duality gap*. The equality case

$$d^* = p^* \qquad (\textit{strong duality})$$

does not hold in general. However, strong duality does hold when convex problems satisfy a *constraint qualification*. We will denote by $\operatorname{int}(\mathcal{X})$ the interior of the set $\mathcal{X}$.

Definition B.25 (Strong constraint qualification) *Assume that* $\text{int}(\mathcal{X}) \neq \emptyset$. *Then, the* strong constraint qualification *or* Slater's condition *is defined as*

$$\exists\, \overline{\mathbf{x}} \in \text{int}(\mathcal{X})\colon g(\overline{\mathbf{x}}) < 0. \tag{B.7}$$

A function $h\colon \mathcal{X} \to \mathbb{R}$ is said to be *affine* if it can be defined for all $\mathbf{x} \in \mathcal{X}$ by $h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, for some $\mathbf{w} \in \mathbb{R}^N$ and $b \in \mathbb{R}$.

Definition B.26 (Weak constraint qualification) *Assume that* $\text{int}(\mathcal{X}) \neq \emptyset$. *Then, the* weak constraint qualification *or* weak Slater's condition *is defined as*

$$\exists\, \overline{\mathbf{x}} \in \text{int}(\mathcal{X})\colon \forall i \in [m], \big(g_i(\overline{\mathbf{x}}) < 0\big) \vee \big(g_i(\overline{\mathbf{x}}) = 0 \wedge g_i \ \textit{affine}\big). \tag{B.8}$$

We next present sufficient and necessary conditions for solutions to constrained optimization problems, based on the saddle point of the Lagrangian and Slater's condition.

Theorem B.27 (Saddle point — sufficient condition) *Let P be a constrained optimization problem over $\mathcal{X} = \mathbb{R}^N$. If $(\mathbf{x}^*, \boldsymbol{\alpha}^*)$ is a saddle point of the associated Lagrangian, that is,*

$$\forall \mathbf{x} \in \mathbb{R}^N, \forall \boldsymbol{\alpha} \geq 0, \quad \mathcal{L}(\mathbf{x}^*, \boldsymbol{\alpha}) \leq \mathcal{L}(\mathbf{x}^*, \boldsymbol{\alpha}^*) \leq \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}^*), \tag{B.9}$$

then $\mathbf{x}^*$ is a solution of the problem P.

Proof: By the first inequality, the following holds:

$$\begin{aligned} \forall \boldsymbol{\alpha} \geq 0, \mathcal{L}(\mathbf{x}^*, \boldsymbol{\alpha}) \leq \mathcal{L}(\mathbf{x}^*, \boldsymbol{\alpha}^*) &\Rightarrow \forall \boldsymbol{\alpha} \geq 0, \boldsymbol{\alpha} \cdot g(\mathbf{x}^*) \leq \boldsymbol{\alpha}^* \cdot g(\mathbf{x}^*) \\ &\Rightarrow g(\mathbf{x}^*) \leq 0 \wedge \boldsymbol{\alpha}^* \cdot g(\mathbf{x}^*) = 0\,, \end{aligned} \tag{B.10}$$

where $g(\mathbf{x}^*) \leq 0$ in (B.10) follows by letting $\boldsymbol{\alpha} \to +\infty$ and $\boldsymbol{\alpha}^* \cdot g(\mathbf{x}^*) = 0$ follows by letting $\boldsymbol{\alpha} \to 0$. In view of (B.10), the second inequality in (B.9) gives,

$$\forall \mathbf{x}, \mathcal{L}(\mathbf{x}^*, \boldsymbol{\alpha}^*) \leq \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}^*) \Rightarrow \forall \mathbf{x}, f(\mathbf{x}^*) \leq f(\mathbf{x}) + \boldsymbol{\alpha}^* \cdot g(\mathbf{x}).$$

Thus, for all $\mathbf{x}$ satisfying the constraints, that is $g(\mathbf{x}) \leq 0$, we have

$$f(\mathbf{x}^*) \leq f(\mathbf{x}),$$

which completes the proof. □

Theorem B.28 (Saddle point — necessary condition) *Assume that f and g_i, $i \in [m]$, are* convex functions *and that Slater's condition holds. Then, if $\mathbf{x}$ is a solution of the constrained optimization problem, there exists $\boldsymbol{\alpha} \geq 0$ such that $(\mathbf{x}, \boldsymbol{\alpha})$ is a saddle point of the Lagrangian.*

Theorem B.29 (Saddle point — necessary condition) *Assume that f and g_i, $i \in [m]$, are* convex differentiable functions *and that the weak Slater's condition holds. If $\mathbf{x}$ is a solution of the constrained optimization problem, then there exists $\boldsymbol{\alpha} \geq 0$ such that $(\mathbf{x}, \boldsymbol{\alpha})$ is a saddle point of the Lagrangian.*

We conclude with a theorem providing necessary and sufficient optimality conditions when the problem is convex, the objective function differentiable, and the constraints qualified.

Theorem B.30 (Karush-Kuhn-Tucker's theorem) *Assume that $f, g_i\colon \mathcal{X} \to \mathbb{R}, \forall i \in [m]$ are convex and differentiable and that the constraints are qualified. Then $\overline{\mathbf{x}}$ is a solution of the constrained program if and if only there exists $\overline{\boldsymbol{\alpha}} \geq 0$ such that,*

$$\nabla_{\mathbf{x}} \mathcal{L}(\overline{\mathbf{x}}, \overline{\boldsymbol{\alpha}}) = \nabla_{\mathbf{x}} f(\overline{\mathbf{x}}) + \overline{\boldsymbol{\alpha}} \cdot \nabla_{\mathbf{x}} g(\overline{\mathbf{x}}) = 0 \tag{B.11}$$

$$\nabla_{\boldsymbol{\alpha}} \mathcal{L}(\overline{\mathbf{x}}, \overline{\boldsymbol{\alpha}}) = g(\overline{\mathbf{x}}) \leq 0 \tag{B.12}$$

$$\overline{\boldsymbol{\alpha}} \cdot g(\overline{\mathbf{x}}) = \sum_{i=1}^{m} \overline{\boldsymbol{\alpha}}_i g_i(\overline{\mathbf{x}}) = 0\,. \tag{B.13}$$

The conditions B.11–B.13 are known as the KKT conditions. *Note that the last two KKT conditions are equivalent to*

$$g(\overline{\mathbf{x}}) \leq 0 \wedge (\forall i \in \{1, \ldots, m\}, \bar{\alpha}_i g_i(\overline{\mathbf{x}}) = 0). \tag{B.14}$$

These equalities are known as complementarity conditions.

Proof: For the forward direction, since the constraints are qualified, if $\overline{\mathbf{x}}$ is a solution, then there exists $\overline{\boldsymbol{\alpha}}$ such that the $(\overline{\mathbf{x}}, \overline{\boldsymbol{\alpha}})$ is a saddle point of the Lagrangian and all three conditions are satisfied (the first condition follows by definition of a saddle point, and the second two conditions follow from (B.10)).

In the opposite direction, if the conditions are met, then for any $\mathbf{x}$ such that $g(\mathbf{x}) \leq 0$, we can write

$$\begin{aligned} f(\mathbf{x}) - f(\overline{\mathbf{x}}) &\geq \nabla_{\mathbf{x}} f(\overline{\mathbf{x}}) \cdot (\mathbf{x} - \overline{\mathbf{x}}) && \text{(convexity of } f\text{)} \\ &= -\sum_{i=1}^{m} \overline{\boldsymbol{\alpha}}_i \nabla_{\mathbf{x}} g_i(\overline{\mathbf{x}}) \cdot (\mathbf{x} - \overline{\mathbf{x}}) && \text{(first condition)} \\ &\geq -\sum_{i=1}^{m} \overline{\boldsymbol{\alpha}}_i [g_i(\mathbf{x}) - g_i(\overline{\mathbf{x}})] && \text{(convexity of } g_i\text{s)} \\ &= -\sum_{i=1}^{m} \overline{\boldsymbol{\alpha}}_i g_i(\mathbf{x}) \geq 0, && \text{(third condition)} \end{aligned}$$

which shows that $f(\overline{\mathbf{x}})$ is the minimum of f over the set of points satisfying the constraints. □

B.4 Fenchel duality

In this section, we present an alternative theory of convex optimization or convex analysis where the functions f considered may be non-differentiable and take infinite values.

Throughout, this section, the set $\mathcal{X}$ denotes a Hilbert space with the inner product denoted by $\langle \cdot, \cdot \rangle$. However, the results presented can be straightforwardly extended to the case of a Banach space. We consider functions taking values in $[-\infty, +\infty]$. The *domain of a function* $f \colon \mathcal{X} \to [-\infty, +\infty]$ is defined as the set

$$\operatorname{dom}(f) = \{x \in \mathcal{X} \colon f(x) < +\infty\}. \tag{B.15}$$

We extend the definition of convexity and say that $f \colon \mathcal{X} \to [-\infty, +\infty]$ is convex if it is convex over $\operatorname{dom}(f)$, that is if for all $x, x' \in \operatorname{dom}(f)$ and all $t \in [0, 1]$,

$$f(tx + (1-t)x') \leq tu + (1-t)v, \tag{B.16}$$

for all $(u, v) \in \mathbb{R}^2$ with $u \geq f(x)$ and $v \geq f(x')$. A convex function is said to be *proper* if it takes values in $(-\infty, +\infty]$ and if it is not uniformly equal to $+\infty$. It is said to be *closed* when its epigraph is closed.

B.4.1 Subgradients

Definition B.31 *Let $f \colon \mathcal{X} \to (-\infty, +\infty]$ be a convex function. Then, a vector $g \in \mathcal{X}$ is a* subgradient *of f at a point $x \in \operatorname{dom}(f)$ if the following inequality holds for all $z \in \mathcal{X}$:*

$$f(z) \geq f(x) + \langle z - x, g \rangle. \tag{B.17}$$

The set of all subgradients at x is called the subdifferential of f at x *and is denoted by $\partial f(x)$ with $\partial f(x) = \emptyset$ for $x \notin \operatorname{dom}(f)$.*

Thus, g is a subgradient at x iff the hyperplane with normal vector g passing through the point $(x, f(x))$ is below the graph of f, that is iff it is supporting the graph of f. Figure 14.1 illustrates these definitions.

The following lemma shows that if f is differentiable at $x \in \operatorname{dom}(f)$, then its subdifferential is reduced to its gradient at x.

Lemma B.32 *If f is differentiable at $x \in \operatorname{dom}(f)$, then $\partial f(x) = \{\nabla f(x)\}$.*

Proof: Clearly, the gradient $\nabla f(x)$ is always a subgradient at x. Now, let g be in $\partial f(x)$. Then, by definition of a subgradient, for any $\epsilon \in \mathbb{R}$,

$$f(x + \epsilon(\nabla f(x) - g)) \geq f(x) + \epsilon\langle \nabla f(x) - g, g\rangle.$$

A first-order Taylor series expansion gives

$$f(x + \epsilon(\nabla f(x) - g)) - f(x) = \epsilon\langle \nabla f(x), \nabla f(x) - g\rangle + o(\epsilon\|\nabla f(x) - g\|).$$

In view of that, the first inequality can be rewritten as

$$\epsilon\|\nabla f(x) - g\|^2 \leq o(\epsilon\|\nabla f(x) - g\|),$$

which implies $\|\nabla f(x) - g\| = o(1)$ and $\nabla f(x) = g$. □

Proposition B.33 *Let $f\colon \mathcal{X} \to (-\infty, +\infty]$ be a proper function. Then, x^* is a global minimizer of f iff $\partial f(x^*)$ contains 0.*

Proof: Since f is proper, if x^* is a minimizer, $f(x^*)$ cannot be $+\infty$. Thus x^* must be in $\mathrm{dom}(f)$ (and thus $\partial f(x)$ is not defined to be empty). Now, x^* is a global minimizer iff for all $z \in \mathcal{X}$, $f(z) \geq f(x^*)$, that is iff 0 is a subgradient of f at x^*. □

B.4.2 Core

The *core* of a set $\mathcal{C} \subseteq \mathcal{X}$ is denoted by $\mathrm{core}(\mathcal{C})$ and defined as follows:

$$\mathrm{core}(\mathcal{C}) = \{x \in \mathcal{C}\colon \forall u \in \mathcal{X}, \exists \epsilon > 0 \mid \forall t \in [0, \epsilon], (x + tu) \in \mathcal{C}\}. \tag{B.18}$$

Thus, for $x \in \mathrm{core}(\mathcal{C})$, for any direction u, $(x + tu)$ is in $\mathcal{C}$ for t sufficiently small. In view of this definition, $\mathrm{core}(\mathcal{C})$ clearly contains the interior of $\mathcal{C}$, $\mathrm{int}(\mathcal{C})$.

Proposition B.34 *Let $h\colon \mathcal{X} \to [-\infty, +\infty]$ be a convex function. If there exists $x_0 \in \mathrm{core}(\mathrm{dom}(h))$ such that $h(x_0) > -\infty$, then $h(x) > -\infty$ for all $x \in \mathcal{X}$.*

Proof: Let x be in $\mathcal{X}$. Since x_0 is in $\mathrm{core}(\mathrm{dom}(h))$, there exists $t > 0$ such that $x_0' = x_0 + t(x_0 - x)$ is in $\mathrm{dom}(h)$, that is such that $h(x_0') < +\infty$. Since $x_0 = \frac{1}{1+t}x_0' + \frac{t}{1+t}x$, by convexity, the following holds:

$$h(x_0) \leq \frac{1}{1+t}h(x_0') + \frac{t}{1+t}v \iff v \geq \frac{1+t}{t}\Big[h(x_0) - \frac{1}{1+t}h(x_0')\Big] \tag{B.19}$$

for all $v \geq h(x)$. This implies $h(x) \geq \frac{1+t}{t}[h(x_0) - \frac{1}{1+t}h(x_0')] > -\infty$, which concludes the proof. □

The proof of the following result is left as an exercise (Exercise B.3).

Proposition B.35 *Let $h\colon \mathcal{X} \to (-\infty, +\infty]$ be a convex function. Then, h admits a subdifferential at any $x \in \mathrm{core}(\mathrm{dom}(h))$.*

B.4.3 Conjugate functions

Definition B.36 (Conjugate functions) *Let $\mathcal{X}$ be a Hilbert space. The* conjugate function *or* Fenchel conjugate *of a function $f\colon \mathcal{X} \mapsto [-\infty, +\infty]$ is the function $f^*\colon \mathcal{X} \mapsto [-\infty, +\infty]$ defined by*

$$f^*(u) = \sup_{x \in \mathcal{X}} \big\{\langle u, x\rangle - f(x)\big\}. \tag{B.20}$$

Note that f^* is convex as the pointwise supremum of the set of affine and thus convex functions $u \mapsto \langle x, u\rangle - f(x)$. Also, if there exists x such that $f(x) < +\infty$, then $f > -\infty$. Conjugation is order-reversing: for any f and g, if $f \leq g$, then $g^* \leq f^*$. Also, it is straightforward to see that if f is closed proper convex, then $f^{**} = f$.

Figure B.3 illustrates the definition of conjugate functions. As shown by the figure, conjugate functions correspond to a dual description of the epigraph of a function in terms of supporting hyperplanes and their crossing points.

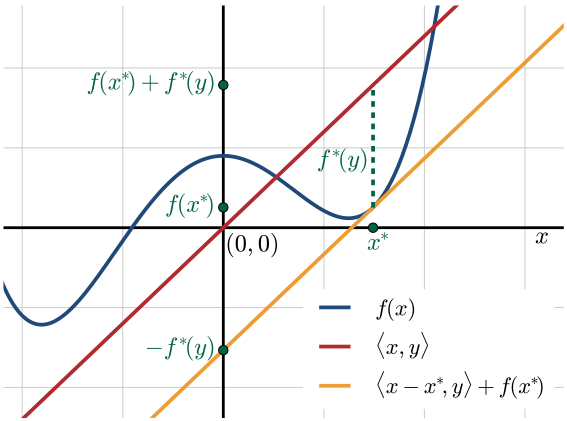


Figure B.3
Illustration of the conjugate f^* of a function f. Given y, x^* is the point at which the distance between the hyperplane of equation $z = \langle x, y\rangle$ with normal y (slope y in dimension one) and the plot of $f(x)$ is the largest. This largest distance is equal to $f^*(y)$. The parallel hyperplane $z = \langle x - x^*, y\rangle + f(x^*)$ with normal y and passing through the point $(x^*, f(x^*)$ is shown. This is a supporting hyperplane of the plot of $f(x)$. The point at which it intercepts the y-axis (crossing point) has y-coordinate $-f^*(y)$.

Lemma B.37 (Conjugate of extended relative entropy) *Let $p_0 \in \Delta$ be a distribution over $\mathcal{X}$ such that $p_0(x) > 0$ for all $x \in \mathcal{X}$. Define $f\colon \mathbb{R}^{\mathcal{X}} \to \mathbb{R}$ by*

$$f(p) = \begin{cases} \mathsf{D}(p\|p_0) & \text{if } p \in \Delta \\ +\infty & \text{otherwise.} \end{cases}$$

Then, the conjugate function $f^*\colon \mathbb{R}^{\mathcal{X}} \to \mathbb{R}$ of f is defined by

$$\forall q \in \mathbb{R}^{\mathcal{X}},\ f^*(q) = \log\left(\sum_{x\in\mathcal{X}} p_0(x) e^{q(x)}\right).$$

Proof: By definition of f, for any $q \in \mathbb{R}^{\mathcal{X}}$, we can write

$$\sup_{p\in\mathbb{R}^{\mathcal{X}}} \big(\langle p, q\rangle - \mathsf{D}(p\|p_0)\big) = \sup_{p\in\Delta} \big(\langle p, q\rangle - \mathsf{D}(p\|p_0)\big). \tag{B.21}$$

Fix $q \in \mathbb{R}^{\mathcal{X}}$ and let $\bar{q} \in \Delta$ be defined for all $x \in \mathcal{X}$ by

$$\bar{q}(x) = \frac{p_0(x)e^{q(x)}}{\sum_{x\in\mathcal{X}} p_0(x)e^{q(x)}} = \frac{p_0(x)e^{q(x)}}{\mathbb{E}_{p_0}[e^q]}. \tag{B.22}$$

Then, the following holds for all $p \in \Delta$:

$$\langle p, q\rangle - \mathsf{D}(p\|p_0) = \mathop{\mathbb{E}}_{p}[\log(e^q)] - \mathop{\mathbb{E}}_{p}\left[\log\frac{p}{p_0}\right] = \mathop{\mathbb{E}}_{p}\left[\log\frac{p_0 e^q}{p}\right] = -\mathsf{D}(p\|\bar{q}) + \log \mathop{\mathbb{E}}_{p_0}[e^q].$$

Since $\mathsf{D}(p\|\bar{q}) \geq 0$ and $\mathsf{D}(p\|\bar{q}) = 0$ for $p = \bar{q}$, this shows that $\sup_{p\in\Delta}\big(p\cdot q - \mathsf{D}(p\|p_0)\big) = \log\big(\mathbb{E}_{p_0}[e^q]\big)$ and concludes the proof. □

Table B.1 gives a series of other examples of functions and their conjugates. The following is an immediate consequence of the definition of the conjugate functions.

Table B.1
Examples of functions g and their conjugates g^*.

$g(x)$	$\mathbf{dom}(g)$	$g^*(y)$	$\mathbf{dom}(g^*)$
$f(ax)\ (a \neq 0)$	$\mathcal{X}$	$f^*\left(\frac{y}{a}\right)$	$\mathcal{X}^*$
$f(x+b)$	$\mathcal{X}$	$f^*(y) - \langle b, y\rangle$	$\mathcal{X}^*$
$af(x)\ (a > 0)$	$\mathcal{X}$	$af^*\left(\frac{y}{a}\right)$	$\mathcal{X}^*$
$\alpha x + \beta$	$\mathbb{R}$	$\begin{cases} -\beta & \text{if } y = \alpha \\ +\infty & \text{otherwise} \end{cases}$	$\mathbb{R}$
$\frac{\lvert x\rvert^p}{p}\ (p > 1)$	$\mathbb{R}$	$\frac{\lvert y\rvert^q}{q}\ (\frac{1}{p} + \frac{1}{q} = 1)$	$\mathbb{R}$
$\frac{-x^p}{p}\ (0 < p < 1)$	$\mathbb{R}_+$	$\frac{-(-y)^q}{q}\ (\frac{1}{p} + \frac{1}{q} = 1)$	$\mathbb{R}_-$
$\sqrt{1+x^2}$	$\mathbb{R}$	$-\sqrt{1-(y)^2}$	$[-1, 1]$
$-\log(x)$	$\mathbb{R}_+ \setminus \{0\}$	$-(1+\log(-y))$	$\mathbb{R}_- \setminus \{0\}$
e^x	$\mathbb{R}$	$\begin{cases} y\log(y) - y\,, & \text{if } y > 0 \\ 0\,, & \text{if } y = 0 \end{cases}$	$\mathbb{R}_+$
$\log\left(1+e^x\right)$	$\mathbb{R}$	$\begin{cases} y\log(y) + (1-y)\log(1-y)\,, & \text{if } 0 < y < 1 \\ 0\,, & \text{if } y = 0, 1 \end{cases}$	$[0, 1]$
$-\log\left(1-e^x\right)$	$\mathbb{R}$	$\begin{cases} y\log(y) - (1+y)\log(1+y)\,, & \text{if } y > 0 \\ 0\,, & \text{if } y = 0 \end{cases}$	$\mathbb{R}_+$

Proposition B.38 (Fenchel's inequality) *Let $\mathcal{X}$ be a Hilbert space. For any function $f\colon \mathcal{X} \mapsto [-\infty, +\infty]$ and any $x \in \mathrm{dom}(f)$ and $u \in \mathcal{X}$, the following inequality holds:*

$$f(x) + f^*(u) \geq \langle u, x\rangle\,. \tag{B.23}$$

Equality holds iff u is a subgradient of f at x.

We will denote by A^* the adjoint operator of a bounded (or continuous) linear map $A\colon \mathcal{X} \to \mathcal{Y}$. Also, we denote by $\mathrm{cont}(f)$ the set of points $x \in \mathcal{X}$ at which $f\colon \mathcal{X} \to [-\infty, +\infty]$ is finite and continuous.

Theorem B.39 (Fenchel duality theorem) *Let $\mathcal{X}$ and $\mathcal{Y}$ be two Hilbert spaces, $f\colon \mathcal{X} \to (-\infty, +\infty]$ and $g\colon \mathcal{Y} \to (-\infty, +\infty]$ two convex functions and $A\colon \mathcal{X} \to \mathcal{Y}$ a bounded linear map. Then, the following two optimization problems (*Fenchel problems*)*

$$p^* = \inf_{x\in\mathcal{X}} \{f(x) + g(Ax)\}$$
$$d^* = \sup_{y\in\mathcal{Y}} \{-f^*(A^*y) - g^*(-y)\}$$

satisfy the weak duality $p^* \geq d^*$. *If further f and g satisfy the condition*

$$0 \in \mathrm{core}\left(\mathrm{dom}(g) - A(\mathrm{dom}(f))\right),$$

or the stronger condition

$$A(\mathrm{dom}(f)) \cap \mathrm{cont}(g) \neq \emptyset,$$

then strong duality *holds, that is $p^* = d^*$ and the supremum in the dual problem is attained if $d^* \in \mathbb{R}$.*

Proof: By Fenchel's inequality (proposition B.38) applied to both f and g, for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, the following inequalities hold:

$$f(x) + f^*(A^*y) \geq \langle A^*y, x\rangle = \langle y, Ax\rangle = -\langle -y, Ax\rangle \geq -g(Ax) - g^*(-y).$$

Comparing the leftmost and the rightmost terms gives

$$f(x) + f^*(A^*y) \geq -g(Ax) - g^*(-y) \iff f(x) + g(Ax) \geq -f^*(A^*y) - g^*(-y).$$

Taking the infimum over $x \in \mathcal{X}$ of the left-hand side and the supremum over $y \in \mathcal{Y}$ of the right-hand side of the last inequality yields $p^* \geq d^*$.

Consider now the function $h\colon \mathcal{Y} \to [-\infty, +\infty]$ defined for all $u \in \mathcal{Y}$ by

$$h(u) = \inf_{x\in\mathcal{X}} \{f(x) + g(Ax + u)\}. \tag{B.24}$$

Since $(x, u) \mapsto f(x) + g(Ax + u)$ is convex, h is convex as the infimum over one argument of that function. u is in $\operatorname{dom}(h)$ iff there exists $x \in \mathcal{X}$ such that $f(x) + g(Ax + u) < +\infty$, that is iff there exists $x \in \mathcal{X}$ such that $f(x) < +\infty$ and $g(Ax + u) < +\infty$, that is iff there exists $x \in \operatorname{dom}(f)$ such that $(Ax + u) \in \operatorname{dom}(g)$. Thus, we have $\operatorname{dom}(h) = \operatorname{dom}(g) - A\operatorname{dom}(f)$.

If $p^* = -\infty$, then strong duality clearly holds. Otherwise, $p^* > -\infty$. If $0 \in \operatorname{core}\big(\operatorname{dom}(g) - A(\operatorname{dom}(f))\big) = \operatorname{core}(\operatorname{dom}(h))$, then 0 is in $\operatorname{dom}(h)$ and $p^* < +\infty$. Thus, $p^* = h(0)$ is in $\mathbb{R}$. By proposition B.34, since $h(0) > -\infty$ and $0 \in \operatorname{core}(\operatorname{dom}(h))$, h takes values in $(-\infty, +\infty]$. Thus, by proposition B.35, h admits a subgradient $-y$ at 0. By definition of y, for all $x \in \mathcal{X}$ and $u \in \mathcal{Y}$,

$$\begin{aligned} h(0) &\leq h(u) + \langle y, u\rangle \\ &\leq f(x) + g(Ax + u) + \langle y, u\rangle \\ &= \{f(x) - \langle A^*y, x\rangle\} + \{g(Ax + u) + \langle y, u\rangle + \langle A^*y, x\rangle\} \\ &= \{f(x) - \langle A^*y, x\rangle\} + \{g(Ax + u) + \langle y, Ax + u\rangle\}. \end{aligned}$$

Taking the infimum over u and the supremum over x yields

$$h(0) \leq -f^*(A^*y) - g^*(-y) \leq d^* \leq p^* = h(0),$$

which proves $d^* = p^*$ and that the supremum defining d^* is reached at y.

Finally, assume that $A(\operatorname{dom}(f)) \cap \operatorname{cont}(g) \neq \emptyset$ and let $u \in A(\operatorname{dom}(f)) \cap \operatorname{cont}(g)$. Then, $u = Ax$ with $x \in \operatorname{dom}(f)$ and $u \in \operatorname{cont}(g) \subseteq \operatorname{dom}(g)$. Thus, we have $0 = u - Ax \in \operatorname{dom}(g) - A\operatorname{dom}(f)$. Since g is continuous at u and $g(u)$ is finite, for any $v \in \mathcal{X}$, there exists $\epsilon > 0$ such that $g(u + tv)$ is finite for all $t \in [0, \epsilon]$, thus $w_t = (u + tv) \in \operatorname{dom}(g)$. Therefore, for any $t \in [0, \epsilon]$, $tv = w_t - u = w_t - Ax \in \operatorname{dom}(g) - A\operatorname{dom}(f)$, which shows that $0 \in \operatorname{core}\big(\operatorname{dom}(g) - A(\operatorname{dom}(f))\big)$. □

To illustrate the theorem, consider the case where A is the identity operator. The primal optimization problem is then $\min_x \big\{f(x) + g(x)\big\}$. Figure B.4 illustrates the Fenchel duality theorem in that case. The primal problem consists of finding the point x^* at which the distance between the plots of $f(x)$ and $-g(x)$ is minimal since $f(x) + g(x) = f(x) - (-g(x))$. As shown by the figure, under the conditions of the theorem, this coincides with seeking y^* at which the difference of the conjugate values of $f(x)$ and $-g(x)$, that is the difference $-f^*(y) - g^*(-y)$ is maximal.

B.5 Chapter notes

The results presented in this appendix are based on several key theorems: theorem B.3 due to Fermat (1629); theorem B.27 due to Lagrange (1797), theorem B.30 due to Karush [1939] and Kuhn and Tucker [1951], and theorem B.39 due to Werner Fenchel, based on the notion of conjugate functions or Legendre transformations. For a more extensive material on convex optimization, we strongly recommend the books of Boyd and Vandenberghe [2004], Bertsekas, Nedić, and Ozdaglar [2003], Rockafellar [1997], Borwein and Lewis [2000] and Borwein and Zhu [2005] which have formed the basis for part of the material presented in this appendix. In particular, our table of conjugate functions is extracted from [Borwein and Lewis, 2000].

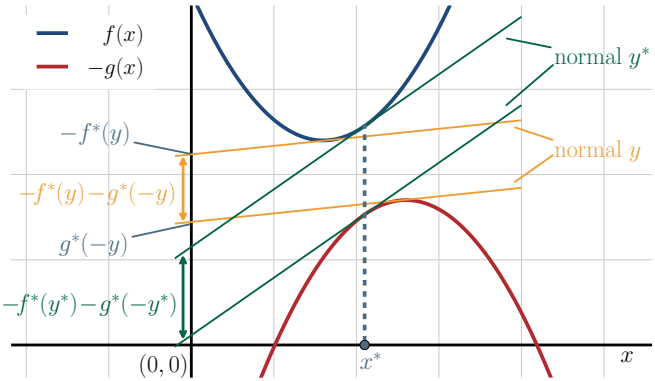


Figure B.4
Illustration of Fenchel Duality; $\min_x \{f(x) + g(x)\} = \max_y \{ - f^*(y) - g^*(-y)\}$.

B.6 Exercises

B.1 Give the conjugate of the function f defined by $f(x) = |x|$ for all $x \in \mathbb{R}$.

B.2 Prove the correctness of the conjugate function g^* for each function g of Table B.1.

B.3 Give the proof of Proposition B.35.

C Probability Review

In this appendix, we give a brief review of some basic notions of probability and will also define the notation that is used throughout the textbook.

C.1 Probability

A *probability space* is a tuple consisting of three components: a *sample space*, an *events set*, and a *probability distribution*:

- *sample space* Ω: Ω is the set of all elementary events or outcomes possible in a trial, for example, each of the six outcomes in $\{1, \ldots, 6\}$ when casting a die.
- *events set* $\mathcal{F}$: $\mathcal{F}$ is a *σ-algebra*, that is a set of subsets of Ω containing Ω that is closed under complementation and countable union (therefore also countable intersection). An example of an event may be "the die lands on an odd number".
- *probability distribution*: $\mathbb{P}$ is a mapping from the set of all events $\mathcal{F}$ to $[0, 1]$ such that $\mathbb{P}[\Omega] = 1$, $\mathbb{P}[\emptyset] = 1$, and, for all mutually exclusive events $A_1, \ldots, A_n$,
$$\mathbb{P}[A_1 \cup \ldots \cup A_n] = \sum_{i=1}^{n} \mathbb{P}[A_i].$$
The discrete probability distribution associated with a fair die can be defined by $\mathbb{P}[A_i] = 1/6$ for $i \in \{1 \ldots 6\}$, where A_i is the event that the die lands on value i.

C.2 Random variables

Definition C.1 (Random variables) *A* random variable X *is a function* $X \colon \Omega \to \mathbb{R}$ *that is* measurable*, that is such that for any interval* I*, the subset of the sample space* $\{\omega \in \Omega \colon X(\omega) \in I\}$ *is an event.*

The *probability mass function* of a discrete random variable X is defined as the function $x \mapsto \mathbb{P}[X = x]$. The *joint probability mass function* of discrete random variables X and Y is defined as the function $(x, y) \mapsto \mathbb{P}[X = x \wedge Y = y]$.

A probability distribution is said to be *absolutely continuous* when it admits a *probability density function*, that is a function f associated to a real-valued random variable X that satisfies for all $a, b \in \mathbb{R}$
$$\mathbb{P}[a \leq X \leq b] = \int_a^b f(x)dx\,. \tag{C.1}$$

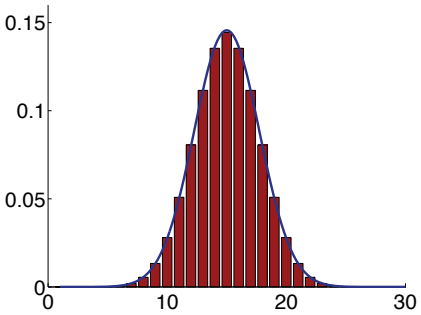


Figure C.1
Approximation of the binomial distribution (in red) by a normal distribution (in blue).

Definition C.2 (Binomial distribution) *A random variable X is said to follow a* binomial distribution $B(n,p)$ *with $n \in \mathbb{N}$ and $p \in [0,1]$ if for any $k \in \{0,1,\ldots,n\}$,*

$$\mathbb{P}[X = k] = \binom{n}{k} p^k (1-p)^{n-k} .$$

Definition C.3 (Normal distribution) *A random variable X is said to follow a* normal *(or* Gaussian*)* distribution $N(\mu, \sigma^2)$ *with $\mu \in \mathbb{R}$ and $\sigma > 0$ if its probability density function is given by,*

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

The standard normal *distribution $N(0,1)$ is the normal distribution with zero mean and unit variance.*

The normal distribution is often used to approximate a binomial distribution. Figure C.1 illustrates that approximation.

Definition C.4 (Laplace distribution) *A random variable X is said to follow a* Laplace distribution *with location parameter $\mu \in \mathbb{R}$ and scale parameter $b > 0$ if its probability density function is given by,*

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right).$$

Definition C.5 (Gibbs distributions) *Given a set $\mathcal{X}$ and* feature function *$\Phi \colon \mathcal{X} \to \mathbb{R}^N$, a random variable X is said to follow a* Gibbs distribution *with parameter $\mathbf{w} \in \mathbb{R}^N$ if for any $x \in \mathcal{X}$,*

$$\mathbb{P}[X = x] = \frac{\exp(\mathbf{w} \cdot \Phi(x))}{\sum_{x \in \mathcal{X}} \exp(\mathbf{w} \cdot \Phi(x))} .$$

The normalizing quantity in the denominator $Z = \sum_{x \in \mathcal{X}} \exp(\mathbf{w} \cdot \Phi(x))$ is also called the partition function*.*

Definition C.6 (Poisson distribution) *A random variable X is said to follow a* Poisson distribution *with $\lambda > 0$ if for any $k \in \mathbb{N}$,*

$$\mathbb{P}[X = k] = \frac{\lambda^k e^{-\lambda}}{k!} .$$

The definition of the following family of distributions uses the notion of independence of random variables defined in the next section.

Definition C.7 (χ^2 distribution) *The* χ^2 distribution *(or* chi-squared distribution*) with k* degrees of freedom *is the distribution of the sum of the squares of k independent random variables, each following a standard normal distribution.*

C.3 Conditional probability and independence

Definition C.8 (Conditional probability) *The* conditional probability *of event A given event B is defined by*

$$\mathbb{P}[A \mid B] = \frac{\mathbb{P}[A \cap B]}{\mathbb{P}[B]}, \tag{C.2}$$

when $\mathbb{P}[B] \neq 0$.

Definition C.9 (Independence) *Two events A and B are said to be* independent *if*

$$\mathbb{P}[A \cap B] = \mathbb{P}[A]\,\mathbb{P}[B]. \tag{C.3}$$

Equivalently, A and B are independent iff $\mathbb{P}[A \mid B] = \mathbb{P}[A]$ when $\mathbb{P}[B] \neq 0$.

A sequence of random variables is said to be *independent and identically distributed (i.i.d.)* when the random variables are mutually independent and follow the same distribution.

The following are basic probability formulae related to the notion of conditional probability. They hold for any events A, B, and $A_1, \ldots, A_n$, with the additional constraint $\mathbb{P}[B] \neq 0$ needed for the Bayes formula to be well defined:

$$\mathbb{P}[A \cup B] = \mathbb{P}[A] + \mathbb{P}[B] - \mathbb{P}[A \cap B] \qquad (\textit{sum rule}) \tag{C.4}$$

$$\mathbb{P}[\bigcup_{i=1}^{n} A_i] \leq \sum_{i=1}^{n} \mathbb{P}[A_i] \qquad (\textit{union bound}) \tag{C.5}$$

$$\mathbb{P}[A \mid B] = \frac{\mathbb{P}[B \mid A]\,\mathbb{P}[A]}{\mathbb{P}[B]} \qquad (\textit{Bayes formula}) \tag{C.6}$$

$$\mathbb{P}[\bigcap_{i=1}^{n} A_i] = \mathbb{P}[A_1]\,\mathbb{P}[A_2 \mid A_1] \cdots \mathbb{P}[A_n \mid \bigcap_{i=1}^{n-1} A_i] \qquad (\textit{chain rule}). \tag{C.7}$$

The sum rule follows immediately from the decomposition of $A \cup B$ as the union of the disjoint sets A and $(B - A \cap B)$. The union bound is a direct consequence of the sum rule. The Bayes formula follows immediately from the definition of conditional probability and the observation that: $\mathbb{P}[A|B]\,\mathbb{P}[B] = \mathbb{P}[B|A]\,\mathbb{P}[A] = \mathbb{P}[A \cap B]$. Similarly, the chain rule follows the observation that $\mathbb{P}[A_1]\,\mathbb{P}[A_2|A_1] = \mathbb{P}[A_1 \cap A_2]$; using the same argument shows recursively that the product of the first k terms of the right-hand side equals $\mathbb{P}[\bigcap_{i=1}^{k} A_i]$.

Finally, assume that $\Omega = A_1 \cup A_2 \cup \ldots \cup A_n$ with $A_i \cap A_j = \emptyset$ for $i \neq j$, i.e., the A_is are mutually disjoint. Then, the following formula is valid for any event B:

$$\mathbb{P}[B] = \sum_{i=1}^{n} \mathbb{P}[B \mid A_i]\,\mathbb{P}[A_i] \qquad (\textit{theorem of total probability}). \tag{C.8}$$

This follows the observation that $\mathbb{P}[B \mid A_i]\,\mathbb{P}[A_i] = \mathbb{P}[B \cap A_i]$ by definition of the conditional probability and the fact that the events $B \cap A_i$ are mutually disjoint.

C.4 Expectation and Markov's inequality

Definition C.10 (Expectation) *The* expectation *or* mean *of a random variable X is denoted by $\mathbb{E}[X]$ and defined by*

$$\mathbb{E}[X] = \sum_{x} x\,\mathbb{P}[X = x]. \tag{C.9}$$

When X follows a probability distribution $\mathcal{D}$, we will also write $\mathbb{E}_{x \sim \mathcal{D}}[x]$ instead of $\mathbb{E}[X]$ to explicitly indicate the distribution. A fundamental property of expectation, which is straightforward to verify using its definition, is that it is linear, that is, for any two random variables X and Y and any $a, b \in \mathbb{R}$, the following holds:

$$\mathbb{E}[aX + bY] = a\,\mathbb{E}[X] + b\,\mathbb{E}[Y]. \tag{C.10}$$

Furthermore, when X and Y are independent random variables, then the following identity holds:

$$\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y]. \tag{C.11}$$

Indeed, by definition of expectation and of independence, we can write

$$\begin{aligned}\mathbb{E}[XY] = \sum_{x,y} xy\,\mathbb{P}[X = x \wedge Y = y] &= \sum_{x,y} xy\,\mathbb{P}[X = x]\,\mathbb{P}[Y = y] \\ &= \Big(\sum_x x\,\mathbb{P}[X = x]\Big)\Big(\sum_y y\,\mathbb{P}[Y = y]\Big),\end{aligned}$$

where in the last step we used Fubini's theorem . The following provides a simple bound for a non-negative random variable in terms of its expectation, known as *Markov's inequality*.

Theorem C.11 (Markov's inequality) *Let X be a non-negative random variable with $\mathbb{E}[X] < \infty$. Then for all $t > 0$,*

$$\mathbb{P}\left[X \geq t\,\mathbb{E}[X]\right] \leq \frac{1}{t}. \tag{C.12}$$

Proof: The proof steps are as follows:

$$\begin{aligned}\mathbb{P}[X \geq t\,\mathbb{E}[X]] &= \sum_{x \geq t\,\mathbb{E}[X]} \mathbb{P}[X = x] && \text{(by definition)} \\ &\leq \sum_{x \geq t\,\mathbb{E}[X]} \mathbb{P}[X = x]\frac{x}{t\,\mathbb{E}[X]} && \Big(\text{using } \frac{x}{t\,\mathbb{E}[X]} \geq 1\Big) \\ &\leq \sum_x \mathbb{P}[X = x]\frac{x}{t\,\mathbb{E}[X]} && \text{(extending non-negative sum)} \\ &= \mathbb{E}\left[\frac{X}{t\,\mathbb{E}[X]}\right] = \frac{1}{t} && \text{(linearity of expectation).}\end{aligned}$$

This concludes the proof. □

C.5 Variance and Chebyshev's inequality

Definition C.12 (Variance — Standard deviation) *The* variance *of a random variable X is denoted by* $\mathrm{Var}[X]$ *and defined by*

$$\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]. \tag{C.13}$$

The standard deviation *of a random variable X is denoted by σ_X and defined by*

$$\sigma_X = \sqrt{\mathrm{Var}[X]}. \tag{C.14}$$

For any random variable X and any $a \in \mathbb{R}$, the following basic properties hold for the variance, which can be proven straightforwardly:

$$\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \tag{C.15}$$

$$\mathrm{Var}[aX] = a^2\,\mathrm{Var}[X]. \tag{C.16}$$

Furthermore, when X and Y are independent, then

$$\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]. \tag{C.17}$$

Indeed, using the linearity of expectation and the identity $\mathbb{E}[X]\,\mathbb{E}[Y] - \mathbb{E}[XY] = 0$ which holds by the independence of X and Y, we can write

$$\begin{aligned}\mathrm{Var}[X + Y] &= \mathbb{E}[(X + Y)^2] - \mathbb{E}[X + Y]^2 \\ &= \mathbb{E}[X^2 + Y^2 + 2XY] - (\mathbb{E}[X]^2 + \mathbb{E}[Y]^2 + 2\,\mathbb{E}[XY]) \\ &= (\mathbb{E}[X^2] - \mathbb{E}[X]^2) + (\mathbb{E}[Y^2] - \mathbb{E}[Y]^2) + 2(\mathbb{E}[X]\,\mathbb{E}[Y] - \mathbb{E}[XY]) \\ &= \mathrm{Var}[X] + \mathrm{Var}[Y].\end{aligned}$$

The following inequality known as *Chebyshev's inequality* bounds the deviation of a random variable from its expectation in terms of its standard deviation.

Theorem C.13 (Chebyshev's inequality) *Let X be a random variable with $\mathrm{Var}[X] < +\infty$. Then, for all $t > 0$, the following inequality holds:*

$$\mathbb{P}\left[|X - \mathbb{E}[X]| \geq t\sigma_X\right] \leq \frac{1}{t^2}. \tag{C.18}$$

Proof: Observe that:

$$\mathbb{P}\left[|X - \mathbb{E}[X]| \geq t\sigma_X\right] = \mathbb{P}[(X - \mathbb{E}[X])^2 \geq t^2\sigma_X^2].$$

The result follows by application of Markov's inequality to $(X - \mathbb{E}[X])^2$. □

We will use Chebyshev's inequality to prove the following theorem.

Theorem C.14 (Weak law of large numbers) *Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of independent random variables with the same mean μ and variance $\sigma^2 < \infty$. Let $\overline{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$, then, for any $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}[|\overline{X}_n - \mu| \geq \epsilon] = 0. \tag{C.19}$$

Proof: Since the variables are independent, we can write

$$\mathrm{Var}[\overline{X}_n] = \sum_{i=1}^n \mathrm{Var}\left[\frac{X_i}{n}\right] = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n}.$$

Thus, by Chebyshev's inequality (with $t = \epsilon/(\mathrm{Var}[\overline{X}_n])^{1/2}$), the following holds:

$$\mathbb{P}[|\overline{X}_n - \mu| \geq \epsilon] \leq \frac{\sigma^2}{n\epsilon^2},$$

which implies (C.19). □

Example C.15 (Applying Chebyshev's inequality) Suppose we roll a pair of fair dice n times. Can we give a good estimate of the total value of the n rolls? If we compute the mean and variance, we find $\mu = 7n$ and $\sigma^2 = 35/6n$ (we leave it to the reader to verify these expressions). Thus, applying Chebyshev's inequality, we see that the final sum will lie within $7n \pm 10\sqrt{\frac{35}{6}n}$ in at least 99 percent of all experiments. Therefore, the odds are better than 99 to 1 that the sum will be between 6.975M and 7.025M after 1M rolls.

Definition C.16 (Covariance) *The* covariance *of two random variables X and Y is denoted by* $\mathrm{Cov}(X, Y)$ *and defined by*

$$\mathrm{Cov}(X, Y) = \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\right]. \tag{C.20}$$

Two random variables X and Y are said to be *uncorrelated* when $\mathrm{Cov}(X, Y) = 0$. It is straightforward to see that if two random variables X and Y are independent then they are uncorrelated, but the converse does not hold in general. The covariance defines a positive semidefinite and symmetric bilinear form:

- symmetry: $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$ for any two random variables X and Y;
- bilinearity: $\mathrm{Cov}(X + X', Y) = \mathrm{Cov}(X, Y) + \mathrm{Cov}(X', Y)$ and $\mathrm{Cov}(aX, Y) = a\,\mathrm{Cov}(X, Y)$ for any random variables X, X', and Y and $a \in \mathbb{R}$;
- positive semidefiniteness: $\mathrm{Cov}(X, X) = \mathrm{Var}[X] \geq 0$ for any random variable X.

The following Cauchy-Schwarz inequality holds for random variables X and Y with $\mathrm{Var}[X] < +\infty$ and $\mathrm{Var}[Y] < +\infty$:

$$|\,\mathrm{Cov}(X, Y)| \leq \sqrt{\mathrm{Var}[X]\,\mathrm{Var}[Y]}. \tag{C.21}$$

The following definition extends the notion of covariance to a vector of random variables.

Definition C.17 *The* covariance matrix *of a vector of random variables* $\mathbf{X} = (X_1, \ldots, X_N)$ *is the matrix in* $\mathbb{R}^{N\times N}$ *denoted by* $\mathbf{C}(\mathbf{X})$ *and defined by*

$$\mathbf{C}(\mathbf{X}) = \mathbb{E}\left[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^\top\right]. \tag{C.22}$$

Thus, $\mathbf{C}(\mathbf{X}) = (\mathrm{Cov}(X_i, X_j))_{ij}$. It is straightforward to show that

$$\mathbf{C}(\mathbf{X}) = \mathbb{E}[\mathbf{X}\mathbf{X}^\top] - \mathbb{E}[\mathbf{X}]\,\mathbb{E}[\mathbf{X}]^\top. \tag{C.23}$$

We close this appendix with the following well-known theorem of probability.

Theorem C.18 (Central limit theorem) *Let* $X_1, \ldots, X_n$ *be a sequence of i.i.d. random variables with mean* μ *and standard deviation* σ*. Let* $\overline{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$ *and* $\overline{\sigma}_n^2 = \sigma^2/n$*. Then,* $(\overline{X}_n - \mu)/\overline{\sigma}_n$ converges to the $N(0,1)$ in distribution*, that is for any* $t \in \mathbb{R}$*,*

$$\lim_{n\to\infty} \mathbb{P}[(\overline{X}_n - \mu)/\overline{\sigma}_n \leq t] = \int_{-\infty}^t \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, dx\,.$$

C.6 Moment-generating functions

The expectation $\mathbb{E}[X^p]$ is called the *pth-moment* of the random variable X. The moment-generating function of a random variable X is a key function from which its different moments can be straightforwardly computed via differentiation at zero. It can therefore be crucial for specifying the distribution of X or analyzing its properties.

Definition C.19 (Moment-generating function) *The* moment-generating function *of a random variable* X *is the function* $M_X : t \mapsto \mathbb{E}[e^{tX}]$ *defined over the set of* $t \in \mathbb{R}$ *for which the expectation is finite.*

If M_X is differentiable at zero, then the pth-moment of X is given by $\mathbb{E}[X^p] = M_X^{(p)}(0)$. We will present in the next chapter a general bound on the moment-generating function of a zero-mean bounded random variable (Lemma D.1). Here, we illustrate its computation in two special cases.

Example C.20 (Standard normal distribution) Let X be a random variable following a normal distribution with mean 0 and variance 1. Then, M_X is defined for all $t \in \mathbb{R}$ by

$$M_X(t) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} e^{tx} dx = e^{\frac{t^2}{2}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-t)^2} dx = e^{\frac{t^2}{2}}\,, \tag{C.24}$$

by recognizing that the last integrand is the probability density function of a normal distribution with mean t and variance 1.

Example C.21 (χ^2 distribution) Let X be a random variable following a χ^2 distribution with k degrees of freedom. We can write $X = \sum_{i=1}^k X_i^2$ where the X_is are independent and follow a standard normal distribution.

Let $t < 1/2$. By the i.i.d. assumption about the variables X_i, we can write

$$\mathbb{E}[e^{tX}] = \mathbb{E}\left[\prod_{i=1}^k e^{tX_i^2}\right] = \prod_{i=1}^k \mathbb{E}\left[e^{tX_i^2}\right] = \mathbb{E}\left[e^{tX_1^2}\right]^k.$$

By definition of the standard normal distribution, we have

$$\begin{aligned}\mathbb{E}[e^{tX_1^2}] &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{tx^2} e^{\frac{-x^2}{2}} dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{(1-2t)\frac{-x^2}{2}} dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{e^{\frac{-u^2}{2}}}{\sqrt{1-2t}} du = (1-2t)^{-\frac{1}{2}}\,,\end{aligned}$$

where we used the change of variable $u = \sqrt{1-2t}\,x$. In view of that, the moment-generating function of the χ^2 distribution is given by

$$\forall t < 1/2,\ M_X(t) = \mathbb{E}[e^{tX}] = (1-2t)^{-\frac{k}{2}}\,. \tag{C.25}$$

C.7 Exercises

C.1 Let $f\colon (0,+\infty) \to \mathbb{R}_+$ be a function admitting an inverse f^{-1} and let X be a random variable. Show that if for any $t > 0$, $\mathbb{P}[X > t] \leq f(t)$, then, for any $\delta > 0$, with probability at least $1-\delta$, $X \leq f^{-1}(\delta)$.

C.2 Let X be a discrete random variable taking non-negative integer values. Show that $\mathbb{E}[X] = \sum_{n\geq 1} \mathbb{P}[X \geq n]$ (*Hint*: rewrite $\mathbb{P}[X = n]$ as $\mathbb{P}[X \geq n] - \mathbb{P}[X \geq n+1]$).

D Concentration Inequalities

In this appendix, we present several *concentration inequalities* used in the proofs given in this book. Concentration inequalities give probability bounds for a random variable to be concentrated around its mean, or for it to deviate from its mean or some other value.

D.1 Hoeffding's inequality

We first present Hoeffding's inequality, whose proof makes use of the general *Chernoff bounding technique*. Given a random variable X and $\epsilon > 0$, this technique consists of proceeding as follows to bound $\mathbb{P}[X \geq \epsilon]$. For any $t > 0$, first Markov's inequality is used to bound $\mathbb{P}[X \geq \epsilon]$:

$$\mathbb{P}[X \geq \epsilon] = \mathbb{P}[e^{tX} \geq e^{t\epsilon}] \leq e^{-t\epsilon}\, \mathbb{E}[e^{tX}]\,. \tag{D.1}$$

Then, an upper bound $g(t)$ is found for $\mathbb{E}[e^{tX}]$ and t is selected to minimize $e^{-t\epsilon}g(t)$. For Hoeffding's inequality, the following lemma provides an upper bound on $\mathbb{E}[e^{tX}]$.

Lemma D.1 (Hoeffding's lemma) *Let X be a random variable with $E[X] = 0$ and $a \leq X \leq b$ with $b > a$. Then, for any $t > 0$, the following inequality holds:*

$$\mathbb{E}[e^{tX}] \leq e^{\frac{t^2(b-a)^2}{8}}\,. \tag{D.2}$$

Proof: By the convexity of $x \mapsto e^x$, for all $x \in [a, b]$, the following holds:

$$e^{tx} \leq \frac{b-x}{b-a}e^{ta} + \frac{x-a}{b-a}e^{tb}\,.$$

Thus, using $\mathbb{E}[X] = 0$,

$$\mathbb{E}[e^{tX}] \leq \mathbb{E}\left[\frac{b-X}{b-a}e^{ta} + \frac{X-a}{b-a}e^{tb}\right] = \frac{b}{b-a}e^{ta} + \frac{-a}{b-a}e^{tb} = e^{\phi(t)}\,,$$

where,

$$\phi(t) = \log\left(\frac{b}{b-a}e^{ta} + \frac{-a}{b-a}e^{tb}\right) = ta + \log\left(\frac{b}{b-a} + \frac{-a}{b-a}e^{t(b-a)}\right)\,.$$

For any $t > 0$, the first and second derivative of ϕ are given below:

$$\begin{aligned}
\phi'(t) &= a - \frac{ae^{t(b-a)}}{\frac{b}{b-a} - \frac{a}{b-a}e^{t(b-a)}} = a - \frac{a}{\frac{b}{b-a}e^{-t(b-a)} - \frac{a}{b-a}}\,,\\
\phi''(t) &= \frac{-abe^{-t(b-a)}}{[\frac{b}{b-a}e^{-t(b-a)} - \frac{a}{b-a}]^2}\\
&= \frac{\alpha(1-\alpha)e^{-t(b-a)}(b-a)^2}{[(1-\alpha)e^{-t(b-a)} + \alpha]^2}\\
&= \frac{\alpha}{[(1-\alpha)e^{-t(b-a)} + \alpha]}\frac{(1-\alpha)e^{-t(b-a)}}{[(1-\alpha)e^{-t(b-a)} + \alpha]}(b-a)^2\,.
\end{aligned}$$

where α denotes $\frac{-a}{b-a}$. Note that $\phi(0) = \phi'(0) = 0$ and that $\phi''(t) = u(1-u)(b-a)^2$ where $u = \frac{\alpha}{[(1-\alpha)e^{-t(b-a)}+\alpha]}$. Since u is in $[0,1]$, $u(1-u)$ is upper bounded by $1/4$ and $\phi''(t) \leq \frac{(b-a)^2}{4}$. Thus, by the second order expansion of function ϕ, there exists $\theta \in [0,t]$ such that:

$$\phi(t) = \phi(0) + t\phi'(0) + \frac{t^2}{2}\phi''(\theta) \leq t^2 \frac{(b-a)^2}{8}\,, \tag{D.3}$$

which completes the proof. □

The lemma can be used to prove the following result known as *Hoeffding's inequality*.

Theorem D.2 (Hoeffding's inequality) *Let $X_1, \ldots, X_m$ be independent random variables with X_i taking values in $[a_i, b_i]$ for all $i \in [m]$. Then, for any $\epsilon > 0$, the following inequalities hold for $S_m = \sum_{i=1}^m X_i$:*

$$\mathbb{P}[S_m - \mathbb{E}[S_m] \geq \epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^m (b_i - a_i)^2} \tag{D.4}$$

$$\mathbb{P}[S_m - \mathbb{E}[S_m] \leq -\epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^m (b_i - a_i)^2}\,. \tag{D.5}$$

Proof: Using the Chernoff bounding technique and lemma D.1, we can write:

$$\begin{aligned}
\mathbb{P}[S_m - \mathbb{E}[S_m] \geq \epsilon] &\leq e^{-t\epsilon}\,\mathbb{E}[e^{t(S_m - \mathbb{E}[S_m])}] \\
&= e^{-t\epsilon}\Pi_{i=1}^m\,\mathbb{E}[e^{t(X_i - \mathbb{E}[X_i])}] && \text{(independence of } X_i\text{s)} \\
&\leq e^{-t\epsilon}\Pi_{i=1}^m e^{t^2(b_i-a_i)^2/8} && \text{(lemma D.1)} \\
&= e^{-t\epsilon} e^{t^2 \sum_{i=1}^m (b_i-a_i)^2/8} \\
&\leq e^{-2\epsilon^2 / \sum_{i=1}^m (b_i-a_i)^2}\,,
\end{aligned}$$

where we chose $t = 4\epsilon / \sum_{i=1}^m (b_i - a_i)^2$ to minimize the upper bound. This proves the first statement of the theorem, and the second statement is shown in a similar way. □

When the variance $\sigma^2_{X_i}$ of each random variable X_i is known and the $\sigma^2_{X_i}$s are relatively small, better concentration bounds can be derived (see *Bennett's* and *Bernstein's inequalities* proven in exercise D.6).

D.2 Sanov's theorem

Here, we present a finer upper bound than Hoeffding's inequality expressed in terms of the binary relative entropy.

Theorem D.3 (Sanov's theorem) *Let $X_1, \ldots, X_m$ be independent random variables drawn according to some distribution $\mathcal{D}$ with mean p and support included in $[0,1]$. Then, for any $q \in [0,1]$, the following inequality holds for $\widehat{p} = \frac{1}{m}\sum_{i=1}^m X_i$:*

$$\mathbb{P}[\widehat{p} \geq q] \leq e^{-m\mathsf{D}(q\|p)}\,,$$

where $\mathsf{D}(q\|p) = q\log\frac{q}{p} + (1-q)\log\frac{1-q}{1-p}$ is the binary relative entropy of p and q.

Proof: For any $t > 0$, by convexity of the function $x \mapsto e^{tx}$, the following inequality holds for all $x \in [0,1]$: $e^{tx} = e^{t[(1-x)\cdot 0 + x\cdot 1]} \leq 1 - x + e^t x$. In view of that, for any $t > 0$, we can write

$$\begin{aligned}
\mathbb{P}[\widehat{p} \geq q] &= \mathbb{P}[e^{tm\widehat{p}} \geq e^{tmq}] \\
&= \mathbb{P}[e^{tm\widehat{p}} \geq e^{tmq}] \\
&\leq e^{-tmq}\, \mathbb{E}[e^{tm\widehat{p}}] && \text{(by Markov's inequality)} \\
&= e^{-tmq}\, \mathbb{E}[e^{t\sum_{i=1}^m X_i}] \\
&= e^{-tmq} \prod_{i=1}^m \mathbb{E}[e^{tX_i}] \\
&\leq e^{-tmq} \prod_{i=1}^m \mathbb{E}[1 - X_i + e^t X_i] && (\forall x \in [0,1], e^{tx} \leq 1 - x + e^t x) \\
&= [e^{-tq}(1 - p + e^t p)]^m.
\end{aligned}$$

Now, the function $f\colon t \mapsto e^{-tq}(1-p+e^t p) = (1-p)e^{-tq} + pe^{t(1-q)}$ reaches its minimum at $t = \log \frac{q(1-p)}{p(1-q)}$. Plugging in this value of t in the inequality above yields $\mathbb{P}[\widehat{p} \geq q] \leq e^{-m\mathsf{D}(q\|p)}$. □

Note that for any $\epsilon > 0$, $\epsilon \leq 1 - p$, with the choice $q = p + \epsilon$, the theorem implies

$$\mathbb{P}[\widehat{p} \geq p + \epsilon] \leq e^{-m\mathsf{D}(p+\epsilon\|p)}. \tag{D.6}$$

This is a finer bound than Hoeffding's inequality (Theorem D.2) since, by Pinsker's inequality (Proposition E.7), $\mathsf{D}(p+\epsilon\|p) \geq \frac{1}{2}(2\epsilon)^2 = 2\epsilon^2$. Similarly, we can derive a symmetric bound by applying the theorem to the random variables $Y_i = 1 - X_i$. Then, for any $\epsilon > 0$, $\epsilon \leq p$, with the choice $q = p - \epsilon$, the theorem implies

$$\mathbb{P}[\widehat{p} \leq p - \epsilon] \leq e^{-m\mathsf{D}(p-\epsilon\|p)}. \tag{D.7}$$

D.3 Multiplicative Chernoff bounds

Sanov's theorem can be used to prove the following *multiplicative Chernoff bounds*.

Theorem D.4 (Multiplicative Chernoff bounds) *Let $X_1, \ldots, X_m$ be independent random variables drawn according to some distribution $\mathcal{D}$ with mean p and support included in $[0,1]$. Then, for any $\gamma \in \left[0, \frac{1}{p} - 1\right]$, the following inequality holds for $\widehat{p} = \frac{1}{m}\sum_{i=1}^m X_i$:*

$$\mathbb{P}[\widehat{p} \geq (1+\gamma)p] \leq e^{-\frac{mp\gamma^2}{3}}$$
$$\mathbb{P}[\widehat{p} \leq (1-\gamma)p] \leq e^{-\frac{mp\gamma^2}{2}}.$$

Proof: The proof consists of deriving in each case a finer lower bound for the binary relative entropy than Pinsker's inequality. Using the inequalities $\log(1+x) \geq \frac{x}{1+\frac{x}{2}}$ and $\log(1+x) < x$, we can write

$$\begin{aligned}
-\mathsf{D}((1+\gamma)p\|p) &= (1+\gamma)p \log \frac{p}{(1+\gamma)p} + (1-(1+\gamma)p) \log \left[\frac{1-p}{1-(1+\gamma)p}\right] \\
&= (1+\gamma)p \log \frac{1}{1+\gamma} + (1-p-\gamma p) \log \left[1 + \frac{\gamma p}{1-p-\gamma p}\right] \\
&\leq (1+\gamma)p \frac{-\gamma}{1+\frac{\gamma}{2}} + (1-p-\gamma p)\frac{\gamma p}{1-p-\gamma p} \\
&= \gamma p \left[1 - \frac{1+\gamma}{1+\frac{\gamma}{2}}\right] = \frac{-\frac{\gamma^2 p}{2}}{1+\frac{\gamma}{2}} = \frac{-\gamma^2 p}{2+\gamma} \\
&\leq \frac{-\gamma^2 p}{2+1} = \frac{-\gamma^2 p}{3}.
\end{aligned}$$

Similarly, using the inequalities $(1-x)\log(1-x) \geq -x+\frac{x^2}{2}$ valid for $x \in (0,1)$ and $\log(1-x) < -x$, we can write

$$\begin{aligned}
-\mathsf{D}((1-\gamma)p\|p) &= (1-\gamma)p\log\frac{p}{(1-\gamma)p} + (1-(1-\gamma)p)\log\left[\frac{1-p}{1-(1-\gamma)p}\right] \\
&= (1-\gamma)p\log\frac{1}{1-\gamma} + (1-p+\gamma p)\log\left[1-\frac{\gamma p}{1-p+\gamma p}\right] \\
&\leq \Big(\gamma-\frac{\gamma^2}{2}\Big)p + (1-p+\gamma p)\frac{-\gamma p}{1-p+\gamma p} = \frac{-\gamma^2 p}{2}.
\end{aligned}$$

This completes the proof. □

D.4 Binomial distribution tails: Upper bounds

Let $X_1, \ldots, X_m$ be independent random variables taking values in $\{0,1\}$ with $\mathbb{P}[X_i = 1] = p \in [0,1]$ for $i = 1, \ldots, m$. Then, $\sum_{i=1}^m X_i$ follows the binomial distribution $B(m,p)$. We will denote by $\overline{X}$ the average $\overline{X} = \frac{1}{m}\sum_{i=1}^m X_i$. Then, the following equality and inequalities hold:

$$\mathbb{P}\left[\overline{X} - p > \epsilon\right] = \sum_{k=\lceil (p+\epsilon)m\rceil}^{m} \binom{m}{k} p^k (1-p)^{m-k} \qquad \text{(Binomial formula)}$$

$$\mathbb{P}\left[\overline{X} - p > \epsilon\right] \leq e^{-2m\epsilon^2} \qquad \text{(Hoeffding's inequality)}$$

$$\mathbb{P}\left[\overline{X} - p > \epsilon\right] \leq e^{-\frac{m\epsilon^2}{2\sigma^2+\frac{2\epsilon}{3}}} \qquad \text{(Bernstein's inequality)}$$

$$\mathbb{P}\left[\overline{X} - p > \epsilon\right] \leq e^{-m\sigma^2\theta\left(\frac{\epsilon}{\sigma^2}\right)} \qquad \text{(Bennett's inequality)}$$

$$\mathbb{P}\left[\overline{X} - p > \epsilon\right] \leq e^{-m\mathsf{D}(p+\epsilon\|p)} \qquad \text{(Sanov's inequality),}$$

where $\sigma^2 = p(1-p) = \mathrm{Var}[X_i]$ and $\theta(x) = (1+x)\log(1+x) - x$. The last three inequalities are shown in exercises D.6 and D.7. Using Bernstein's inequality, for example, we can see that for ϵ relatively small, that is $\epsilon \ll 2\sigma^2$, the upper bound is approximately of the form $e^{-\frac{m\epsilon^2}{2\sigma^2}}$ and thus admits a Gaussian behavior. For $\epsilon \gg 2\sigma^2$, $e^{-\frac{3m\epsilon}{2}}$, the upper bound admits a Poisson behavior.

Figure D.1 shows a comparison of these bounds for different values of the variance $\sigma^2 = p(1-p)$: small variance ($p = .05$), large variance ($p = .5$).

D.5 Binomial distribution tails: Lower bound

Let X be a random variable following the binomial distribution $B(m,p)$ and let k be an integer such that $p \leq \frac{1}{4}$ and $k \geq mp$ or $p \leq \frac{1}{2}$ and $mp \leq k \leq m(1-p)$. Then, the following inequality known as *Slud's inequality* holds:

$$\mathbb{P}[X \geq k] \geq \mathbb{P}\left[N \geq \frac{k-mp}{\sqrt{mp(1-p)}}\right], \tag{D.8}$$

where N is in standard normal form.

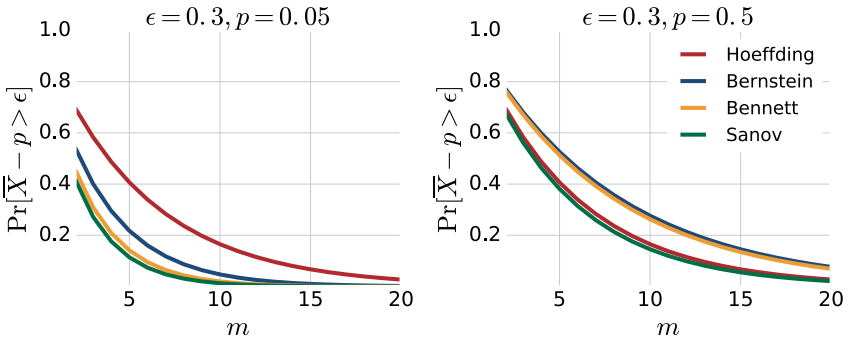


Figure D.1
Comparison of tail bounds for a binomial random variable for $\epsilon = .3$ and $p = .05$ (small variance) or $p = .5$ (maximal variance) as a function of the sample size m.

D.6 Azuma's inequality

This section presents a concentration inequality that is more general than Hoeffding's inequality. Its proof makes use of a Hoeffding's inequality for *martingale differences*.

Definition D.5 (Martingale difference) *A sequence of random variables $V_1, V_2, \ldots$ is a martingale difference sequence with respect to $X_1, X_2, \ldots$ if for all $i > 0$, V_i is a function of $X_1, \ldots, X_i$ and*

$$\mathbb{E}[V_{i+1}|X_1, \ldots, X_i] = 0\,. \tag{D.9}$$

The following result is similar to Hoeffding's lemma.

Lemma D.6 *Let V and Z be random variables satisfying $\mathbb{E}[V|Z] = 0$ and, for some function f and constant $c \geq 0$, the inequalities:*

$$f(Z) \leq V \leq f(Z) + c\,. \tag{D.10}$$

Then, for all $t > 0$, the following upper bound holds:

$$\mathbb{E}[e^{tV}|Z] \leq e^{t^2c^2/8}\,. \tag{D.11}$$

Proof: The proof follows using the same steps as in that of lemma D.1 with conditional expectations used instead of expectations: conditioned on Z, V takes values in $[a, b]$ with $a = f(Z)$ and $b = f(Z) + c$ and its expectation vanishes. □

The lemma is used to prove the following theorem, which is one of the main results of this section.

Theorem D.7 (Azuma's inequality) *Let $V_1, V_2, \ldots$ be a martingale difference sequence with respect to the random variables $X_1, X_2, \ldots$, and assume that for all $i > 0$ there is a constant $c_i \geq 0$ and random variable Z_i, which is a function of $X_1, \ldots, X_{i-1}$, that satisfy*

$$Z_i \leq V_i \leq Z_i + c_i\,. \tag{D.12}$$

Then, for all $\epsilon > 0$ and m, the following inequalities hold:

$$\mathbb{P}\left[\sum_{i=1}^{m} V_i \geq \epsilon\right] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{m} c_i^2}\right) \tag{D.13}$$

$$\mathbb{P}\left[\sum_{i=1}^{m} V_i \leq -\epsilon\right] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{m} c_i^2}\right)\,. \tag{D.14}$$

Proof: For any $k \in [m]$, let $S_k = \sum_{i=1}^{k} V_i$. Then, using Chernoff's bounding technique, for any $t > 0$, we can write

$$\begin{aligned}
\mathbb{P}\left[S_m \geq \epsilon\right] &\leq e^{-t\epsilon}\,\mathbb{E}[e^{tS_m}] \\
&= e^{-t\epsilon}\,\mathbb{E}\left[e^{tS_{m-1}}\,\mathbb{E}[e^{tV_m}|X_1,\ldots,X_{m-1}]\right] \\
&\leq e^{-t\epsilon}\,\mathbb{E}[e^{tS_{m-1}}]e^{t^2c_m^2/8} && \text{(lemma D.6)} \\
&\leq e^{-t\epsilon}e^{t^2\sum_{i=1}^{m}c_i^2/8} && \text{(iterating previous argument)} \\
&= e^{-2\epsilon^2/\sum_{i=1}^{m}c_i^2}\,,
\end{aligned}$$

where we chose $t = 4\epsilon/\sum_{i=1}^{m} c_i^2$ to minimize the upper bound. This proves the first statement of the theorem, and the second statement is shown in a similar way. □

D.7 McDiarmid's inequality

The following is the main result of this section. Its proof makes use of Azuma's inequality.

Theorem D.8 (McDiarmid's inequality) *Let $X_1,\ldots,X_m \in \mathcal{X}^m$ be a set of $m \geq 1$ independent random variables and assume that there exist $c_1,\ldots,c_m > 0$ such that $f\colon \mathcal{X}^m \to \mathbb{R}$ satisfies the following conditions:*

$$\left|f(x_1,\ldots,x_i,\ldots,x_m) - f(x_1,\ldots,x_i',\ldots x_m)\right| \leq c_i\,, \tag{D.15}$$

for all $i \in [m]$ and any points $x_1,\ldots,x_m,x_i' \in \mathcal{X}$. Let $f(S)$ denote $f(X_1,\ldots,X_m)$, then, for all $\epsilon > 0$, the following inequalities hold:

$$\mathbb{P}[f(S) - \mathbb{E}[f(S)] \geq \epsilon] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{m}c_i^2}\right) \tag{D.16}$$

$$\mathbb{P}[f(S) - \mathbb{E}[f(S)] \leq -\epsilon] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{m}c_i^2}\right). \tag{D.17}$$

Proof: Define a sequence of random variables V_k, $k \in [m]$, as follows: $V = f(S) - \mathbb{E}[f(S)]$, $V_1 = \mathbb{E}[V|X_1] - \mathbb{E}[V]$, and for $k > 1$,

$$V_k = \mathbb{E}[V|X_1,\ldots,X_k] - \mathbb{E}[V|X_1,\ldots,X_{k-1}]\,.$$

Note that $V = \sum_{k=1}^{m} V_k$. Furthermore, the random variable $\mathbb{E}[V|X_1,\ldots,X_k]$ is a function of $X_1,\ldots,X_k$. Conditioning on $X_1,\ldots,X_{k-1}$ and taking its expectation is therefore:

$$\mathbb{E}\left[\mathbb{E}[V|X_1,\ldots,X_k]|X_1,\ldots,X_{k-1}\right] = \mathbb{E}[V|X_1,\ldots,X_{k-1}],$$

which implies $\mathbb{E}[V_k|X_1,\ldots,X_{k-1}] = 0$. Thus, the sequence $(V_k)_{k\in[m]}$ is a martingale difference sequence. Next, observe that, since $\mathbb{E}[f(S)]$ is a scalar, V_k can be expressed as follows:

$$V_k = \mathbb{E}[f(S)|X_1,\ldots,X_k] - \mathbb{E}[f(S)|X_1,\ldots,X_{k-1}]\,.$$

Thus, we can define an upper bound W_k and lower bound U_k for V_k by:

$$\begin{aligned}
W_k &= \sup_x \mathbb{E}[f(S)|X_1,\ldots,X_{k-1},x] - \mathbb{E}[f(S)|X_1,\ldots,X_{k-1}] \\
U_k &= \inf_x \mathbb{E}[f(S)|X_1,\ldots,X_{k-1},x] - \mathbb{E}[f(S)|X_1,\ldots,X_{k-1}].
\end{aligned}$$

Now, by (D.15), for any $k \in [m]$, the following holds:

$$W_k - U_k = \sup_{x,x'} \mathbb{E}[f(S)|X_1,\ldots,X_{k-1},x] - \mathbb{E}[f(S)|X_1,\ldots,X_{k-1},x'] \leq c_k\,, \tag{D.18}$$

thus, $U_k \leq V_k \leq U_k + c_k$. In view of these inequalities, we can apply Azuma's inequality to $V = \sum_{k=1}^{m} V_k$, which yields exactly (D.16) and (D.17). □

McDiarmid's inequality is used in several of the proofs in this book. It can be understood in terms of stability: if changing any of its argument affects f only in a limited way, then, its deviations from its mean can be exponentially bounded. Note also that Hoeffding's in-

equality (theorem D.2) is a special instance of McDiarmid's inequality where f is defined by $f\colon (x_1,\dots,x_m) \mapsto \frac{1}{m}\sum_{i=1}^m x_i$.

D.8 Normal distribution tails: Lower bound

If N is a random variable following the standard normal distribution, then for $u > 0$,

$$\mathbb{P}[N \geq u] \geq \frac{1}{2}\Big(1 - \sqrt{1 - e^{-u^2}}\Big). \tag{D.19}$$

D.9 Khintchine-Kahane inequality

The following inequality is useful in a variety of different contexts, including in the proof of a lower bound for the empirical Rademacher complexity of linear hypotheses (chapter 6).

Theorem D.9 (Khintchine-Kahane inequality) *Let $(\mathbb{H}, \|\cdot\|)$ be a normed vector space and let $\mathbf{x}_1,\dots,\mathbf{x}_m$ be $m \geq 1$ elements of $\mathbb{H}$. Let $\boldsymbol{\sigma} = (\sigma_1,\dots,\sigma_m)^\top$ with σ_is independent uniform random variables taking values in $\{-1,+1\}$ (Rademacher variables). Then, the following inequalities hold:*

$$\frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\Big\|\sum_{i=1}^m \sigma_i \mathbf{x}_i\Big\|^2\right] \leq \left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\Big\|\sum_{i=1}^m \sigma_i \mathbf{x}_i\Big\|\right]\right)^2 \leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\Big\|\sum_{i=1}^m \sigma_i \mathbf{x}_i\Big\|^2\right]. \tag{D.20}$$

Proof: The second inequality is a direct consequence of the convexity of $x \mapsto x^2$ and Jensen's inequality (theorem B.20).

To prove the left-hand side inequality, first note that for any $\beta_1,\dots,\beta_m \in \mathbb{R}$, expanding the product $\prod_{i=1}^m(1+\beta_i)$ leads exactly to the sum of all monomials $\beta_1^{\delta_1}\cdots\beta_m^{\delta_m}$, with exponents $\delta_1,\dots,\delta_m$ in $\{0,1\}$. We will use the notation $\beta_1^{\delta_1}\cdots\beta_m^{\delta_m} = \beta^{\boldsymbol{\delta}}$ and $|\boldsymbol{\delta}| = \sum_{i=1}^m \delta_m$ for any $\boldsymbol{\delta} = (\delta_1,\dots,\delta_m) \in \{0,1\}^m$. In view of that, for any $(\alpha_1,\dots,\alpha_m) \in \mathbb{R}^m$ and $t > 0$, the following equality holds:

$$t^2\prod_{i=1}^m(1+\alpha_i/t) = t^2\sum_{\boldsymbol{\delta}\in\{0,1\}^m}\alpha^{\boldsymbol{\delta}}/t^{|\boldsymbol{\delta}|} = \sum_{\boldsymbol{\delta}\in\{0,1\}^m} t^{2-|\boldsymbol{\delta}|}\alpha^{\boldsymbol{\delta}}.$$

Differentiating both sides with respect to t and setting $t = 1$ yields

$$2\prod_{i=1}^m(1+\alpha_i) - \sum_{j=1}^m \alpha_j \prod_{i\neq j}(1+\alpha_i) = \sum_{\boldsymbol{\delta}\in\{0,1\}^m}(2-|\boldsymbol{\delta}|)\alpha^{\boldsymbol{\delta}}\,. \tag{D.21}$$

For any $\boldsymbol{\sigma} \in \{-1,+1\}^m$, let $S_{\boldsymbol{\sigma}}$ be defined by $S_{\boldsymbol{\sigma}} = \|s_{\boldsymbol{\sigma}}\|$ with $s_{\boldsymbol{\sigma}} = \sum_{i=1}^m \sigma_i \mathbf{x}_i$. Then, setting $\alpha_i = \sigma_i\sigma_i'$, multiplying both sides of (D.21) by $S_{\boldsymbol{\sigma}}S_{\boldsymbol{\sigma}'}$, and taking the sum over all $\boldsymbol{\sigma},\boldsymbol{\sigma}' \in \{-1,+1\}^m$ yields

$$\begin{aligned}
\sum_{\boldsymbol{\sigma},\boldsymbol{\sigma}'\in\{-1,+1\}^m}&\left(2\prod_{i=1}^m(1+\sigma_i\sigma_i') - \sum_{j=1}^m \sigma_j\sigma_j'\prod_{i\neq j}(1+\sigma_i\sigma_i')\right)S_{\boldsymbol{\sigma}}S_{\boldsymbol{\sigma}'} \\
&= \sum_{\boldsymbol{\sigma},\boldsymbol{\sigma}'\in\{-1,+1\}^m}\sum_{\boldsymbol{\delta}\in\{0,1\}^m}(2-|\boldsymbol{\delta}|)\sigma^{\boldsymbol{\delta}}\sigma'^{\boldsymbol{\delta}}S_{\boldsymbol{\sigma}}S_{\boldsymbol{\sigma}'} \\
&= \sum_{\boldsymbol{\delta}\in\{0,1\}^m}(2-|\boldsymbol{\delta}|)\sum_{\boldsymbol{\sigma},\boldsymbol{\sigma}'\in\{-1,+1\}^m}\sigma^{\boldsymbol{\delta}}\sigma'^{\boldsymbol{\delta}}S_{\boldsymbol{\sigma}}S_{\boldsymbol{\sigma}'} \\
&= \sum_{\boldsymbol{\delta}\in\{0,1\}^m}(2-|\boldsymbol{\delta}|)\Big[\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m}\sigma^{\boldsymbol{\delta}}S_{\boldsymbol{\sigma}}\Big]^2.
\end{aligned} \tag{D.22}$$

Note that the terms of the right-hand sum with $|\boldsymbol{\delta}| \geq 2$ are non-positive. The terms with $|\boldsymbol{\delta}| = 1$ are null: since $S_{\boldsymbol{\sigma}} = S_{-\boldsymbol{\sigma}}$, we have $\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m}\sigma^{\boldsymbol{\delta}}S_{\boldsymbol{\sigma}} = 0$ in that case. Thus, the right-hand side can be upper bounded by the term with $\boldsymbol{\delta} = 0$, that is, $2\Big(\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m} S_{\boldsymbol{\sigma}}\Big)^2$. The left-hand

side of (D.22) can be rewritten as follows:

$$\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m}(2^{m+1}-m2^{m-1})S_{\boldsymbol{\sigma}}^2+2^{m-1}\sum_{\substack{\boldsymbol{\sigma}\in\{-1,+1\}^m\\ \boldsymbol{\sigma}'\in B(\boldsymbol{\sigma},1)}}S_{\boldsymbol{\sigma}}S_{\boldsymbol{\sigma}'}$$

$$=2^m\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m}S_{\boldsymbol{\sigma}}^2+2^{m-1}\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m}S_{\boldsymbol{\sigma}}\Big(\sum_{\boldsymbol{\sigma}'\in B(\boldsymbol{\sigma},1)}S_{\boldsymbol{\sigma}'}-(m-2)S_{\boldsymbol{\sigma}}\Big), \quad \text{(D.23)}$$

where $B(\boldsymbol{\sigma},1)$ denotes the set of $\boldsymbol{\sigma}'$ that differ from $\boldsymbol{\sigma}$ in exactly one coordinate $j\in[m]$, that is the set of $\boldsymbol{\sigma}'$ with Hamming distance one from $\boldsymbol{\sigma}$. Note that for any such $\boldsymbol{\sigma}'$, $s_{\boldsymbol{\sigma}}-s_{\boldsymbol{\sigma}'}=2\sigma_j\mathbf{x}_j$ for one coordinate $j\in[m]$, thus, $\sum_{\boldsymbol{\sigma}'\in B(\boldsymbol{\sigma},1)}s_{\boldsymbol{\sigma}}-s_{\boldsymbol{\sigma}'}=2s_{\boldsymbol{\sigma}}$. In light of that and using the triangle inequality, we can write

$$\begin{aligned}(m-2)S_{\boldsymbol{\sigma}}=\|ms_{\boldsymbol{\sigma}}\|-\|2s_{\boldsymbol{\sigma}}\|&=\Big\|\sum_{\boldsymbol{\sigma}'\in B(\boldsymbol{\sigma},1)}s_{\boldsymbol{\sigma}}\Big\|-\Big\|\sum_{\boldsymbol{\sigma}'\in B(\boldsymbol{\sigma},1)}s_{\boldsymbol{\sigma}}-s_{\boldsymbol{\sigma}'}\Big\|\\&\leq\Big\|\sum_{\boldsymbol{\sigma}'\in B(\boldsymbol{\sigma},1)}s_{\boldsymbol{\sigma}'}\Big\|\leq\sum_{\boldsymbol{\sigma}'\in B(\boldsymbol{\sigma},1)}S_{\boldsymbol{\sigma}'}.\end{aligned}$$

Thus, the second sum of (D.23) is non-negative and the left-hand side of (D.22) can be lower bounded by the first sum $2^m\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m}S_{\boldsymbol{\sigma}}^2$. Combining this with the upper bound found for (D.22) gives

$$2^m\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m}S_{\boldsymbol{\sigma}}^2\leq 2\Big[\sum_{\boldsymbol{\sigma}\in\{-1,+1\}^m}S_{\boldsymbol{\sigma}}\Big]^2.$$

Dividing both sides by 2^{2m} and using $\mathbb{P}[\boldsymbol{\sigma}]=1/2^m$ gives $\mathbb{E}_{\boldsymbol{\sigma}}[S_{\boldsymbol{\sigma}}^2]\leq 2(\mathbb{E}_{\boldsymbol{\sigma}}[S_{\boldsymbol{\sigma}}])^2$ and completes the proof. □

The constant $1/2$ appearing in the first inequality of (D.20) is optimal. To see this, consider the case where $m=2$ and $\mathbf{x}_1=\mathbf{x}_2=\mathbf{x}$ for some non-zero vector $\mathbf{x}\in\mathbb{H}$. Then, the left-hand side of the first inequality is $\frac{1}{2}\sum_{i=1}^m\|\mathbf{x}_i\|^2=\|\mathbf{x}\|^2$ and the right-hand side $\big(\mathbb{E}_{\boldsymbol{\sigma}}\left[\|(\sigma_1+\sigma_2)\mathbf{x}\|\right]\big)^2=\|\mathbf{x}\|^2(\mathbb{E}_{\boldsymbol{\sigma}}[|\sigma_1+\sigma_2|])^2=\|\mathbf{x}\|^2$.

Note that when the norm $\|\cdot\|$ corresponds to an inner product, as in the case of a Hilbert space $\mathbb{H}$, we can write

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\Big\|\sum_{i=1}^m\sigma_i\mathbf{x}_i\Big\|^2\right]=\sum_{i,j=1}^m\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sigma_i\sigma_j(\mathbf{x}_i\cdot\mathbf{x}_j)\Big]=\sum_{i,j=1}^m\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}[\sigma_i\sigma_j](\mathbf{x}_i\cdot\mathbf{x}_j)=\sum_{i=1}^m\|\mathbf{x}_i\|^2,$$

since by the independence of the random variables σ_i, for $i\neq j$, $\mathbb{E}_{\boldsymbol{\sigma}}[\sigma_i\sigma_j]=\mathbb{E}_{\boldsymbol{\sigma}}[\sigma_i]\,\mathbb{E}_{\boldsymbol{\sigma}}[\sigma_j]=0$. Thus, (D.20) can then be rewritten as follows:

$$\frac{1}{2}\sum_{i=1}^m\|\mathbf{x}_i\|^2\leq\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\Big\|\sum_{i=1}^m\sigma_i\mathbf{x}_i\Big\|\right]\right)^2\leq\sum_{i=1}^m\|\mathbf{x}_i\|^2. \quad \text{(D.24)}$$

D.10 Maximal inequality

The following gives an upper bound on the expectation of the maximum of a finite set of random variables that is useful in several contexts.

Theorem D.10 (Maximal inequality) *Let $X_1\dots X_n$ be $n\geq 1$ real-valued random variables such that for all $j\in[n]$ and $t>0$, $\mathbb{E}[e^{tX_j}]\leq e^{\frac{t^2r^2}{2}}$ for some $r>0$. Then, the following inequality holds:*

$$\mathbb{E}\left[\max_{j\in[n]}X_j\right]\leq r\sqrt{2\log n}.$$

Proof: For any $t>0$, by the convexity of exp and Jensen's inequality, the following holds:

$$e^{t\,\mathbb{E}[\max_{j\in[n]}X_j]}\leq\mathbb{E}[e^{t\max_{j\in[n]}X_j}]=\mathbb{E}\left[\max_{j\in[n]}e^{tX_j}\right]\leq\mathbb{E}\Big[\sum_{j\in[n]}e^{tX_j}\Big]\leq ne^{\frac{t^2r^2}{2}}.$$

Taking the log of both sides yields

$$\mathbb{E}\left[\max_{j\in[n]} X_j\right] \leq \frac{\log n}{t} + \frac{tr^2}{2}. \tag{D.25}$$

Choosing $t = \frac{\sqrt{2\log n}}{r}$, which minimizes the right-hand side, gives the upper bound $r\sqrt{2\log n}$. □

Note that, in view of the expression of their moment-generating function (equation (C.24)), for standard Gaussian random variables X_j, the assumptions of the theorem hold as equalities: $\mathbb{E}[e^{tX_j}] = e^{\frac{t^2}{2}}$.

Corollary D.11 (Maximal inequality) *Let $X_1 \dots X_n$ be $n \geq 1$ real-valued random variables such that for all $j \in [n]$, $X_j = \sum_{i=1}^m Y_{ij}$ where, for each fixed $j \in [n]$, Y_{ij} are independent zero mean random variables taking values in $[-r_i, +r_i]$, for some $r_i > 0$. Then, the following inequality holds:*

$$\mathbb{E}\left[\max_{j\in[n]} X_j\right] \leq r\sqrt{2\log n},$$

with $r = \sqrt{\sum_{i=1}^m r_i^2}$.

Proof: By the independence of the Y_{ij}s for fixed j and Hoeffding's lemma (Lemma D.1), the following inequality holds for all $j \in [n]$:

$$\mathbb{E}[e^{tX_j}] = \mathbb{E}\left[\prod_{i=1}^m e^{tY_{ij}}\right] = \prod_{i=1}^m \mathbb{E}[e^{tY_{ij}}] \leq \prod_{i=1}^m e^{\frac{t^2 r_i^2}{2}} = e^{\frac{t^2 r^2}{2}}, \tag{D.26}$$

with $r^2 = \sum_{i=1}^m r_i^2$. The result then follows immediately by Theorem D.10. □

D.11 Chapter notes

Several of the concentration inequalities presented in this chapter are based on a bounding technique due to Chernoff [1952]. Theorem D.3 is due to Sanov [1957]. For the exponential inequality of exercise D.7, which is an alternative form of Sanov's inequality, see [Hagerup and Rüb, 1990] and the references therein. The multiplicative Chernoff bounds presented in this chapter (Theorem D.4) were given by Angluin and Valiant [1979]. Hoeffding's inequality and lemma (Lemma D.1 and Theorem D.2) are due to Hoeffding [1963]. The improved version of Azuma's inequality [Hoeffding, 1963, Azuma, 1967] presented in this chapter is due to McDiarmid [1989]. The improvement is a reduction of the exponent by a factor of 4. This also appears in McDiarmid's inequality, which is derived from the inequality for bounded martingale sequences. The inequalities presented in exercise D.6 are due to Bernstein [1927] and Bennett [1962]; the exercise is from Devroye and Lugosi [1995].

The binomial inequality of section D.5 is due to Slud [1977]. The tail bound of section D.8 is due to Tate [1953] (see also Anthony and Bartlett [1999]). The Khintchine-Kahane inequality was first studied in the case of real-valued variables $x_1, \dots, x_m$ by Khintchine [1923], with better constants and simpler proofs later provided by Szarek [1976], Haagerup [1982], and Tomaszewski [1982]. The inequality was extended to normed vector spaces by Kahane [1964]. The proof presented here is due to Latała and Oleszkiewicz [1994] and provides the best possible constants.

D.12 Exercises

D.1 Twins paradox. Professor Mamoru teaches at a university whose computer science and math building has $F = 30$ floors.

(1) Assume that the floors are independent and that they have the same probability to be selected by someone taking the elevator. How many people should take the elevator in order to make it likely (probability more than half) that two of them go to the same floor?

(*Hint*: use the Taylor series expansion of $e^{-x} = 1 - x + \ldots$ and give an approximate general expression of the solution.)

(2) Professor Mamoru is popular, and his floor is in fact more likely to be selected than others. Assuming that all other floors are equiprobable, derive the general expression of the probability that two people go to the same floor, using the same approximation as before. How many people should take the elevator in order to make it likely that two of them go to the same floor when the probability of Professor Mamoru's floor is .25, .35, or .5? When $q = .5$, would the answer change if the number of floors were instead $F = 1{,}000$?

(3) The probability models assumed in (1) and (2) are both naive. If you had access to the data collected by the elevator guard, how would you define a more faithful model?

D.2 Estimating label bias. Let $\mathcal{D}$ be a distribution over $\mathcal{X}$ and let $f\colon \mathcal{X} \times \{-1, +1\}$ be a labeling function. Suppose we wish to find a good approximation of the label bias of the distribution $\mathcal{D}$, that is of p_+ defined by:

$$p_+ = \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[f(x) = +1]. \tag{D.27}$$

Let $\mathcal{S}$ be a finite labeled sample of size m drawn i.i.d. according to $\mathcal{D}$. Use $\mathcal{S}$ to derive an estimate $\widehat{p}_+$ of p_+. Show that for any $\delta > 0$, with probability at least $1 - \delta$, $|p_+ - \widehat{p}_+| \leq \sqrt{\frac{\log(2/\delta)}{2m}}$.

D.3 Biased coins. Professor Moent has two coins in his pocket, coin x_A and coin x_B. Both coins are slightly biased, i.e., $\mathbb{P}[x_A = 0] = 1/2 - \epsilon/2$ and $\mathbb{P}[x_B = 0] = 1/2 + \epsilon/2$, where $0 < \epsilon < 1$ is a small positive number, 0 denotes heads and 1 denotes tails. He likes to play the following game with his students. He picks a coin $x \in \{x_A, x_B\}$ from his pocket uniformly at random, tosses it m times, reveals the sequence of 0s and 1s he obtained and asks which coin was tossed. Determine how large m needs to be for a student's coin prediction error to be at most $\delta > 0$.

(a) Let S be a sample of size m. Professor Moent's best student, Oskar, plays according to the decision rule $f_o\colon \{0,1\}^m \to \{x_A, x_B\}$ defined by $f_o(S) = x_A$ iff $N(S) < m/2$, where $N(S)$ is the number of 0's in sample S.

Suppose m is even, then show that

$$error(f_o) \geq \frac{1}{2}\,\mathbb{P}\left[N(S) \geq \frac{m}{2} \middle| x = x_A\right]. \tag{D.28}$$

(b) Assuming m even, show that

$$error(f_o) > \frac{1}{4}\left[1 - \left[1 - e^{-\frac{m\epsilon^2}{1-\epsilon^2}}\right]^{\frac{1}{2}}\right]. \tag{D.29}$$

(c) Argue that if m is odd, the probability can be lower bounded by using $m+1$ in the bound in (a) and conclude that for both odd and even m,

$$error(f_o) > \frac{1}{4}\left[1 - \left[1 - e^{-\frac{2\lceil m/2\rceil\epsilon^2}{1-\epsilon^2}}\right]^{\frac{1}{2}}\right]. \tag{D.30}$$

(d) Using this bound, how large must m be if Oskar's error is at most δ, where $0 < \delta < 1/4$. What is the asymptotic behavior of this lower bound as a function of ϵ?

(e) Show that no decision rule $f\colon \{0,1\}^m \to \{x_A, x_B\}$ can do better than Oskar's rule f_o. Conclude that the lower bound of the previous question applies to all rules.

D.4 Concentration bounds. Let X be a non-negative random variable satisfying $\mathbb{P}[X > t] \leq ce^{-2mt^2}$ for all $t > 0$ and some $c > 0$. Show that $\mathbb{E}[X^2] \leq \frac{\log(ce)}{2m}$ (*Hint*: to do that, use the

identity $\mathbb{E}[X^2] = \int_0^{+\infty} \mathbb{P}[X^2 > t]dt$, write $\int_0^{+\infty} = \int_0^u + \int_u^{+\infty}$, bound the first term by u and find the best u to minimize the upper bound).

D.5 Comparison of Hoeffding's and Chebyshev's inequalities. Let $X_1, \ldots, X_m$ be a sequence of random variables taking values in $[0, 1]$ with the same mean μ and variance $\sigma^2 < \infty$ and let $\overline{X} = \frac{1}{m}\sum_{i=1}^m X_i$.

(a) For any $\epsilon > 0$, give a bound on $\mathbb{P}[|\overline{X} - \mu| > \epsilon]$ using Chebyshev's inequality, then Hoeffding's inequality. For what values of σ is Chebyshev's inequality tighter?

(b) Assume that the random variables X_i take values in $\{0, 1\}$. Show that $\sigma^2 \leq \frac{1}{4}$. Use this to simplify Chebyshev's inequality. Choose $\epsilon = .05$ and plot Chebyshev's inequality thereby modified and Hoeffding's inequality as a function of m (you can use your preferred program for generating the plots).

D.6 Bennett's and Bernstein's inequalities. The objective of this problem is to prove these two inequalities.

(a) Show that for any $t > 0$, and any random variable X with $\mathbb{E}[X] = 0$, $\mathbb{E}[X^2] = \sigma^2$, and $X \leq c$,

$$\mathbb{E}[e^{tX}] \leq e^{f(\sigma^2/c^2)}, \tag{D.31}$$

where

$$f(x) = \log\Big(\frac{1}{1+x}e^{-ctx} + \frac{x}{1+x}e^{ct}\Big).$$

(b) Show that $f''(x) \leq 0$ for $x \geq 0$.

(c) Using Chernoff's bounding technique, show that

$$\mathbb{P}\left[\frac{1}{m}\sum_{i=1}^m X_i \geq \epsilon\right] \leq e^{-tm\epsilon + \sum_{i=1}^m f(\sigma^2_{X_i}/c^2)},$$

where ($\sigma^2_{X_i}$ is the variance of X_i.

(d) Show that $f(x) \leq f(0) + xf'(0) = (e^{ct} - 1 - ct)x$.

(e) Using the bound derived in (4), find the optimal value of t.

(f) *Bennett's inequality*. Let $X_1, \ldots, X_m$ be independent real-valued random variables with zero mean such that for $i = 1, \ldots, m$, $X_i \leq c$. Let $\sigma^2 = \frac{1}{m}\sum_{i=1}^m \sigma^2_{X_i}$. Show that

$$\mathbb{P}\left[\frac{1}{m}\sum_{i=1}^m X_i > \epsilon\right] \leq \exp\left(-\frac{m\sigma^2}{c^2}\theta\Big(\frac{\epsilon c}{\sigma^2}\Big)\right), \tag{D.32}$$

where $\theta(x) = (1 + x)\log(1 + x) - x$.

(g) *Bernstein's inequality*. Show that under the same conditions as Bennett's inequality

$$\mathbb{P}\left[\frac{1}{m}\sum_{i=1}^m X_i > \epsilon\right] \leq \exp\left(-\frac{m\epsilon^2}{2\sigma^2 + 2c\epsilon/3}\right). \tag{D.33}$$

(*Hint*: show that for all $x \geq 0$, $\theta(x) \geq h(x) = \frac{3}{2}\frac{x^2}{x+3}$.)

(h) Write Hoeffding's inequality assuming the same conditions. For what values of σ is Bernstein's inequality better than Hoeffding's inequality?

D.7 Exponential inequality. Let X be a random variable following a binomial distribution $B(m, p)$.

(a) Use Sanov's inequality to show that the following *exponential inequality* holds for any $\epsilon > 0$:

$$\mathbb{P}\left[\frac{X}{m} - p > \epsilon\right] \leq \left[\left(\frac{p}{p+\epsilon}\right)^{p+\epsilon} \left(\frac{1-p}{1-(p+\epsilon)}\right)^{1-(p+\epsilon)}\right]^m. \tag{D.34}$$

(b) Use that to show that the following holds:

$$\mathbb{P}\left[\frac{X}{m} - p > \epsilon\right] \leq \left(\frac{p}{p+\epsilon}\right)^{m(p+\epsilon)} e^{m\epsilon}. \tag{D.35}$$

(c) Prove that

$$\mathbb{P}\left[\frac{X}{m} - p > \epsilon\right] \leq e^{-mp\theta(\frac{\epsilon}{p})}, \tag{D.36}$$

where θ is defined as in exercise D.6.

E Notions of Information Theory

This chapter introduces some basic notions of information theory useful for the presentation of several learning algorithms and their properties. The definitions and theorems are given in the case of discrete random variables or distributions, but they can be straightforwardly extended to the continuous case.

We start with the notion of *entropy*, which can be viewed as a measure of the uncertainty of a random variable.

E.1 Entropy

Definition E.1 (Entropy) *The* entropy *of a discrete random variable X with probability mass function $p(x) = \mathbb{P}[X = x]$ is denoted by $\mathsf{H}(X)$ and defined by*

$$\mathsf{H}(X) = -\mathbb{E}\left[\log(p(X))\right] = -\sum_{x \in \mathcal{X}} p(x) \log(p(x)). \tag{E.1}$$

We define by the same expression the entropy of a distribution p *and abusively denote that by* $\mathsf{H}(p)$.

The base of the logarithm is not critical in this definition since it only affects the value by a multiplicative constant. Thus, unless otherwise specified, we will consider the natural logarithm (base e). If we use base 2, then $-\log_2(p(x))$ is the number of bits needed to represent $p(x)$. Thus, by definition, the entropy of X can be viewed as the average number of bits (or amount of information) needed for the description of the random variable X. By the same property, the entropy is always non-negative:

$$\mathsf{H}(X) \geq 0. \tag{E.2}$$

As an example, the entropy of a biased coin X_p taking value 1 with probability p and 0 with probability $1 - p$ is given by

$$\mathsf{H}(X_p) = -p \log p - (1 - p) \log(1 - p). \tag{E.3}$$

The corresponding function of p is often referred to as the *binary entropy function*. Figure E.1 shows a plot of that function when using base 2 for the logarithm. As can be seen from the figure, the entropy is a concave function.[26] It reaches its maximum at $p = \frac{1}{2}$, which corresponds to the most uncertain case, and its minima at $p = 0$ or $p = 1$ which correspond to the fully certain cases. More generally, assume that the input space $\mathcal{X}$ has a finite cardinality $N \geq 1$. Then, by Jensen's

[26] We will see later that the entropy function is always concave.

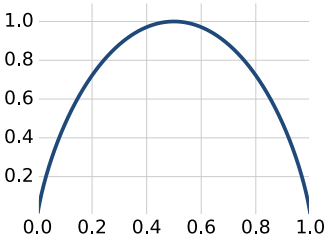


Figure E.1
A plot of the binary entropy as a function of the bias p.

inequality, in view of the concavity of logarithm, the following inequality holds:

$$\mathsf{H}(X) = \mathbb{E}\left[\log \frac{1}{p(X)}\right] \leq \log \mathbb{E}\left[\frac{1}{p(X)}\right] = \log\left(\sum_{x \in \mathcal{X}} \frac{p(x)}{p(x)}\right) = \log N. \tag{E.4}$$

Thus, more generally, the maximum value of the entropy is $\log N$, that is the entropy of the uniform distribution.

The entropy is a lower bound on lossless data compression and is therefore a critical quantity to consider in information theory. It is also closely related to the notions of entropy in thermodynamics and quantum physics.

E.2 Relative entropy

Here, we introduce a measure of divergence between two distributions p and q, *relative entropy*, which is related to the notion of entropy. The following is its definition in the discrete case.

Definition E.2 (Relative entropy) *The* relative entropy *(or Kullback-Leibler divergence) of two distributions p and q is denoted by $\mathsf{D}(p\|q)$ and defined by*

$$\mathsf{D}(p\|q) = \mathop{\mathbb{E}}_{p}\left[\log \frac{p(X)}{q(X)}\right] = \sum_{x \in \mathcal{X}} p(x) \log \left[\frac{p(x)}{q(x)}\right], \tag{E.5}$$

with the conventions $0 \log 0 = 0$, $0 \log \frac{0}{0} = 0$, and $a \log \frac{a}{0} = +\infty$ for $a > 0$.

Note that, in view of these conventions, whenever $q(x) = 0$ for some x in the support of p ($p(x) > 0$), the relative entropy is infinite: $\mathsf{D}(p\|q) = \infty$. Thus, the relative entropy does not provide an informative measure of the divergence of p and q in such cases.

As for the entropy, the base of the logarithm is not critical in the definition of the relative entropy and we will consider the natural logarithm unless otherwise specified. If we use base 2, the relative entropy can be interpreted in terms of coding length. Ideally, one could design for p an optimal code with average length the entropy $\mathsf{H}(p)$. The relative entropy is the average number of additional bits needed to encode p when using an optimal code for q instead of one for p since it can be expressed as the difference $\mathsf{D}(p\|q) = \mathbb{E}_p[\log \frac{1}{q(X)}] - \mathsf{H}(p)$, which, as shown by the following proposition, is always non-negative.

Proposition E.3 (Non-negativity of relative entropy) *For any two distributions p and q, the following inequality holds:*

$$\mathsf{D}(p\|q) \geq 0. \tag{E.6}$$

Furthermore, $\mathsf{D}(p\|q) = 0$ iff $p = q$.

Proof: By the concavity of logarithm and Jensen's inequality, the following holds:

$$-\mathsf{D}(p\|q) = \sum_{x\colon p(x)>0} p(x)\log\left(\frac{q(x)}{p(x)}\right) \le \log\left(\sum_{x\colon p(x)>0} p(x)\frac{q(x)}{p(x)}\right)$$
$$= \log\left(\sum_{x\colon p(x)>0} q(x)\right) \le \log(1) = 0.$$

Thus, the relative entropy is always non-negative for all distributions p and q. The equality $\mathsf{D}(p\|q) = 0$ can hold only if both of the inequalities above are equalities. The last one implies that $\sum_{x\colon p(x)>0} q(x) = 1$. Since the log function is strictly concave, the first inequality can be an equality only if $\frac{q(x)}{p(x)}$ is some constant α over $\{x\colon p(x) > 0\}$. Since $p(x)$ sums to one over that set, we must have $\sum_{x\colon p(x)>0} q(x) = \alpha$. Thus, $\alpha = 1$, which implies $q(x) = p(x)$ for all $x \in \{x\colon p(x) > 0\}$ and thus for all x. Finally, by definition, for any distribution p, $\mathsf{D}(p\|p) = 0$, which completes the proof. □

The relative entropy is not a distance. It is asymmetric: in general, $\mathsf{D}(p\|q) \neq \mathsf{D}(q\|p)$ for two distributions p and q. Furthermore, in general, the relative entropy does not verify the triangle inequality.

Corollary E.4 (Log-sum inequality) *For any set of non-negative real numbers $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$, the following inequality holds:*

$$\sum_{i=1}^{n} a_i \log\left(\frac{a_i}{b_i}\right) \ge \left(\sum_{i=1}^{n} a_i\right) \log\left(\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}\right), \tag{E.7}$$

with the conventions $0\log 0 = 0$, $0\log\frac{0}{0} = 0$, and $a\log\frac{a}{0} = +\infty$ for $a > 0$.

Furthermore, equality holds in (E.7) iff $\frac{a_i}{b_i}$ is a constant (does not depend on i).

Proof: With the conventions adopted, it is clear that the equality holds if $\sum_{i=1}^{n} a_i = 0$, that is $a_i = 0$ for all $i \in [n]$, or $\sum_{i=1}^{n} b_i = 0$, that is $b_i = 0$ for all $i \in [n]$. Thus, we can assume that $\sum_{i=1}^{n} a_i \neq 0$ and $\sum_{i=1}^{n} b_i \neq 0$. Since the inequality is invariant by scaling of the a_is or b_is, we can multiply them by positive constants such that $\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i = 1$. The inequality then coincides with the non-negativity of the relative entropy of the distributions thereby defined by a_is and b_is and the result holds by Proposition E.3. □

Corollary E.5 (Joint convexity of relative entropy) *The relative entropy function $(p, q) \mapsto \mathsf{D}(p\|q)$ is convex.*

Proof: For any $\alpha \in [0, 1]$ and any four probability distributions p_1, p_2, q_1, q_2, by the Log-sum inequality (Corollary E.4), the following holds for any fixed x:

$$(\alpha p_1(x) + (1-\alpha)p_2(x))\log\left[\frac{\alpha p_1(x) + (1-\alpha)p_2(x)}{\alpha q_1(x) + (1-\alpha)q_2(x)}\right]$$
$$\le \alpha p_1(x)\log\left[\frac{\alpha p_1(x)}{\alpha q_1(x)}\right] + (1-\alpha)p_2(x)\log\left[\frac{(1-\alpha)p_1(x)}{(1-\alpha)q_2(x)}\right]. \tag{E.8}$$

Summing up these inequalities over all x yields:

$$\mathsf{D}\Big(\alpha p_1 + (1-\alpha)p_2 \| \alpha q_1 + (1-\alpha)q_2\Big) \le \alpha\mathsf{D}(p_1\|q_1) + (1-\alpha)\mathsf{D}(p_2\|q_2), \tag{E.9}$$

which concludes the proof. □

Corollary E.6 (Concavity of the entropy) *The entropy function $p \mapsto \mathsf{H}(p)$ is concave.*

Proof: Observe that for any fixed distribution p_0 over $\mathcal{X}$, by definition of the relative entropy, we can write

$$\mathsf{D}(p\|p_0) = \sum_{x\in\mathcal{X}} p(x)\log(p(x)) - \sum_{x\in\mathcal{X}} p(x)\log(p_0(x)). \tag{E.10}$$

Thus, $\mathsf{H}(p) = -\mathsf{D}(p\|p_0) - \sum_{x\in\mathcal{X}} p(x)\log(p_0(x))$. By Corollary E.5, the first term is a concave function of p. The second term is linear in p and therefore is concave. Thus, H is concave as a sum of two concave functions. □

Proposition E.7 (Pinsker's inequality) *For any two distributions p and q, the following inequality holds:*

$$\mathsf{D}(p\|q) \geq \frac{1}{2}\|p-q\|_1^2. \tag{E.11}$$

Proof: We first show that the inequality holds for distributions over a set $\mathcal{A} = \{a_0, a_1\}$ of cardinality 2. Let $p_0 = p(a_0)$ and $q_0 = q(a_0)$. Fix $p_0 \in [0,1]$ and consider the function $f\colon q_0 \mapsto f(q_0)$ defined by

$$f(q_0) = p_0\log\frac{p_0}{q_0} + (1-p_0)\log\frac{1-p_0}{1-q_0} - 2(p_0-q_0)^2. \tag{E.12}$$

Observe that $f(p_0) = 0$ and that for $q_0 \in (0,1)$,

$$f'(q_0) = -\frac{p_0}{q_0} + \frac{1-p_0}{1-q_0} + 4(p_0-q_0) = (q_0-p_0)\left[\frac{1}{(1-q_0)q_0} - 4\right]. \tag{E.13}$$

Since $(1-q_0)q_0 \leq \frac{1}{4}$, $[\frac{1}{(1-q_0)q_0} - 4]$ is non-negative. Thus, $f'(q_0) \leq 0$ for $q_0 \leq p_0$ and $f'(q_0) \geq 0$ for $q_0 \geq p_0$. Thus, f reaches its minimum at $q_0 = p_0$, which implies $f(q_0) \geq f(p_0) = 0$ for all q_0. Since $f(q_0)$ can be expressed as follows:

$$f(q_0) = \mathsf{D}(p\|q) - 2(p_0-q_0)^2 \tag{E.14}$$

$$= \mathsf{D}(p\|q) - \frac{1}{2}\Big[|p_0-q_0| + |(1-p_0)-(1-q_0)|\Big]^2 \tag{E.15}$$

$$= \mathsf{D}(p\|q) - \frac{1}{2}\|p-q\|_1^2 \geq 0, \tag{E.16}$$

this proves the inequality for a set $\mathcal{A} = \{a_0, a_1\}$ of cardinality 2.

Now, consider the distributions p' and q' defined over $\mathcal{A} = \{a_0, a_1\}$ with $p'(a_0) = \sum_{x\in a_0} p(x)$, and $q'(a_0) = \sum_{x\in a_0} q(x)$ where $a_0 = \{x \in \mathcal{X}\colon p(x) \geq q(x)\}$ and $a_1 = \{x \in \mathcal{X}\colon p(x) < q(x)\}$. By the Log-sum inequality (Corollary E.4),

$$\mathsf{D}(p\|q) = \sum_{x\in a_0} p(x)\log\left[\frac{p(x)}{q(x)}\right] + \sum_{x\in a_1} p(x)\log\left[\frac{p(x)}{q(x)}\right] \tag{E.17}$$

$$\geq p(a_0)\log\left[\frac{p(a_0)}{q(a_0)}\right] + p(a_1)\log\left[\frac{p(a_1)}{q(a_1)}\right] \tag{E.18}$$

$$= \mathsf{D}(p'\|q'). \tag{E.19}$$

Combining this inequality with the observation that

$$\|p'-q'\|_1 = (p(a_0)-q(a_0)) - (p(a_1)-q(a_1)) \tag{E.20}$$

$$= \sum_{x\in a_0}(p(x)-q(x)) - \sum_{x\in a_1}(p(x)-q(x)) \tag{E.21}$$

$$= \sum_{x\in\mathcal{X}}|p(x)-q(x)| \tag{E.22}$$

$$= \|p-q\|_1, \tag{E.23}$$

shows that $\mathsf{D}(p\|q) \geq \mathsf{D}(p'\|q') \geq \frac{1}{2}\|p-q\|_1^2$ and concludes the proof. □

Definition E.8 (Conditional relative entropy) *Let p and q be two probability distributions defined over $\mathcal{X}\times\mathcal{Y}$ and r a distribution over $\mathcal{X}$. Then, the* conditional relative entropy *of p and q with respect to the marginal r is defined as the expectation of the relative entropy of $p(\cdot|X)$ and $q(\cdot|X)$ with respect to r:*

$$\mathop{\mathbb{E}}_{X\sim r}\Big[\mathsf{D}\big(p(\cdot|X)\|q(\cdot|X)\big)\Big] = \sum_{x\in\mathcal{X}} r(x)\sum_{y\in\mathcal{Y}} p(y|x)\log\frac{p(y|x)}{q(y|x)} = \mathsf{D}(\widetilde{p}\|\widetilde{q}), \tag{E.24}$$

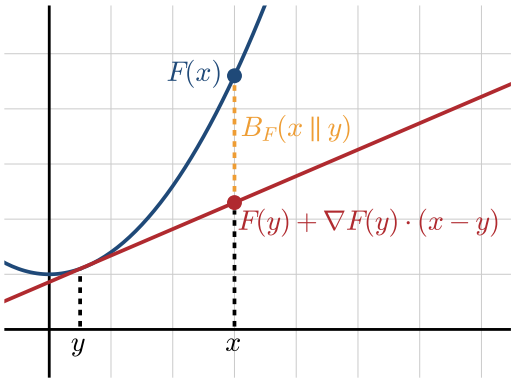


Figure E.2
Illustration of the quantity measured by the Bregman divergence defined based on a convex and differentiable function F. The divergence measures the distance between $F(x)$ and the hyperplane tangent to the curve at point y.

where $\widetilde{p}(x,y) = r(x)p(y|x)$ *and* $\widetilde{q}(x,y) = r(x)q(y|x)$*, with the conventions* $0 \log 0 = 0$*,* $0 \log \frac{0}{0} = 0$*, and* $a \log \frac{a}{0} = +\infty$ *for* $a > 0$*.*

E.3 Mutual information

Definition E.9 (Mutual information) *Let* X *and* Y *be two random variables with joint probability distribution function* $p(\cdot,\cdot)$ *and marginal probability distribution functions* $p(x)$ *and* $p(y)$*. Then, the* mutual information *of* X *and* Y *is denoted by* $I(X,Y)$ *and defined as follows:*

$$I(X,Y) = \mathsf{D}(p(x,y) \| p(x)p(y)) \tag{E.25}$$

$$= \mathop{\mathbb{E}}_{p(x,y)} \left[\log \frac{p(X,Y)}{p(X)p(Y)} \right] = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log \left[\frac{p(x,y)}{p(x)p(y)} \right], \tag{E.26}$$

with the conventions $0 \log 0 = 0$*,* $0 \log \frac{0}{0} = 0$*, and* $a \log \frac{a}{0} = +\infty$ *for* $a > 0$*.*

When the random variables X and Y are independent, their joint distributions is the product of the marginals $p(x)$ and $p(y)$. Thus, the mutual information is a measure of the closeness of the joint distribution $p(x,y)$ to its value when X and Y are independent, where closeness is measured via the relative entropy divergence. As such, it can be viewed as a measure of the amount of information that each random variable can provide about the other. Note that by Proposition E.3, the equality $I(X,Y) = 0$ holds iff $p(x,y) = p(x)p(y)$ for all x, y, that is iff X and Y are independent.

E.4 Bregman divergences

Here we introduce the so-called *unnormalized relative entropy* $\widetilde{\mathsf{D}}$ defined for all non-negative functions p, q in $\mathbb{R}^{\mathcal{X}}$ by

$$\widetilde{\mathsf{D}}(p \| q) = \sum_{x \in \mathcal{X}} p(x) \log \left[\frac{p(x)}{q(x)} \right] + \big(q(x) - p(x) \big), \tag{E.27}$$

Table E.1
Examples of Bregman divergences and corresponding convex functions.

	$\mathsf{B}_F(x \,\|\, y)$	$F(x)$
Squared L_2-distance	$\|x-y\|^2$	$\|x\|^2$
Mahalanobis distance	$(x-y)^\top Q(x-y)$	$x^\top Q x$
Unnormalized relative entropy	$\widetilde{\mathsf{D}}(x \,\|\, y)$	$\sum_{i\in I} x(i)\log(x(i)) - x(i)$

with the conventions $0\log 0 = 0$, $0\log\frac{0}{0} = 0$, and $a\log\frac{a}{0} = +\infty$ for $a > 0$. The relative entropy coincides with the unnormalized relative entropy when restricted to $\Delta\times\Delta$, where Δ is the family of distributions defined over $\mathcal{X}$. The relative entropy inherits several properties of the unnormalized relative entropy, in particular, it can be shown that $\widetilde{\mathsf{D}}(p\|q) \geq 0$. Many of these properties are in fact shared by a broader family of divergences known as *Bregman divergences*.

Definition E.10 (Bregman divergences) *Let F be a convex and differentiable function defined over a convex (open) set $\mathcal{C}$ in a Hilbert space $\mathbb{H}$. Then, the* Bregman divergence B_F *associated to F is defined for all $x, y \in \mathcal{C}$ by*

$$\mathsf{B}_F(x\|y) = F(x) - F(y) - \langle \nabla F(y), x - y\rangle\,. \tag{E.28}$$

Thus, $\mathsf{B}_F(x \,\|\, y)$ measures the difference of $F(x)$ and its linear approximation. Figure E.2 illustrates this definition. Table E.1 provides several examples of Bregman divergences along with their corresponding convex functions $F(x)$. Note that, although the unnormalized relative entropy is a Bregman divergence, the relative entropy is not a Bregman divergence since it is defined over the simplex which is not an open set and has an empty interior.

The following proposition presents several general properties of Bregman divergences.

Proposition E.11 *Let F be a convex and differentiable function defined over a convex set $\mathcal{C}$ in a Hilbert space $\mathbb{H}$. Then, the following properties hold:*

1. $\forall x, y \in \mathcal{C},\ \mathsf{B}_F(x \,\|\, y) \geq 0$.
2. $\forall x, y, z \in \mathcal{C}, \langle \nabla F(x) - \nabla F(y), x - z\rangle = \mathsf{B}_F(x \,\|\, y) + \mathsf{B}_F(z \,\|\, x) - \mathsf{B}_F(z \,\|\, y)$.
3. B_F *is convex in its first argument. If additionally F is strictly convex, then B_F is strictly convex in its first argument.*
4. *Linearity: let G be a convex and differentiable function over $\mathcal{C}$, then, for any $\alpha, \beta \in \mathbb{R}$, $\mathsf{B}_{\alpha F+\beta G} = \alpha\mathsf{B}_F + \beta\mathsf{B}_G$.*

   For the following properties, we will assume additionally that F is strictly convex.

5. *Projection: for any $y \in \mathcal{C}$ and any closed convex set $\mathcal{K} \subseteq \mathcal{C}$, the* B_F-projection of y over K, $P_{\mathcal{K}}(y) = \operatorname{argmin}_{x\in\mathcal{K}} \mathsf{B}_F(x \,\|\, y)$, *is unique.*
6. *Pythagorean theorem: for $y \in \mathcal{C}$ and any closed convex set $\mathcal{K} \subseteq \mathcal{C}$, the following holds for all $x \in \mathcal{K}$:*
$$\mathsf{B}_F(x \,\|\, y) \geq \mathsf{B}_F(x \,\|\, P_{\mathcal{K}}(y)) + \mathsf{B}_F(P_{\mathcal{K}}(y) \,\|\, y). \tag{E.29}$$
7. *Conjugate divergence: assume that F is closed proper strictly convex, and that the norm of its gradient tends to infinity near the boundary of $\mathcal{C}$: $\lim_{x\to\partial\mathcal{C}} \|\nabla F(x)\| = +\infty$. The pair $(\mathcal{C}, F)$ is then said to be a* convex function of Legendre type. *Then, the conjugate of F, F^*, is differentiable and the following holds for all $x, y \in \mathcal{C}$:*
$$\mathsf{B}_F(x \,\|\, y) = \mathsf{B}_{F^*}(\nabla F(y) \,\|\, \nabla F(x)). \tag{E.30}$$

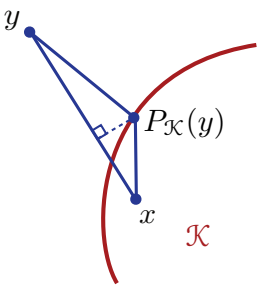


Figure E.3
A depiction of the Pythagorean theorem stated in proposition E.11, where the squared length of each line illustrates the magnitude of the Bregman divergence between the points it connects.

Proof: Property (1) holds by convexity of the function F (the graph of F is above its tangent, see equation (B.3)).

Property (2) follows directly from the definition of the Bregman divergence:

$$\begin{aligned}
&\mathsf{B}_F(x \,\|\, y) + \mathsf{B}_F(z \,\|\, x) - \mathsf{B}_F(z \,\|\, y) \\
&\quad = -\langle \nabla F(y), x - y\rangle - \langle \nabla F(x), z - x\rangle + \langle \nabla F(y), z - y\rangle \\
&\quad = \langle \nabla F(x) - \nabla F(y), x - z\rangle \ .
\end{aligned}$$

Property (3) holds since $x \mapsto F(x) - F(y) - \langle \nabla F(y), x - y\rangle$ is convex as a sum of the convex function $x \mapsto F(x)$ and the affine and thus convex function $x \mapsto -F(y) - \langle \nabla F(y), x - y\rangle$. Similarly, B_F is strictly convex with respect to its first argument if F is strictly convex, as a sum of a strictly convex function and an affine function.

Property (4) follows from a series of equalities:

$$\begin{aligned}
\mathsf{B}_{\alpha F + \beta G} &= \alpha F(x) + \beta G(x) - \alpha F(y) - \beta G(y) - \big\langle \nabla\big(\alpha F(y) + \beta G(y)\big), x - y\big\rangle \\
&= \alpha\big(F(x) - F(y) - \langle \nabla F(y), x - y\rangle\big) + \beta\big(G(x) - G(y) - \langle \nabla G(y), x - y\rangle\big) \\
&= \alpha \mathsf{B}_F + \beta \mathsf{B}_G,
\end{aligned}$$

where we have used the fact that both the gradient and inner-product are linear functions.

Property (5) holds since, by Property (3), $\min_{x \in \mathcal{K}} \mathsf{B}_F(x \,\|\, y)$ is a convex optimization problem with a strictly convex objective function.

For property (6), fix $y \in \mathcal{C}$ and let J be the function defined for all $\alpha \in [0, 1]$ by

$$J(\alpha) = \mathsf{B}_F(\alpha x + (1 - \alpha) P_{\mathcal{K}}(y) \,\|\, y).$$

Since $\mathcal{C}$ is convex, for any $\alpha \in [0, 1]$, $\alpha x + (1 - \alpha) P_{\mathcal{K}}(y)$ is in $\mathcal{C}$. F is differentiable over $\mathcal{C}$ therefore J is also differentiable as a composition of F with $\alpha \mapsto \alpha x + (1 - \alpha) P_{\mathcal{K}}(y)$. By definition of $P_{\mathcal{K}}(y)$, for any $\alpha \in (0, 1]$,

$$\frac{J(\alpha) - J(0)}{\alpha} = \frac{\mathsf{B}_F(\alpha x + (1 - \alpha) P_{\mathcal{K}}(y) \,\|\, y) - \mathsf{B}_F(P_{\mathcal{K}}(y) \,\|\, y)}{\alpha} \geq 0. \tag{E.31}$$

This implies that $J'(0) \geq 0$. From the following expression of $J(\alpha)$:

$$J(\alpha) = F(\alpha x + (1 - \alpha) P_{\mathcal{K}}(y)) - F(y) - \langle \nabla F(y), \alpha x + (1 - \alpha) P_{\mathcal{K}}(y) - y\rangle\,, \tag{E.32}$$

we can compute its derivative at 0:

$$\begin{aligned}
J'(0) &= \langle x - P_{\mathcal{K}}(y), \nabla F(P_{\mathcal{K}}(y))\rangle - \langle \nabla F(y), x - P_{\mathcal{K}}(y)\rangle \\
&= -\mathsf{B}_F(x \,\|\, P_{\mathcal{K}}(y)) + F(x) - F(P_{\mathcal{K}}(y)) - \langle \nabla F(y), x - P_{\mathcal{K}}(y)\rangle \\
&= -\mathsf{B}_F(x \,\|\, P_{\mathcal{K}}(y)) + F(x) - F(P_{\mathcal{K}}(y)) - \langle \nabla F(y), x - y\rangle - \langle \nabla F(y), y - P_{\mathcal{K}}(y)\rangle \\
&= -\mathsf{B}_F(x \,\|\, P_{\mathcal{K}}(y)) + \mathsf{B}_F(x \,\|\, y) + F(y) - F(P_{\mathcal{K}}(y)) - \langle \nabla F(y), y - P_{\mathcal{K}}(y)\rangle \\
&= -\mathsf{B}_F(x \,\|\, P_{\mathcal{K}}(y)) + \mathsf{B}_F(x \,\|\, y) - \mathsf{B}_F(P_{\mathcal{K}}(y) \,\|\, y) \geq 0,
\end{aligned}$$

which concludes the proof of Property (6).

For property (7), note that, by definition, for any y, F^* is defined by

$$F^*(y) = \sup_{x \in \mathcal{C}} \big\{ \langle x, y \rangle - F(x) \big\}. \tag{E.33}$$

F^* is convex and admits a sub-differential at any y. By the strict convexity of F, the function $x \mapsto \langle x, y \rangle - F(x)$ is strictly concave and differentiable over $\mathcal{C}$ and the norm of its gradient, $y - \nabla F(x)$, tends to infinity near the boundary of $\mathcal{C}$ (by the corresponding property assumed for F). Thus, its supremum is reached at a unique point $x_y \in \mathcal{C}$ where its gradient is zero, that is at x_y with $\nabla F(x_y) = y$. This implies that for any y, $\partial F^*(y)$, the subdifferential of F^*, is reduced to a singleton. Thus, F^* is differentiable and its gradient at y is $\nabla F^*(y) = x_y = \nabla^{-1} F(y)$. Since F^* is convex and differentiable, its Bregman divergence is well defined. Furthermore, $F^*(y) = \big\langle \nabla F^{-1}(y), y \big\rangle - F(\nabla F^{-1}(y))$ since $x_y = \nabla^{-1} F(y)$. For any $x, y \in \mathcal{C}$, using the definition of B_{F^*} and the expression of $\nabla F^*(y)$ and $F^*(y)$ we can write

$$\begin{aligned}
&\mathsf{B}_{F^*}(\nabla F(y) \,\|\, \nabla F(x)) \\
&= F^*(\nabla F(y)) - F^*(\nabla F(x)) - \big\langle \nabla^{-1} F(\nabla F(x)), \nabla F(y) - \nabla F(x) \big\rangle \\
&= F^*(\nabla F(y)) - F^*(\nabla F(x)) - \langle x, \nabla F(y) - \nabla F(x) \rangle \\
&= \big\langle \nabla^{-1} F(\nabla F(y)), \nabla F(y) \big\rangle - F(\nabla^{-1} F(\nabla F(y))) \\
&\qquad - \big\langle \nabla^{-1} F(\nabla F(x)), \nabla F(x) \big\rangle + F(\nabla^{-1} F(\nabla F(x))) - \langle x, \nabla F(y) - \nabla F(x) \rangle \\
&= \langle y, \nabla F(y) \rangle - F(y) - \langle x, \nabla F(x) \rangle + F(x) - \langle x, \nabla F(y) - \nabla F(x) \rangle \\
&= \langle y, \nabla F(y) \rangle - F(y) + F(x) - \langle x, \nabla F(y) \rangle \\
&= F(x) - F(y) - \langle x - y, \nabla F(y) \rangle = \mathsf{B}_F(x \,\|\, y),
\end{aligned}$$

which completes the proof. □

Notice that while the unnormalized relative entropy (and thus the relative entropy) are convex functions of the pair of their arguments, this in general does not hold for all Bregman divergences, only convexity with respect to the first argument is guaranteed.

The notion of Bregman divergence can be extended to the case of non-differentiable functions (see section 14.3).

E.5 Chapter notes

The notion of entropy presented in this chapter is due to Shannon [1948] who, more generally, within the same article, set the foundation of information theory. More general definitions of entropy (*Rényi entropy*) and relative entropy (*Rényi divergence*) were later introduced by Rényi [1961]. The Kullback-Leibler divergence was introduced in [Kullback and Leibler, 1951].

Pinsker's inequality is due to Pinsker [1964]. Finer inequalities relating the relative entropy and the L_1-norm were later given by Csiszár [1967] and Kullback [1967]. See [Reid and Williamson, 2009] for a generalization of such inequalities to the case of f-divergences. The notion of Bregman divergence is due to Bregman [1967].

For a more extensive material on information theory, we strongly recommend the book of Cover and Thomas [2006].

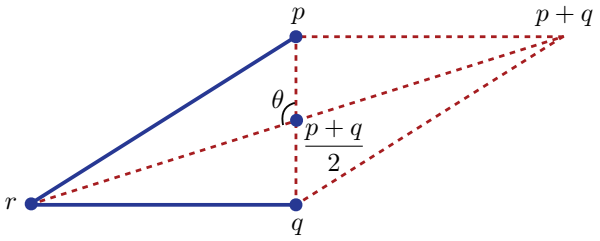


Figure E.4
An illustration of the parallelogram identity.

E.6 Exercises

E.1 Parallelogram identity. Prove the following *parallelogram identity* for any three distributions p, q, and r on $\mathcal{X}$:

$$\mathsf{D}(p\,\|\,r) + \mathsf{D}(q\,\|\,r) = 2\mathsf{D}\Big(\frac{p+q}{2}\,\|\,r\Big) + \mathsf{D}\Big(p\,\|\,\frac{p+q}{2}\Big) + \mathsf{D}\Big(q\,\|\,\frac{p+q}{2}\Big). \tag{E.34}$$

Does the equality hold if we replace the relative entropy by the norm-2 squared? Figure E.4 illustrates a particular example of this identity. Note, in the example we have

$$\begin{aligned}\|p-r\|^2 &= \big\|(p-\frac{p+q}{2}) + (\frac{p+q}{2}-r)\big\|^2 \\ &= \big\|p-\frac{p+q}{2}\big\|^2 + \big\|\frac{p+q}{2}-r\big\|^2 - 2\cos(\pi-\theta)\big\|p-\frac{p+q}{2}\big\|\big\|\frac{p+q}{2}-r\big\|\end{aligned}$$

and

$$\begin{aligned}\|q-r\|^2 &= \big\|(q-\frac{p+q}{2}) + (\frac{p+q}{2}-r)\big\|^2 \\ &= \big\|q-\frac{p+q}{2}\big\|^2 + \big\|\frac{p+q}{2}-r\big\|^2 - 2\cos(\theta)\big\|q-\frac{p+q}{2}\big\|\big\|\frac{p+q}{2}-r\big\|\,.\end{aligned}$$

Summing these two quantities shows the identity holds for the example.

F Notation

Table F.1
Summary of notation.

$\mathbb{R}$	Set of real numbers
$\mathbb{R}_+$	Set of non-negative real numbers
$\mathbb{R}^n$	Set of n-dimensional real-valued vectors
$\mathbb{R}^{n \times m}$	Set of $n \times m$ real-valued matrices
$[a, b]$	Closed interval between a and b
(a, b)	Open interval between a and b
$\{a, b, c\}$	Set containing elements a, b and c
$[n]$	The set $\{1, 2, \ldots, n\}$
$\mathbb{N}$	Set of natural numbers, i.e., $\{0, 1, \ldots\}$
$\log$	Logarithm with base e
$\log_a$	Logarithm with base a
$\mathcal{S}$	An arbitrary set
$\lvert\mathcal{S}\rvert$	Number of elements in $\mathcal{S}$
$s \in \mathcal{S}$	An element in set $\mathcal{S}$
$\mathcal{X}$	Input space
$\mathcal{Y}$	Target space
$\mathbb{H}$	Feature space
$\langle \cdot, \cdot \rangle$	Inner product in feature space
$\mathbf{v}$	An arbitrary vector
$\mathbf{1}$	Vector of all ones
v_i	ith component of $\mathbf{v}$
$\lVert\mathbf{v}\rVert$	L_2 norm of $\mathbf{v}$
$\lVert\mathbf{v}\rVert_p$	L_p norm of $\mathbf{v}$
$\mathbf{u} \circ \mathbf{v}$	Hadamard or entry-wise product of vectors $\mathbf{u}$ and $\mathbf{v}$

$f \circ g$	Composition of functions f and g
$T_1 \circ T_2$	Composition of weighted transducers T_1 and T_2
$\mathbf{M}$	An arbitrary matrix
$\|\mathbf{M}\|_2$	Spectral norm of $\mathbf{M}$
$\|\mathbf{M}\|_F$	Frobenius norm of $\mathbf{M}$
$\mathbf{M}^\top$	Transpose of $\mathbf{M}$
$\mathbf{M}^\dagger$	Pseudo-inverse of $\mathbf{M}$
$\mathrm{Tr}[\mathbf{M}]$	Trace of $\mathbf{M}$
$\mathbf{I}$	Identity matrix
$K \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	Kernel function over $\mathcal{X}$
$\mathbf{K}$	Kernel matrix
$1_{\mathcal{A}}$	Indicator function indicating membership in subset $\mathcal{A}$
h_S	The hypothesis function returned when training with sample S
$R(\cdot)$	Generalization error or risk
$\widehat{R}_S(\cdot)$	Empirical error or risk with respect to sample S
$\widehat{R}_{S,\rho}(\cdot)$	Empirical margin error with margin ρ and with respect to sample S
$\mathfrak{R}_m(\cdot)$	Rademacher complexity over all samples of size m
$\widehat{\mathfrak{R}}_S(\cdot)$	Empirical Rademacher complexity with respect to sample S
$N(0,1)$	Standard normal distribution
$\mathop{\mathbb{E}}_{x \sim \mathcal{D}}[\cdot]$	Expectation over x drawn from distribution $\mathcal{D}$
Σ^*	Kleene closure over a set of characters Σ

Bibliography

Shivani Agarwal and Partha Niyogi. Stability and generalization of bipartite ranking algorithms. In *Conference On Learning Theory*, pages 32–47, 2005.

Shivani Agarwal, Thore Graepel, Ralf Herbrich, Sariel Har-Peled, and Dan Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

Nir Ailon and Mehryar Mohri. An efficient reduction of ranking to classification. In *Conference On Learning Theory*, pages 87–98, 2008.

Mark A. Aizerman, E. M. Braverman, and Lev I. Rozonoèr. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25: 821–837, 1964.

Cyril Allauzen and Mehryar Mohri. N-way composition of weighted finite-state transducers. *International Journal of Foundations of Computer Science*, 20(4):613–627, 2009.

Cyril Allauzen, Corinna Cortes, and Mehryar Mohri. Large-scale training of SVMs with automata kernels. In *International Conference on Implementation and Application of Automata*, pages 17–27, 2010.

Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

Noga Alon and Joel Spencer. *The Probabilistic Method*. John Wiley, 1992.

Noga Alon, Shai Ben-David, Nicolò Cesa-Bianchi, and David Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of ACM*, 44:615–631, July 1997.

Yasemin Altun and Alexander J. Smola. Unifying divergence minimization and statistical inference via convex duality. In *Conference On Learning Theory*, pages 139–153, 2006.

Galen Andrew and Jianfeng Gao. Scalable training of l_1-regularized log-linear models. In *Proceedings of ICML*, pages 33–40, 2007.

Dana Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39(3):337–350, 1978.

Dana Angluin. Inference of reversible languages. *Journal of the ACM*, 29(3):741–765, 1982.

Dana Angluin and Leslie G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *J. Comput. Syst. Sci.*, 18(2):155–193, 1979.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

Patrick Assouad. Densité et dimension. *Annales de l'institut Fourier*, 33(3):233–282, 1983.

Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367, 1967.

Maria-Florina Balcan, Nikhil Bansal, Alina Beygelzimer, Don Coppersmith, John Langford, and Gregory B. Sorkin. Robust reductions from ranking to classification. *Machine Learning*, 72(1-2): 139–153, 2008.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3, 2002.

Peter L. Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Machine Learning*, 48:85–113, September 2002a.

Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Localized Rademacher complexities. In *Conference on Computational Learning Theory*, volume 2375, pages 79–97. Springer-Verlag, 2002b.

Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *Journal of the ACM*, 47:2000, 2000.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Neural Information Processing Systems*, 2001.

George Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57:33–45, 1962.

Christian Berg, Jens P.R. Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*, volume 100. Springer, 1984.

Francesco Bergadano and Stefano Varricchio. Learning behaviors of automata from shortest counterexamples. In *European Conference on Computational Learning Theory*, pages 380–391, 1995.

Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Comp. Linguistics*, 22(1), 1996.

Joseph Berkson. Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, 39:357—–365, 1944.

Sergei Natanovich Bernstein. Sur l'extension du théorème limite du calcul des probabilités aux sommes de quantités dépendantes. *Mathematische Annalen*, 97:1–59, 1927.

Dimitri P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, 1987.

Dmitri P. Bertsekas, Angelica Nedić, and Asuman E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

Laurence Bisht, Nader H. Bshouty, and Hanna Mazzawi. On optimal learning algorithms for multiplicity automata. In *Conference On Learning Theory*, pages 184–198, 2006.

Avrim Blum and Yishay Mansour. From external to internal regret. In *Conference On Learning Theory*, pages 621–636, 2005.

Avrim Blum and Yishay Mansour. Learning, regret minimization, and equilibria. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*, chapter 4, pages 4–30. Cambridge University Press, 2007.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

Jonathan Borwein and Qiji Zhu. *Techniques of Variational Analysis*. Springer, New York, 2005.

Jonathan M. Borwein and Adrian S. Lewis. *Convex Analysis and Nonlinear Optimization, Theory and Examples*. Springer, 2000.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Conference On Learning Theory*, pages 144–152, 1992.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

Leo Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11:1493–1517, October 1999.

Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

Nicolò Cesa-Bianchi. Analysis of two gradient-based algorithms for on-line regression. *Journal of Computer System Sciences*, 59(3):392–411, 1999.

Nicolò Cesa-Bianchi and Gábor Lugosi. Potential-based algorithms in online prediction and game theory. In *Conference On Learning Theory*, pages 48–64, 2001.

Nicolò Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. In *Neural Information Processing Systems*, pages 359–366, 2001.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. In *Conference On Learning Theory*, pages 217–232, 2005.

Parinya Chalermsook, Bundit Laekhanukit, and Danupon Nanongkai. Pre-reduction graph products: Hardnesses of properly learning dfas and approximating edp on dags. In *Symposium on Foundations of Computer Science*, pages 444–453. IEEE, 2014.

Bernard Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, New York, NY, USA, 2000.

Stanley F. Chen and Ronald Rosenfeld. A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, 8(1), 2000.

Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 12 1952.

Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic regression, Adaboost and Bregman distances. *Machine Learning*, 48:253–285, September 2002.

Corinna Cortes and Mehryar Mohri. AUC optimization vs. error rate minimization. In *Neural Information Processing Systems*, 2003.

Corinna Cortes and Mehryar Mohri. Confidence intervals for the area under the ROC curve. In *Neural Information Processing Systems*, volume 17, Vancouver, Canada, 2005. MIT Press.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational kernels: Theory and algorithms. *Journal of Machine Learning Research*, 5:1035–1062, 2004.

Corinna Cortes, Leonid Kontorovich, and Mehryar Mohri. Learning languages with rational kernels. In *Conference On Learning Theory*, volume 4539 of *Lecture Notes in Computer Science*, pages 349–364. Springer, Heidelberg, Germany, June 2007a.

Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. An alternative ranking problem for search engines. In *Workshop on Experimental Algorithms*, pages 1–22, 2007b.

Corinna Cortes, Mehryar Mohri, and Jason Weston. A general regression framework for learning string-to-string mappings. In *Predicted Structured Data*. MIT Press, 2007c.

Corinna Cortes, Mehryar Mohri, Dmitry Pechyony, and Ashish Rastogi. Stability of transductive regression algorithms. In *International Conference on Machine Learning*, Helsinki, Finland, July 2008a.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Learning sequence kernels. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, Cancún, Mexico, October 2008b.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Neural Information Processing Systems*, Vancouver, Canada, 2010a. MIT Press.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In *Conference on Artificial Intelligence and Statistics*, 2010b.

Corinna Cortes, Spencer Greenberg, and Mehryar Mohri. Relative deviation learning bounds and generalization with unbounded loss functions. *ArXiv 1310.5796*, October 2013. URL `http://arxiv.org/pdf/1310.5796v4.pdf`.

Corinna Cortes, Mehryar Mohri, and Umar Syed. Deep boosting. In *International Conference on Machine Learning*, pages 1179–1187, 2014.

Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Umar Syed. Structural Maxent models. In *International Conference on Machine Learning*, pages 391–399, 2015.

David Cossock and Tong Zhang. Statistical analysis of Bayes optimal subset ranking. *IEEE Transactions on Information Theory*, 54(11):5140–5154, 2008.

Thomas M. Cover and Joy M. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

Trevor F. Cox and Michael A. A. Cox. *Multidimensional Scaling*. Chapman & Hall/CRC, 2nd edition, 2000.

Koby Crammer and Yoram Singer. Improved output coding for classification using continuous relaxation. In *Neural Information Processing Systems*, 2001.

Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2, 2002.

Robert Crites and Andrew Barto. Improving elevator performance using reinforcement learning. In *Neural Information Processing Systems*, pages 1017–1023. MIT Press, 1996.

Imre Csiszár. Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*, 2:299–318, 1967.

Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39(1):1–49, 2001.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, pages 1470–1480, 1972.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, 2003.

Colin De la Higuera. *Grammatical inference: learning automata and grammars*. Cambridge University Press, 2010.

Giulia DeSalvo, Mehryar Mohri, and Umar Syed. Learning with deep cascades. In *Conference on Algorithmic Learning Theory*, pages 254–269, 2015.

Luc Devroye and Gábor Lugosi. Lower bounds in pattern recognition and learning. *Pattern Recognition*, 28(7):1011–1018, 1995.

Luc Devroye and T. J. Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979a.

Luc Devroye and T. J. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604, 1979b.

Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

Harris Drucker and Corinna Cortes. Boosting decision trees. In *Neural Information Processing Systems*, pages 479–485, 1995.

Harris Drucker, Robert E. Schapire, and Patrice Simard. Boosting performance in neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):705–719, 1993.

Miroslav Dudík, Steven J. Phillips, and Robert E. Schapire. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, 8, 2007.

Richard M. Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

Richard M. Dudley. A course on empirical processes. *Lecture Notes in Mathematics*, 1097:2 – 142, 1984.

Richard M. Dudley. Universal Donsker classes and metric entropy. *Annals of Probability*, 14(4): 1306–1326, 1987.

Richard M. Dudley. *Uniform Central Limit Theorems*. Cambridge University Press, 1999.

Nigel Duffy and David P. Helmbold. Potential boosters? In *Neural Information Processing Systems*, pages 258–264, 1999.

Aryeh Dvoretzky. On stochastic approximation. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 39–55, 1956.

Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *International World Wide Web Conference*, pages 613–622, 2001.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

James P. Egan. *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

Andrzej Ehrenfeucht, David Haussler, Michael J. Kearns, and Leslie G. Valiant. A general lower bound on the number of examples needed for learning. In *Conference On Learning Theory*, pages 139–154, 1988.

Jane Elith, Steven J. Phillips, Trevor Hastie, Miroslav Dudík, Yung En Chee, and Colin J. Yates. A statistical explanation of MaxEnt for ecologists. *Diversity and Distributions*, 1, 2011.

Eyal Even-Dar and Yishay Mansour. Learning rates for q-learning. *Machine Learning*, 5:1–25, 2003.

Dean P. Foster and Rakesh V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21:40–55, 1997.

Dean P. Foster and Rakesh V. Vohra. Asymptotic calibration. *Biometrika*, pages 379–390, 1998.

Dean P. Foster and Rakesh V. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29(1-2):7–35, 1999.

Yoav Freund. Boosting a weak learning algorithm by majority. In *Information and Computation*, pages 202–216. Morgan Kaufmann Publishers Inc., 1990.

Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121:256–285, September 1995.

Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Conference On Learning Theory*, pages 325–332, 1996.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer System Sciences*, 55(1):119–139, 1997.

Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296, 1999a.

Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, October 1999b.

Yoav Freund, Michael J. Kearns, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. Efficient learning of typical finite automata from random walks. In *Proceedings the ACM Symposium on Theory of Computing*, pages 315–324, 1993.

Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4, 2003.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

Jerome H. Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 38(2), 2000.

E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

E. Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

Joshua Goodman. Exponential priors for maximum entropy models. In *Proceedings of HLT-NAACL*, pages 305–312, 2004.

David M. Green and John A Swets. *Signal Detection Theory and Psychophysics*. Wiley, 1966.

Michelangelo Grigni, Vincent Mirelli, and Christos H Papadimitriou. On the difficulty of designing good classifiers. *SIAM Journal on Computing*, 30(1):318–323, 2000.

Adam J. Grove and Dale Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 692–699, 1998.

Uffe Haagerup. The best constants in the Khintchine inequality. *Studia Math*, 70(3):231–283, 1982.

Torben Hagerup and Christine Rüb. A guided tour of chernoff bounds. *Information Processing Letters*, 33(6):305–308, 1990.

Jihun Ham, Daniel D. Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *International Conference on Machine Learning*, 2004.

James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.

James Hannan. Approximation to Bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

Sergiu Hart and Andreu M. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

David Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217 – 232, 1995.

David Haussler. Convolution Kernels on Discrete Structures. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz, 1999.

David Haussler, Nick Littlestone, and Manfred K. Warmuth. Predicting {0,1}-functions on randomly drawn points (extended abstract). In *Symposium on Foundations of Computer Science*, pages 100–109, 1988.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, Cambridge, MA, 2000.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Klaus-Uwe Höffgen, Hans-Ulrich Simon, and Kevin S. Van Horn. Robust trainability of single neurons. *Journal of Computer and Systems Sciences*, 50(1):114–125, 1995.

John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *International Conference on Machine Learning*, pages 408–415, 2008.

Tommi Jaakkola, Michael I. Jordan, and Satinder P. Singh. Convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 6:1185–1201, 1994.

Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *ACM Special Interest Group on Information Retrieval*, pages 41–48, 2000.

E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

E. T. Jaynes. *Papers on probability, statistics, and statistical physics*. Synthese library. D. Reidel Pub. Co., 1983.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Knowledge and Discovery and Data Mining*, pages 133–142, 2002.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189—206, 1984.

Jean-Pierre Kahane. Sur les sommes vectorielles $\sum \pm u_n$. *Comptes Rendus Hebdomadaires des S'eances de l'Académie des Sciences, Paris*, 259:2577–2580, 1964.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. In *Conference On Learning Theory*, pages 26–40, 2003.

William Karush. *Minima of Functions of Several Variables with Inequalities as Side Constraints*. Master's thesis, Department of Mathematics, University of Chicago, 1939.

Jun'ichi Kazama and Jun'ichi Tsujii. Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of EMNLP*, pages 137–144, 2003.

Michael J. Kearns and Yishay Mansour. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In *International Conference on Machine Learning*, pages 269–277, 1998.

Michael J. Kearns and Yishay Mansour. On the boosting ability of top-down decision tree learning algorithms. *Journal of Computer and System Sciences*, 58(1):109–128, 1999.

Michael J. Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.

Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts (extended abstract). In *Symposium on Foundations of Computer Science*, pages 382–391, 1990.

Michael J. Kearns and Leslie G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. Technical Report 14, Harvard University, 1988.

Michael J. Kearns and Leslie G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of ACM*, 41(1):67–95, 1994.

Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

Aleksandr Khintchine. Über dyadische brüche. *Mathematische Zeitschrift*, 18(1):109–116, 1923.

Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23(1):462–466, 1952.

George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

Jyrki Kivinen and Manfred K. Warmuth. Boosting as entropy projection. In *Conference On Learning Theory*, pages 134–144, 1999.

Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

Vladimir Koltchinskii and Dmitry Panchenko. Rademacher processes and bounding the risk of function learning. In *High Dimensional Probability II*, pages 443–459. Birkhäuser, 2000.

Vladmir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30, 2002.

Leonid Kontorovich, Corinna Cortes, and Mehryar Mohri. Learning linearly separable languages. In *Algorithmic Learning Theory*, pages 288–303, 2006.

Leonid Kontorovich, Corinna Cortes, and Mehryar Mohri. Kernel methods for learning languages. *Theoretical Computer Science*, 405:223–236, 2008.

Harold W. Kuhn and Albert W. Tucker. Nonlinear programming. In *2nd Berkeley Symposium*, pages 481–492, Berkeley, 1951. University of California Press.

Solomon Kullback. A lower bound for discrimination information in terms of variation. *IEEE Transactions on Information Theory*, 13(1):126–127, 1967.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.

Harold Kushner. Stochastic approximation: a survey. *Wiley Interdisciplinary Reviews Computational Statistics*, 2(1):87–96, 2010.

Harold J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, volume 26 of *Applied Mathematical Sciences*. Springer-Verlag, 1978.

Vitaly Kuznetsov, Mehryar Mohri, and Umar Syed. Multi-class deep boosting. In *Neural Information Processing Systems*, 2014.

John Lafferty. Additive models, boosting, and inference for generalized divergences. In *Conference On Learning Theory*, pages 125–133, 1999.

John D. Lafferty, Stephen Della Pietra, and Vincent J. Della Pietra. Statistical learning algorithms based on bregman distances. In *Proceedings of the Canadian Workshop on Information Theory*, 1997.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289, 2001.

Rafał Latała and Krzysztof Oleszkiewicz. On the best constant in the khintchine-kahane inequality. *Studia Math*, 109(1):101–104, 1994.

Guy Lebanon and John D. Lafferty. Boosting and maximum likelihood for exponential models. In *Neural Information Processing Systems*, pages 447–454, 2001.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, New York, 1991.

Ehud Lehrer. A wide range no-regret theorem. *Games and Economic Behavior*, 42(1):101–115, 2003.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987.

Nick Littlestone. From on-line to batch learning. In *Conference On Learning Theory*, pages 269–284, 1989.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. In *Symposium on Foundations of Computer Science*, pages 256–261, 1989.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Michael L. Littman. *Algorithms for Sequential Decision Making*. PhD thesis, Brown University, 1996.

Philip M. Long and Rocco A. Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78:287–304, March 2010.

M. Lothaire. *Combinatorics on Words*. Cambridge University Press, 1982.

M. Lothaire. *Mots*. Hermès, 1990.

M. Lothaire. *Applied Combinatorics on Words*. Cambridge University Press, 2005.

Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of CoNLL-2002*, pages 49–55, 2002.

Christopher D. Manning and Dan Klein. Optimization, maxent models, and conditional estimation without magic. In *Proceedings of HLT-NAACL*, 2003.

Yishay Mansour and David A. McAllester. Boosting with multi-way branching in decision trees. In *Neural Information Processing Systems*, pages 300–306, 1999.

Yishay Mansour and David A. McAllester. Generalization bounds for decision trees. In *Conference On Learning Theory*, pages 69–74, 2000.

Llew Mason, Jonathan Baxter, Peter L. Bartlett, and Marcus R. Frean. Boosting algorithms as gradient descent. In *Neural Information Processing Systems*, pages 512–518, 1999.

Pascal Massart. Some applications of concentration inequalities to statistics. *Annales de la Faculté des Sciences de Toulouse*, IX:245–303, 2000.

Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society B*, 42(2), 1980.

Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman & Hall, 1983.

Colin McDiarmid. On the method of bounded differences. *Surveys in Combinatorics*, 141(1): 148–188, 1989.

Ron Meir and Gunnar Rätsch. Advanced lectures on machine learning, machine learning summer school, canberra, australia. In *Machine Learning Summer School*, pages 118–183, 2002.

Ron Meir and Gunnar Rätsch. *An Introduction to Boosting and Leveraging*, pages 118–183. Springer, 2003.

James Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209(441-458):415, 1909.

Sebastian Mika, Bernhard Scholkopf, Alex J. Smola, Klaus-Robert Muller, Matthias Scholz, and Gunnar Ratsch. Kernel PCA and de-noising in feature spaces. In *Neural Information Processing Systems*, pages 536–542, 1999.

Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

Mehryar Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002.

Mehryar Mohri. Weighted automata algorithms. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, pages 213–254. Springer, 2009.

Mehryar Mohri and Afshin Rostamizadeh. Stability bounds for stationary φ-mixing and β-mixing processes. *Journal of Machine Learning Research*, 11:789–814, 2010.

Mehryar Mohri and Afshin Rostamizadeh. Perceptron mistake bounds. *ArXiv 1305.0208*, March 2013.

Mehryar Mohri, Fernando Pereira, and Michael D. Riley. Weighted automata in text and speech processing. *European Conference on Artificial Intelligence, Workshop on Extended Finite State Models of Language*, 2005.

Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

Albert B.J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1962.

José Oncina, Pedro García, and Enrique Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458, 1993.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

Fernando C. N. Pereira and Michael D. Riley. Speech recognition by composition of weighted finite automata. In *Finite-State Language Processing*, pages 431–453. MIT Press, 1997.

Dominique Perrin. Finite automata. In J. Van Leuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 1–57. Elsevier, 1990.

Steven J. Phillips, Miroslav Dudík, and Robert E. Schapire. A maximum entropy approach to species distribution modeling. In *Proceedings of ICML*, 2004.

Steven J. Phillips, Robet P. Anderson, and Robert E. Schapire. Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, 190:231–259, 2006.

Stephen Della Pietra, Vincent J. Della Pietra, and John D. Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4), 1997.

Mark Semenovich Pinsker. *Information and Information Stability of Random Variables and Processes*. Holden-Day, 1964.

Leonard Pitt and Manfred K. Warmuth. The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of the ACM*, 40(1):95–142, 1993.

John C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods*, pages 185–208. MIT Press, 1999.

David Pollard. *Convergence of Stochastic Processess*. Springer, 1984.

David Pollard. Asymptotics via empirical processes. *Statistical Science*, 4(4):341 – 366, 1989.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Neural Information Processing Systems*, pages 1177–1184, 2007.

Adwait Ratnaparkhi. Maximum entropy models for natural language processing. In *Encyclopedia of Machine Learning*, pages 647–651. Springer, 2010.

Gunnar Rätsch and Manfred K. Warmuth. Maximizing the margin with boosting. In *Conference On Learning Theory*, pages 334–350, 2002.

Gunnar Rätsch, Sebastian Mika, and Manfred K. Warmuth. On the convergence of leveraging. In *NIPS*, pages 487–494, 2001.

Gunnar Rätsch, Takashi Onoda, and Klaus-Robert Müller. Soft margins for AdaBoost. *Machine Learning*, 42:287–320, March 2001.

Mark D. Reid and Robert C. Williamson. Generalised pinsker inequalities. In *22nd Conference on Learning Theory (COLT 2009)*, 2009.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561. University of California Press, 1961.

Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

Ryan M. Rifkin. *Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning*. PhD thesis, Massachusetts Institute of Technology, 2002.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.

R. Tyrrell Rockafellar. *Convex analysis*. Princeton University Press, 1997.

W.H. Rogers and T. J. Wagner. A finite sample distribution-free performance bound for local discrimination rules. *Annals of Statistics*, 6(3):506–514, 1978.

Dana Ron, Yoram Singer, and Naftali Tishby. On the learnability and usage of acyclic probabilistic finite automata. In *Journal of Computer and System Sciences*, pages 31–40, 1995.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

Ronald Rosenfeld. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech & Language*, 10(3):187–228, 1996.

Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323, 2000.

Cynthia Rudin, Ingrid Daubechies, and Robert E. Schapire. The dynamics of AdaBoost: Cyclic behavior and convergence of margins. *Journal of Machine Learning Research*, 5:1557–1595, 2004.

Cynthia Rudin, Corinna Cortes, Mehryar Mohri, and Robert E. Schapire. Margin-based ranking meets boosting in the middle. In *Conference On Learning Theory*, 2005.

Walter Rudin. *Fourier analysis on groups*. Number 12 in Interscience tracts in pure and applied mathematics. John Wiley & Sons, 1990.

I. N. Sanov. On the probability of large deviations of random variables. *Matematicheskii Sbornik*, 42(84):11–44, 1957.

Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13 (1):145–147, 1972.

Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *International Conference on Machine Learning*, volume 521, 1998.

Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, July 1990.

Robert E. Schapire. The boosting approach to machine learning: An overview. In *Nonlinear Estimation and Classification*, pages 149–172. Springer, 2003.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *International Conference on Machine Learning*, pages 322–330, 1997.

Leopold Schmetterer. Stochastic approximation. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pages 587–609, 1960.

Isaac J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, 1938.

Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. MIT Press, 2002.

Bernhard Schölkopf, Ralf Herbrich, Alex J. Smola, and Robert Williamson. A generalized representer theorem. Technical Report 2000-81, Neuro-COLT, 2000.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability and stability in the general learning setting. In *Conference On Learning Theory*, 2009.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1), 1972.

Satinder P. Singh. *Learning to Solve Markovian Decision Processes*. PhD thesis, University of Massachusetts, 1993.

Satinder P. Singh and Dimitri Bertsekas. Reinforcement learning for dynamic channel allocation in cellular telephone systems. In *Neural Information Processing Systems*, pages 974–980. MIT Press, 1997.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

Eric V. Slud. Distribution inequalities for the binomial law. *Annals of Probability*, 5(3):404–412, 1977.

Bharath Sriperumbudur and Zoltán Szabó. Optimal rates for random fourier features. In *Neural Information Processing Systems*, pages 1144–1152, 2015.

Gilles Stoltz and Gábor Lugosi. Internal regret in on-line portfolio selection. In *Conference On Learning Theory*, pages 403–417, 2003.

Rich Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts, 1984.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. MIT Press, 1998.

S.J. Szarek. On the best constants in the Khintchin inequality. *Studia Math*, 58(2):197–208, 1976.

Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2010.

Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. In *Conference On Learning Theory*, pages 74–89, 2002.

Benjamin Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov networks. In *Neural Information Processing Systems*, 2003.

Robert F. Tate. On a double inequality of the normal distribution. *The Annals of Mathematical Statistics*, 1:132–134, 1953.

Joshua Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Gerald Tesauro. Temporal difference learning and TD-gammon. *Communications of the ACM*, 38:58–68, March 1995.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

B. Tomaszewski. Two remarks on the Khintchine-Kahane inequality. In *Colloquium Mathematicum*, volume 46, 1982.

Boris Trakhtenbrot and Janis M. Barzdin. *Finite Automata: Behavior and Synthesis*. North-Holland, 1973.

John N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. In *Machine Learning*, volume 16, pages 185–202, 1994.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 2000.

Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 2006.

Vladimir N. Vapnik and Alexey Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.

Vladimir N. Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16:264, 1971.

Vladimir N. Vapnik and Alexey Chervonenkis. *Theory of Pattern Recognition*. Nauka, 1974.

Santosh S. Vempala. The random projection method. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 65. American Mathematical Society, 2004.

Pierre François Verhulst. Notice sur la loi que la population suit dans son accroissement. *Correspondance mathématique et physique*, 10:113—-121, 1838.

Pierre François Verhulst. Recherches mathématiques sur la loi d'accroissement de la population. *Nouveaux Mémoires de l'Académie Royale des Sciences et Belles-Lettres de Bruxelles*, 18:1—-42, 1845.

Mathukumalli Vidyasagar. *A Theory of Learning and Generalization: With Applications to Neural Networks and Control Systems*. Springer-Verlag, 1997.

Sethu Vijayakumar and Si Wu. Sequential support vector classifiers and regression. *International Conference on Soft Computing*, 1999.

John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

Vladimir G. Vovk. Aggregating strategies. In *Conference On Learning Theory*, pages 371–386, 1990.

Grace Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, 1990.

Christopher J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, 1989.

Christopher J. C. H. Watkins. Dynamic alignment kernels. Technical Report CSD-TR-98-11, Royal Holloway, University of London, 1999.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

André Weil. *L'intégration dans les groupes topologiques et ses applications*, volume 1145. Hermann Paris, 1965.

Kilian Q. Weinberger and Lawrence K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *Conference on Artificial Intelligence*, 2006.

Jason Weston and Chris Watkins. Support vector machines for multi-class pattern recognition. *European Symposium on Artificial Neural Networks*, 4(6), 1999.

Bernard Widrow and Marcian E. Hoff. Adaptive switching circuits. *Neurocomputing: Foundations of Research*, 1988.

Peter M. Williams. Bayesian regularisation and pruning using a Laplace prior. *Neural Computation*, 7:117–143, 1994.

Huan Xu, Shie Mannor, and Constantine Caramanis. Sparse algorithms are not stable: A no-free-lunch theorem. In *Conference on Communication, Control, and Computing*, pages 1299–1303, 2008.

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32:56–134, 2003a.

Tong Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Trans. Inf. Theor.*, 49(3):682–691, 2003b.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.

Index

Adaptive Computation and Machine Learning

Francis Bach, Editor

Bioinformatics: The Machine Learning Approach, Pierre Baldi and Søren Brunak

Reinforcement Learning: An Introduction, Richard S. Sutton and Andrew G. Barto

Graphical Models for Machine Learning and Digital Communication, Brendan J. Frey

Learning in Graphical Models, Michael I. Jordan

Causation, Prediction, and Search, second edition, Peter Spirtes, Clark Glymour, and Richard Scheines

Principles of Data Mining, David Hand, Heikki Mannila, and Padhraic Smyth

Bioinformatics: The Machine Learning Approach, second edition, Pierre Baldi and Soren Brunak

Learning Kernel Classifiers: Theory and Algorithms, Ralf Herbrich

Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Bernhard Schölkopf and Alexander J. Smola

Introduction to Machine Learning, Ethem Alpaydin

Gaussian Processes for Machine Learning, Carl Edward Rasmussen and Christopher K.I. Williams

Semi-Supervised Learning, Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, Eds.

The Minimum Description Length Principle, Peter D. Grünwald

Introduction to Statistical Relational Learning, Lise Getoor and Ben Taskar, Eds.

Probabilistic Graphical Models: Principles and Techniques, Daphne Koller and Nir Friedman

Introduction to Machine Learning, second edition, Ethem Alpaydin

Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation, Masashi Sugiyama and Motoaki Kawanabe

Boosting: Foundations and Algorithms, Robert E. Schapire and Yoav Freund

Machine Learning: A Probabilistic Perspective, Kevin P. Murphy

Foundations of Machine Learning, Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar

Introduction to Machine Learning, third edition, Ethem Alpaydin

Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Elements of Causal Inference, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf

Machine Learning for Data Streams, with Practical Examples in MOA, Albert Bifet, Ricard Gavaldà, Geoffrey Holmes, Bernhard Pfahringer

Foundations of Machine Learning, second edition, Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar