11 Regression

This chapter discusses in depth the learning problem of *regression*, which consists of using data to predict, as closely as possible, the correct real-valued labels of the points or items considered. Regression is a common task in machine learning with a variety of applications, which justifies the specific chapter we reserve to its analysis.

The learning guarantees presented in the previous sections focused largely on classification problems. Here we present generalization bounds for regression, both for finite and infinite hypothesis sets. Several of these learning bounds are based on the familiar notion of Rademacher complexity, which is useful for characterizing the complexity of hypothesis sets in regression as well. Others are based on a combinatorial notion of complexity tailored to regression that we will introduce, *pseudo-dimension*, which can be viewed as an extension of the VC-dimension to regression. We describe a general technique for reducing regression problems to classification and deriving generalization bounds based on the notion of pseudo-dimension. We present and analyze several regression algorithms, including *linear regression*, *kernel ridge regression*, *support-vector regression*, *Lasso*, and several on-line versions of these algorithms. We discuss in detail the properties of these algorithms, including the corresponding learning guarantees.

11.1 The problem of regression

We first introduce the learning problem of regression. Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ a measurable subset of $\mathbb{R}$. Here, we will adopt the stochastic scenario and will denote by $\mathcal{D}$ a distribution over $\mathcal{X} \times \mathcal{Y}$. As discussed in section 2.4.1, the deterministic scenario is a straightforward special case where input points admit a unique label determined by a target function $f \colon \mathcal{X} \to \mathcal{Y}$.

As in all supervised learning problems, the learner receives a labeled sample $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big) \in (\mathcal{X} \times \mathcal{Y})^m$ drawn i.i.d. according to $\mathcal{D}$. Since the labels are real numbers, it is not reasonable to hope that the learner could predict precisely the correct label when it is unique, or precisely its average label. Instead, we can require that its predictions be close to the correct ones. This is the key difference between regression and classification: in regression, the measure of error is based on the magnitude of the difference between the real-valued label predicted and the true or correct one, and not based on the equality or inequality of these two values. We denote by $L\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ the *loss function* used to measure the magnitude of error. The most common loss function used in regression is the *squared loss* L_2 defined by $L(y, y') = |y' - y|^2$ for all $y, y' \in \mathcal{Y}$, or, more generally, an L_p loss defined by $L(y, y') = |y' - y|^p$, for some $p \geq 1$ and all $y, y' \in \mathcal{Y}$.

Given a hypothesis set $\mathcal{H}$ of functions mapping $\mathcal{X}$ to $\mathcal{Y}$, the regression problem consists of using the labeled sample S to find a hypothesis $h \in \mathcal{H}$ with small expected loss or generalization error $R(h)$ with respect to the target f:

$$R(h) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \big[L\big(h(x), y\big)\big] . \tag{11.1}$$

As in the previous chapters, the empirical loss or error of $h \in \mathcal{H}$ is denoted by $\widehat{R}_S(h)$ and defined by

$$\widehat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} L\big(h(x_i), y_i\big) . \tag{11.2}$$

In the common case where L is the squared loss, this represents the *mean squared error* of h on the sample S.

When the loss function L is bounded by some $M > 0$, that is $L(y', y) \leq M$ for all $y, y' \in \mathcal{Y}$ or, more strictly, $L(h(x), y) \leq M$ for all $h \in \mathcal{H}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the problem is referred to as a *bounded regression problem.* Much of the theoretical results presented in the following sections are based on that assumption. The analysis of *unbounded regression problems* is technically more elaborate and typically requires some other types of assumptions.

11.2 Generalization bounds

This section presents learning guarantees for bounded regression problems. We start with the simple case of a finite hypothesis set.

11.2.1 Finite hypothesis sets

In the case of a finite hypothesis, we can derive a generalization bound for regression by a straightforward application of Hoeffding's inequality and the union bound.

Theorem 11.1 *Let L be a bounded loss function. Assume that the hypothesis set $\mathcal{H}$ is finite. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality* *holds for all* $h \in \mathcal{H}$*:*

$$R(h) \leq \widehat{R}_S(h) + M\sqrt{\frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{2m}}\,.$$

Proof: By Hoeffding's inequality, since L takes values in $[0, M]$, for any $h \in \mathcal{H}$, the following holds:

$$\mathbb{P}\Big[R(h) - \widehat{R}_S(h) > \epsilon\Big] \leq e^{-\frac{2m\epsilon^2}{M^2}}\,.$$

Thus, by the union bound, we can write

$$\mathbb{P}\Big[\exists h \in \mathcal{H}\colon R(h) - \widehat{R}_S(h) > \epsilon\Big] \leq \sum_{h\in\mathcal{H}} \mathbb{P}\Big[R(h) - \widehat{R}_S(h) > \epsilon\Big] \leq |\mathcal{H}|e^{-\frac{2m\epsilon^2}{M^2}}\,.$$

Setting the right-hand side to be equal to δ yields the statement of the theorem. □

With the same assumptions and using the same proof, a two-sided bound can be derived: with probability at least $1-\delta$, for all $h \in \mathcal{H}$,

$$|R(h) - \widehat{R}_S(h)| \leq M\sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{2m}}\,.$$

These learning bounds are similar to those derived for classification. In fact, they coincide with the classification bounds given in the inconsistent case when $M = 1$. Thus, all the remarks made in that context apply identically here. In particular, a larger sample size m guarantees better generalization; the bound increases as a function of $\log|\mathcal{H}|$ and suggests selecting, for the same empirical error, a smaller hypothesis set. This is an instance of Occam's razor principle for regression. In the next sections, we present other instances of this principle for the general case of infinite hypothesis sets using the notions of Rademacher complexity and pseudo-dimension.

11.2.2 Rademacher complexity bounds

Here, we show how the Rademacher complexity bounds of theorem 3.3 can be used to derive generalization bounds for regression in the case of the family of L_p loss functions. We first show an upper bound for the Rademacher complexity of a relevant family of functions.

Proposition 11.2 (Rademacher complexity of μ-Lipschitz loss functions) *Let* $L\colon \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$ *be a non-negative loss upper bounded by* $M > 0$ *(*$L(y, y') \leq M$ *for all* $y, y' \in \mathcal{Y}$*) and such that for any fixed* $y' \in \mathcal{Y}$*,* $y \mapsto L(y, y')$ *is* μ*-Lipschitz for some* $\mu > 0$*. Then, for any sample* $S = ((x_1, y_1), \ldots, (x_m, y_m))$*, the Rademacher complexity of the family* $\mathcal{G} = \{(x, y) \mapsto L(h(x), y)\colon h \in \mathcal{H}\}$ *is upper bounded as follows:*

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) \leq \mu\,\widehat{\mathfrak{R}}_S(\mathcal{H})\,.$$

Proof: Since for any fixed y_i, $y \mapsto L(y, y_i)$ is μ-Lipschitz, by Talagrand's contraction lemma (lemma 5.7), we can write

$$\widehat{R}_S(\mathcal{G}) = \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sum_{i=1}^{m} \sigma_i L(h(x_i), y_i) \right] \leq \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sum_{i=1}^{m} \sigma_i \mu \, h(x_i) \right] = \mu \, \widehat{\mathfrak{R}}_S(\mathcal{H}),$$

which completes the proof. □

Theorem 11.3 (Rademacher complexity regression bounds) *Let $L \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a non-negative loss upper bounded by $M > 0$ ($L(y, y') \leq M$ for all $y, y' \in \mathcal{Y}$) and such that for any fixed $y' \in \mathcal{Y}$, $y \mapsto L(y, y')$ is μ-Lipschitz for some $\mu > 0$.*

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \big[L(x, y)\big] \leq \frac{1}{m} \sum_{i=1}^{m} L(x_i, y_i) + 2\mu \, \mathfrak{R}_m(\mathcal{H}) + M \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \big[L(x, y)\big] \leq \frac{1}{m} \sum_{i=1}^{m} L(x_i, y_i) + 2\mu \, \widehat{\mathfrak{R}}_S(\mathcal{H}) + 3M \sqrt{\frac{\log \frac{2}{\delta}}{2m}}\,.$$

Proof: Since for any fixed y_i, $y \mapsto L(y, y_i)$ is μ-Lipschitz, by Talagrand's contraction lemma (lemma 5.7), we can write

$$\widehat{R}_S(\mathcal{G}) = \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sum_{i=1}^{m} \sigma_i L(h(x_i), y_i) \right] \leq \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sum_{i=1}^{m} \sigma_i \mu \, h(x_i) \right] = \mu \, \widehat{\mathfrak{R}}_S(\mathcal{H}).$$

Combining this inequality with the general Rademacher complexity learning bound of theorem 3.3 completes the proof. □

Let $p \geq 1$ and assume that $|h(x) - y| \leq M$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $h \in \mathcal{H}$. Then, since for any y' the function $y \mapsto |y - y'|^p$ is pM^{p-1}-Lipschitz for $(y - y') \in [-M, M]$, the theorem applies to any L_p-loss. As an example, for any $\delta > 0$, with probability at least $1 - \delta$ over a sample S of size m, each of the following inequalities holds for all $h \in \mathcal{H}$:

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \Big[\big|h(x) - y\big|^p\Big] \leq \frac{1}{m} \sum_{i=1}^{m} \big|h(x_i) - y_i\big|^p + 2pM^{p-1} \mathfrak{R}_m(\mathcal{H}) + M^p \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

As in the case of classification, these generalization bounds suggest a trade-off between reducing the empirical error, which may require more complex hypothesis sets, and controlling the Rademacher complexity of $\mathcal{H}$, which may increase the empirical error. An important benefit of the last learning bound of the theorem is that it is data-dependent. This can lead to more accurate learning guarantees. The upper bounds on $\mathfrak{R}_m(\mathcal{H})$ or $\mathfrak{R}_S(\mathcal{H})$ for kernel-based hypotheses (theorem 6.12) can be used directly here to derive generalization bounds in terms of the trace of the kernel matrix or the maximum diagonal entry.

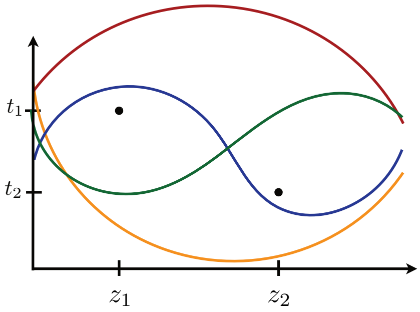

Figure 11.1
Illustration of the shattering of a set of two points $\{z_1, z_2\}$ with witnesses t_1 and t_2.

11.2.3 Pseudo-dimension bounds

As previously discussed in the case of classification, it is sometimes computationally hard to estimate the empirical Rademacher complexity of a hypothesis set. In chapter 3, we introduce other measures of the complexity of a hypothesis set such as the VC-dimension, which are purely combinatorial and typically easier to compute or upper bound. However, the notion of shattering or that of VC-dimension introduced for binary classification are not readily applicable to real-valued hypothesis classes.

We first introduce a new notion of *shattering* for families of real-valued functions. As in previous chapters, we will use the notation $\mathcal{G}$ for a family of functions, whenever we intend to later interpret it (at least in some cases) as the family of loss functions associated to some hypothesis set $\mathcal{H}$: $\mathcal{G} = \{z = (x, y) \mapsto L(h(x), y) \colon h \in \mathcal{H}\}$.

Definition 11.4 (Shattering) *Let $\mathcal{G}$ be a family of functions from a set $\mathcal{Z}$ to $\mathbb{R}$. A set $\{z_1, \ldots, z_m\} \subseteq \mathcal{X}$ is said to be* shattered *by $\mathcal{G}$ if there exist $t_1, \ldots, t_m \in \mathbb{R}$ such that,*

$$\left| \left\{ \begin{bmatrix} \operatorname{sgn}\big(g(z_1) - t_1\big) \\ \vdots \\ \operatorname{sgn}\big(g(z_m) - t_m\big) \end{bmatrix} : g \in \mathcal{G} \right\} \right| = 2^m .$$

When they exist, the threshold values $t_1, \ldots, t_m$ are said to witness *the shattering.*

Thus, $\{z_1, \ldots, z_m\}$ is shattered if for some witnesses $t_1, \ldots, t_m$, the family of functions $\mathcal{G}$ is rich enough to contain a function going above a subset $\mathcal{A}$ of the set of points $\mathcal{I} = \{(z_i, t_i) \colon i \in [m]\}$ and below the others $(\mathcal{I} - \mathcal{A})$, for any choice of the subset $\mathcal{A}$. Figure 11.1 illustrates this shattering in a simple case. The notion of shattering naturally leads to the following definition.

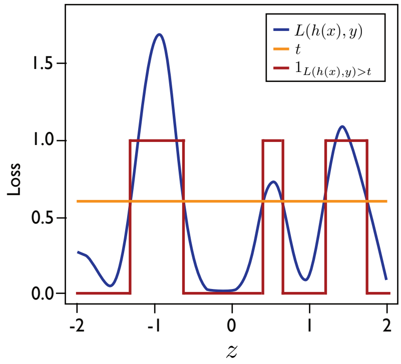

Figure 11.2
A function $g\colon z = (x, y) \mapsto L(h(x), y)$ (in blue) defined as the loss of some fixed hypothesis $h \in \mathcal{H}$, and its thresholded version $(x, y) \mapsto 1_{L(h(x),y)>t}$ (in red) with respect to the threshold t (in yellow).

Definition 11.5 (Pseudo-dimension) *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{X}$ to $\mathbb{R}$. Then, the* pseudo-dimension of $\mathcal{G}$, *denoted by* $\mathrm{Pdim}(\mathcal{G})$, *is the size of the largest set shattered by $\mathcal{G}$.*

By definition of the shattering just introduced, the notion of pseudo-dimension of a family of real-valued functions $\mathcal{G}$ coincides with that of the VC-dimension of the corresponding thresholded functions mapping $\mathcal{X}$ to $\{0, 1\}$:

$$\mathrm{Pdim}(\mathcal{G}) = \mathrm{VCdim}\Big(\big\{(x, t) \mapsto 1_{(g(x)-t)>0}\colon g \in \mathcal{G}\big\}\Big). \tag{11.3}$$

Figure 11.2 illustrates this interpretation. In view of this interpretation, the following two results follow directly the properties of the VC-dimension.

Theorem 11.6 *The pseudo-dimension of hyperplanes in $\mathbb{R}^N$ is given by*

$$\mathrm{Pdim}(\{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} + b\colon \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}) = N + 1.$$

Theorem 11.7 *The pseudo-dimension of a vector space of real-valued functions $\mathcal{H}$ is equal to the dimension of the vector space:*

$$\mathrm{Pdim}(\mathcal{H}) = \dim(\mathcal{H}).$$

The following theorem gives a generalization bound for bounded regression in terms of the pseudo-dimension of a family of loss function $\mathcal{G} = \{z = (x, y) \mapsto L(h(x), y)\colon h \in \mathcal{H}\}$ associated to a hypothesis set $\mathcal{H}$. The key technique to derive these bounds consists of reducing the problem to that of classification by making use of the following general identity for the expectation of a random variable X:

$$\mathbb{E}[X] = -\int_{-\infty}^{0} \mathbb{P}[X < t]dt + \int_{0}^{+\infty} \mathbb{P}[X > t]dt, \tag{11.4}$$

which holds by definition of the Lebesgue integral. In particular, for any distribution $\mathcal{D}$ and any non-negative measurable function f, we can write

$$\underset{z\sim\mathcal{D}}{\mathbb{E}}[f(z)] = \int_0^\infty \underset{z\sim\mathcal{D}}{\mathbb{P}}[f(z) > t]dt\,. \tag{11.5}$$

Theorem 11.8 *Let $\mathcal{H}$ be a family of real-valued functions and $\mathcal{G} = \{(x, y) \mapsto L(h(x), y) \colon h \in \mathcal{H}\}$ the family of loss functions associated to $\mathcal{H}$. Assume that* $\operatorname{Pdim}(\mathcal{G}) = d$ *and that the loss function L is non-negative and bounded by M. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of am i.i.d. sample S of size m drawn from $\mathcal{D}^m$, the following inequality holds for all $h \in \mathcal{H}$:*

$$R(h) \le \widehat{R}_S(h) + M\sqrt{\frac{2d\log\frac{em}{d}}{m}} + M\sqrt{\frac{\log\frac{1}{\delta}}{2m}}\,. \tag{11.6}$$

Proof: Let S be a sample of size m drawn i.i.d. according to $\mathcal{D}$ and let $\widehat{\mathcal{D}}$ denote the empirical distribution defined by S. For any $h \in \mathcal{H}$ and $t \ge 0$, we denote by $c(h, t)$ the classifier defined by $c(h, t) \colon (x, y) \mapsto 1_{L(h(x),y)>t}$. The error of $c(h, t)$ can be defined by

$$R(c(h,t)) = \underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}[c(h,t)(x,y) = 1] = \underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}[L(h(x),y) > t],$$

and, similarly, its empirical error is $\widehat{R}_S(c(h,t)) = \mathbb{P}_{(x,y)\sim\widehat{\mathcal{D}}}[L(h(x),y) > t]$.

Now, in view of the identity (11.5) and the fact that the loss function L is bounded by M, we can write:

$$\begin{aligned}
|R(h) - \widehat{R}_S(h)| &= \Big|\underset{(x,y)\sim\mathcal{D}}{\mathbb{E}}[L(h(x),y)] - \underset{(x,y)\sim\widehat{\mathcal{D}}}{\mathbb{E}}[L(h(x),y)]\Big| \\
&= \left|\int_0^M \left(\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}[L(h(x),y) > t] - \underset{(x,y)\sim\widehat{\mathcal{D}}}{\mathbb{P}}[L(h(x),y) > t]\right) dt\right| \\
&\le M \sup_{t\in[0,M]} \left|\underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}[L(h(x),y) > t] - \underset{(x,y)\sim\widehat{\mathcal{D}}}{\mathbb{P}}[L(h(x),y) > t]\right| \\
&= M \sup_{t\in[0,M]} \left|R(c(h,t)) - \widehat{R}_S(c(h,t))\right|.
\end{aligned}$$

This implies the following inequality:

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}} |R(h) - \widehat{R}_S(h)| > \epsilon\right] \le \mathbb{P}\left[\sup_{\substack{h\in\mathcal{H}\\ t\in[0,M]}} \left|R(c(h,t)) - \widehat{R}_S(c(h,t))\right| > \frac{\epsilon}{M}\right].$$

The right-hand side can be bounded using a standard generalization bound for classification (corollary 3.19) in terms of the VC-dimension of the family of hy-

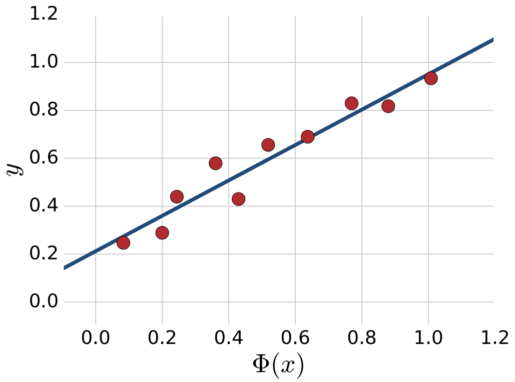

Figure 11.3
For $N = 1$, linear regression consists of finding the line of best fit, measured in terms of the squared loss.

potheses $\{c(h, t)\colon h \in \mathcal{H}, t \in [0, M]\}$, which, by definition of the pseudo-dimension, is precisely $\text{Pdim}(\mathcal{G}) = d$. The resulting bound coincides with (11.6). □

The notion of pseudo-dimension is suited to the analysis of regression as demonstrated by the previous theorem; however, it is not a scale-sensitive notion. There exists an alternative complexity measure, the *fat-shattering dimension*, that is scale-sensitive and that can be viewed as a natural extension of the pseudo-dimension. Its definition is based on the notion of γ-shattering.

Definition 11.9 (γ-shattering) *Let $\mathcal{G}$ be a family of functions from $\mathcal{Z}$ to $\mathbb{R}$ and let $\gamma > 0$. A set $\{z_1, \ldots, z_m\} \subseteq \mathcal{X}$ is said to be* γ-shattered *by $\mathcal{G}$ if there exist $t_1, \ldots, t_m \in \mathbb{R}$ such that for all $\mathbf{y} \in \{-1, +1\}^m$, there exists $g \in \mathcal{G}$ such that:*

$$\forall i \in [m], \quad y_i(g(z_i) - t_i) \geq \gamma\,.$$

Thus, $\{z_1, \ldots, z_m\}$ is γ-shattered if for some witnesses $t_1, \ldots, t_m$, the family of functions $\mathcal{G}$ is rich enough to contain a function going at least γ above a subset $\mathcal{A}$ of the set of points $\mathcal{I} = \{(z_i, t_i)\colon i \in [m]\}$ and at least γ below the others $(\mathcal{I} - \mathcal{A})$, for any choice of the subset $\mathcal{A}$.

Definition 11.10 (γ-fat-dimension) *The* γ-fat-dimension *of $\mathcal{G}$*, $\text{fat}_\gamma(\mathcal{G})$, *is the size of the largest set that is γ-shattered by $\mathcal{G}$.*

Finer generalization bounds than those based on the pseudo-dimension can be derived in terms of the γ-fat-dimension. However, the resulting learning bounds, are not more informative than those based on the Rademacher complexity, which is also a scale-sensitive complexity measure. Thus, we will not detail an analysis based on the γ-fat-dimension.

11.3 Regression algorithms

The results of the previous sections show that, for the same empirical error, hypothesis sets with smaller complexity measured in terms of the Rademacher complexity or in terms of pseudo-dimension benefit from better generalization guarantees. One family of functions with relatively small complexity is that of linear hypotheses. In this section, we describe and analyze several algorithms based on that hypothesis set: *linear regression*, *kernel ridge regression* (KRR), *support vector regression* (SVR), and *Lasso*. These algorithms, in particular the last three, are extensively used in practice and often lead to state-of-the-art performance results.

11.3.1 Linear regression

We start with the simplest algorithm for regression known as *linear regression*. Let $\mathbf{\Phi}\colon \mathcal{X} \to \mathbb{R}^N$ be a feature mapping from the input space $\mathcal{X}$ to $\mathbb{R}^N$ and consider the family of linear hypotheses

$$\mathcal{H} = \{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x) + b \colon \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}. \tag{11.7}$$

Linear regression consists of seeking a hypothesis in $\mathcal{H}$ with the smallest empirical mean squared error. Thus, for a sample $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big) \in (\mathcal{X} \times \mathcal{Y})^m$, the following is the corresponding optimization problem:

$$\min_{\mathbf{w}, b} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b - y_i)^2 . \tag{11.8}$$

Figure 11.3 illustrates the algorithm in the simple case where $N = 1$. The optimization problem admits the simpler formulation:

$$\min_{\mathbf{W}} F(\mathbf{W}) = \frac{1}{m} \|\mathbf{X}^\top \mathbf{W} - \mathbf{Y}\|^2, \tag{11.9}$$

using the notation $\mathbf{X} = \left[\begin{smallmatrix} \Phi(x_1) & \ldots & \Phi(x_m) \\ 1 & \ldots & 1 \end{smallmatrix}\right]$, $\mathbf{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \\ b \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$. The objective function F is convex, by composition of the convex function $\mathbf{u} \mapsto \|\mathbf{u}\|^2$ with the affine function $\mathbf{W} \mapsto \mathbf{X}^\top \mathbf{W} - \mathbf{Y}$, and it is differentiable. Thus, F admits a global minimum at $\mathbf{W}$ if and only if $\nabla F(\mathbf{W}) = 0$, that is if and only if

$$\frac{2}{m} \mathbf{X}(\mathbf{X}^\top \mathbf{W} - \mathbf{Y}) = 0 \Leftrightarrow \mathbf{X}\mathbf{X}^\top \mathbf{W} = \mathbf{X}\mathbf{Y}. \tag{11.10}$$

When $\mathbf{X}\mathbf{X}^\top$ is invertible, this equation admits a unique solution. Otherwise, the equation admits a family of solutions that can be given in terms of the pseudo-inverse of matrix $\mathbf{X}\mathbf{X}^\top$ (see appendix A) by $\mathbf{W} = (\mathbf{X}\mathbf{X}^\top)^\dagger \mathbf{X}\mathbf{Y} + (I - (\mathbf{X}\mathbf{X}^\top)^\dagger (\mathbf{X}\mathbf{X}^\top))\mathbf{W}_0$, where $\mathbf{W}_0$ is an arbitrary matrix in $\mathbb{R}^{N \times N}$. Among these, the solution $\mathbf{W} = (\mathbf{X}\mathbf{X}^\top)^\dagger \mathbf{X}\mathbf{Y}$ is the one with the minimal norm and is often preferred for that reason. Thus, we will write the solutions as

$$\mathbf{W} = \begin{cases} (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{Y} & \text{if } \mathbf{X}\mathbf{X}^\top \text{ is invertible,} \\ (\mathbf{X}\mathbf{X}^\top)^\dagger \mathbf{X}\mathbf{Y} & \text{otherwise.} \end{cases} \tag{11.11}$$

The matrix $\mathbf{X}\mathbf{X}^\top$ can be computed in $O(mN^2)$. The cost of its inversion or that of computing its pseudo-inverse is in $O(N^3)$.[19] Finally, the multiplication with $\mathbf{X}$ and $\mathbf{Y}$ takes $O(mN^2)$. Therefore, the overall complexity of computing the solution $\mathbf{W}$ is in $O(mN^2 + N^3)$. Thus, when the dimension of the feature space N is not too large, the solution can be computed efficiently.

While linear regression is simple and admits a straightforward implementation, it does not benefit from a strong generalization guarantee, since it is limited to minimizing the empirical error without controlling the norm of the weight vector and without any other regularization. Its performance is also typically poor in most applications. The next sections describe algorithms with both better theoretical guarantees and improved performance in practice.

11.3.2 Kernel ridge regression

We first present a learning guarantee for regression with bounded linear hypotheses in a feature space defined by a PDS kernel. This will provide a strong theoretical support for the *kernel ridge regression* algorithm presented in this section. The learning bounds of this section are given for the squared loss. Thus, in particular, the generalization error of a hypothesis h is defined by $R(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[(h(x) - y)^2\big]$.

Theorem 11.11 *Let $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PDS kernel, $\Phi\colon \mathcal{X} \to \mathbb{H}$ a feature mapping associated to K, and $\mathcal{H} = \{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x) \colon \|\mathbf{w}\|_{\mathbb{H}} \le \Lambda\}$. Assume that there exists $r > 0$ such that $K(x,x) \le r^2$ and $M > 0$ such that $|h(x) - y| < M$ for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, each of the following inequalities holds for all $h \in \mathcal{H}$:*

$$R(h) \le \widehat{R}_S(h) + 4M\sqrt{\frac{r^2\Lambda^2}{m}} + M^2\sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

$$R(h) \le \widehat{R}_S(h) + \frac{4M\Lambda\sqrt{\mathrm{Tr}[\mathbf{K}]}}{m} + 3M^2\sqrt{\frac{\log\frac{2}{\delta}}{2m}}\,.$$

[19] In the analysis of the computational complexity of the algorithms discussed in this chapter, the cubic-time complexity of matrix inversion can be replaced by a more favorable complexity $O(N^{2+\omega})$, with $\omega = .376$ using asymptotically faster matrix inversion methods such as that of Coppersmith and Winograd.

Proof: By the bound on the empirical Rademacher complexity of kernel-based hypotheses (theorem 6.12), the following holds for any sample S of size m:

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) \leq \frac{\Lambda\sqrt{\mathrm{Tr}[\mathbf{K}]}}{m} \leq \sqrt{\frac{r^2\Lambda^2}{m}},$$

which implies that $\mathfrak{R}_m(\mathcal{H}) \leq \sqrt{\frac{r^2\Lambda^2}{m}}$. Combining these inequalities with the learning bounds of Theorem 11.3 yield immediately the inequalities claimed. □

The learning bounds of the theorem suggests minimizing a trade-off between the empirical squared loss (first term on the right-hand side), and the norm of the weight vector (upper bound Λ on the norm appearing in the second term), or equivalently the norm squared. Kernel ridge regression is defined by the minimization of an objective function that has precisely this form and thus is directly motivated by the theoretical analysis just presented:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{m} \left(\mathbf{w} \cdot \mathbf{\Phi}(x_i) - y_i\right)^2. \tag{11.12}$$

Here, λ is a positive parameter determining the trade-off between the regularization term $\|\mathbf{w}\|^2$ and the empirical mean squared error. The objective function differs from that of linear regression only by the first term, which controls the norm of $\mathbf{w}$. As in the case of linear regression, the problem can be rewritten in a more compact form as

$$\min_{\mathbf{W}} F(\mathbf{W}) = \lambda\|\mathbf{W}\|^2 + \|\mathbf{X}^\top\mathbf{W} - \mathbf{Y}\|^2, \tag{11.13}$$

where $\mathbf{X} \in \mathbb{R}^{N\times m}$ is the matrix formed by the feature vectors, $\mathbf{X} = \left[\begin{smallmatrix}\Phi(x_1) & \dots & \Phi(x_m)\end{smallmatrix}\right]$, $\mathbf{W} = \mathbf{w}$, and $\mathbf{Y} = (y_1, \ldots, y_m)^\top$. Here too, F is convex, by the convexity of $\mathbf{w} \mapsto \|\mathbf{w}\|^2$ and that of the sum of two convex functions, and is differentiable. Thus F admits a global minimum at $\mathbf{W}$ if and only if

$$\nabla F(\mathbf{W}) = 0 \Leftrightarrow (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{W} = \mathbf{X}\mathbf{Y} \Leftrightarrow \mathbf{W} = (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{Y}. \tag{11.14}$$

Note that the matrix $\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}$ is always invertible, since its eigenvalues are the sum of the non-negative eigenvalues of the symmetric positive semidefinite matrix $\mathbf{X}\mathbf{X}^\top$ and $\lambda > 0$. Thus, kernel ridge regression admits a closed-form solution.

An alternative formulation of the optimization problem for kernel ridge regression equivalent to (11.12) is

$$\min_{\mathbf{w}} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{\Phi}(x_i) - y_i)^2 \quad \text{subject to: } \|\mathbf{w}\|^2 \leq \Lambda^2.$$

This makes the connection with the bounded linear hypothesis set of theorem 11.11 even more evident. Using slack variables ξ_i, for all $i \in [m]$, the problem can be equivalently written as

$$\min_{\mathbf{w}} \sum_{i=1}^{m} \xi_i^2 \quad \text{subject to: } (\|\mathbf{w}\|^2 \leq \Lambda^2) \wedge \big(\forall i \in [m],\ \xi_i = y_i - \mathbf{w} \cdot \mathbf{\Phi}(x_i)\big).$$

This is a convex optimization problem with differentiable objective function and constraints. To derive the equivalent dual problem, we introduce the Lagrangian $\mathcal{L}$, which is defined for all $\boldsymbol{\xi}, \mathbf{w}, \boldsymbol{\alpha}'$, and $\lambda \geq 0$ by

$$\mathcal{L}(\boldsymbol{\xi}, \mathbf{w}, \boldsymbol{\alpha}', \lambda) = \sum_{i=1}^{m} \xi_i^2 + \sum_{i=1}^{m} \alpha'_i(y_i - \xi_i - \mathbf{w} \cdot \mathbf{\Phi}(x_i)) + \lambda(\|\mathbf{w}\|^2 - \Lambda^2)\,.$$

The KKT conditions lead to the following equalities:

$$\begin{aligned}
&\nabla_{\mathbf{w}} \mathcal{L} = -\sum_{i=1}^{m} \alpha'_i \mathbf{\Phi}(x_i) + 2\lambda \mathbf{w} = 0 && \implies && \mathbf{w} = \frac{1}{2\lambda} \sum_{i=1}^{m} \alpha'_i \mathbf{\Phi}(x_i)\\
&\nabla_{\xi_i} \mathcal{L} = 2\xi_i - \alpha'_i = 0 && \implies && \xi_i = \alpha'_i/2\\
&\forall i \in [m], \alpha'_i(y_i - \xi_i - \mathbf{w} \cdot \mathbf{\Phi}(x_i)) = 0\\
&\lambda(\|\mathbf{w}\|^2 - \Lambda^2) = 0.
\end{aligned}$$

Plugging in the expressions of $\mathbf{w}$ and ξ_is in that of $\mathcal{L}$ gives

$$\begin{aligned}
\mathcal{L} &= \sum_{i=1}^{m} \frac{\alpha'^2_i}{4} + \sum_{i=1}^{m} \alpha'_i y_i - \sum_{i=1}^{m} \frac{\alpha'^2_i}{2} - \frac{1}{2\lambda} \sum_{i,j=1}^{m} \alpha'_i \alpha'_j \mathbf{\Phi}(x_i)^\top \mathbf{\Phi}(x_j)\\
&\quad + \lambda\Big(\frac{1}{4\lambda^2} \| \sum_{i=1}^{m} \alpha'_i \mathbf{\Phi}(x_i)\|^2 - \Lambda^2\Big)\\
&= -\frac{1}{4} \sum_{i=1}^{m} \alpha'^2_i + \sum_{i=1}^{m} \alpha'_i y_i - \frac{1}{4\lambda} \sum_{i,j=1}^{m} \alpha'_i \alpha'_j \mathbf{\Phi}(x_i)^\top \mathbf{\Phi}(x_j) - \lambda \Lambda^2\\
&= -\lambda \sum_{i=1}^{m} \alpha_i^2 + 2 \sum_{i=1}^{m} \alpha_i y_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j \mathbf{\Phi}(x_i)^\top \mathbf{\Phi}(x_j) - \lambda \Lambda^2,
\end{aligned}$$

with $\alpha'_i = 2\lambda\alpha_i$. Thus, the equivalent dual optimization problem for KRR can be written as follows:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{Y} - \boldsymbol{\alpha}^\top (\mathbf{X}^\top \mathbf{X}) \boldsymbol{\alpha}\,, \tag{11.15}$$

or, more compactly, as

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} G(\boldsymbol{\alpha}) = -\boldsymbol{\alpha}^\top (\mathbf{K} + \lambda \mathbf{I}) \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{Y}\,, \tag{11.16}$$

where $\mathbf{K} = \mathbf{X}^\top \mathbf{X}$ is the kernel matrix associated to the training sample. The objective function G is concave and differentiable. The optimal solution is obtained by differentiating the function and setting it to zero:

$$\nabla G(\boldsymbol{\alpha}) = 0 \iff 2(\mathbf{K} + \lambda \mathbf{I})\boldsymbol{\alpha} = 2\mathbf{Y} \iff \boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{Y}\,. \tag{11.17}$$

Note that $(\mathbf{K}+\lambda\mathbf{I})$ is invertible, since its eigenvalues are the sum of the eigenvalues of the SPSD matrix $\mathbf{K}$ and $\lambda > 0$. Thus, as in the primal case, the dual optimization problem admits a closed-form solution. By the first KKT equation, $\mathbf{w}$ can be determined from $\boldsymbol{\alpha}$ by

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \boldsymbol{\Phi}(\mathbf{x}_i) = \mathbf{X}\boldsymbol{\alpha} = \mathbf{X}(\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{Y}. \tag{11.18}$$

The hypothesis h solution can be given as follows in terms of $\boldsymbol{\alpha}$:

$$\forall x \in \mathcal{X}, \quad h(x) = \mathbf{w} \cdot \boldsymbol{\Phi}(x) = \sum_{i=1}^{m} \alpha_i K(x_i, x)\,. \tag{11.19}$$

Note that the form of the solution, $h = \sum_{i=1}^{m} \alpha_i K(x_i, \cdot)$, could be immediately predicted using the Representer theorem, since the objective function minimized by KRR falls within the general framework of theorem 6.11. This also could show that $\mathbf{w}$ could be written as $\mathbf{w} = \mathbf{X}\alpha$. This fact, combined with the following simple lemma, can be used to determine $\boldsymbol{\alpha}$ in a straightforward manner, without the intermediate derivation of the dual problem.

Lemma 11.12 *The following identity holds for any matrix* $\mathbf{X}$*:*

$$(\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1}\mathbf{X} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda \mathbf{I})^{-1}\,.$$

Proof: Observe that $(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{X} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})$. Left-multiplying by $(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}$ this equality and right-multiplying it by $(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}$ yields the statement of the lemma. □

Now, using this lemma, the primal solution of $\mathbf{w}$ can be rewritten as follows:

$$\mathbf{w} = (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{Y} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{Y} = \mathbf{X}(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{Y}.$$

Comparing with $\mathbf{w} = \mathbf{X}\alpha$ gives immediately $\boldsymbol{\alpha} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{Y}$.

Our presentation of the KRR algorithm was given for linear hypotheses with no offset, that is we implicitly assumed $b = 0$. It is common to use this formulation and to extend it to the general case by augmenting the feature vector $\boldsymbol{\Phi}(x)$ with an extra component equal to one for all $x \in \mathcal{X}$ and the weight vector $\mathbf{w}$ with an extra component $b \in \mathbb{R}$. For the augmented feature vector $\boldsymbol{\Phi}'(x) \in \mathbb{R}^{N+1}$ and weight vector $\mathbf{w}' \in \mathbb{R}^{N+1}$, we have $\mathbf{w}' \cdot \boldsymbol{\Phi}'(x) = \mathbf{w} \cdot \boldsymbol{\Phi}(x) + b$. Nevertheless, this formulation does not coincide with the general KRR algorithm where a solution of the form $x \mapsto \mathbf{w} \cdot \boldsymbol{\Phi}(x) + b$ is sought. This is because for the general KRR, the regularization term is $\lambda\|\mathbf{w}\|$, while for the extension just described it is $\lambda\|\mathbf{w}'\|$.

Table 11.1
Comparison of the running-time complexity of KRR for computing the solution or the prediction value of a point in both the primal and the dual case. κ denotes the time complexity of computing a kernel value; for polynomial and Gaussian kernels, $\kappa = O(N)$.

	Solution	Prediction
Primal	$O(mN^2 + N^3)$	$O(N)$
Dual	$O(\kappa m^2 + m^3)$	$O(\kappa m)$

In both the primal and dual cases, KRR admits a closed-form solution. Table 11.1 gives the time complexity of the algorithm for computing the solution and the one for determining the prediction value of a point in both cases. In the primal case, determining the solution $\mathbf{w}$ requires computing matrix $\mathbf{X}\mathbf{X}^\top$, which takes $O(mN^2)$, the inversion of $(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})$, which is in $O(N^3)$, and multiplication with $\mathbf{X}$, which is in $O(mN^2)$. Prediction requires computing the inner product of $\mathbf{w}$ with a feature vector of the same dimension that can be achieved in $O(N)$. The dual solution first requires computing the kernel matrix $\mathbf{K}$. Let κ be the maximum cost of computing $K(x, x')$ for all pairs $(x, x') \in \mathcal{X} \times \mathcal{X}$. Then, $\mathbf{K}$ can be computed in $O(\kappa m^2)$. The inversion of matrix $\mathbf{K} + \lambda\mathbf{I}$ can be achieved in $O(m^3)$ and multiplication with $\mathbf{Y}$ takes $O(m^2)$. Prediction requires computing the vector $(K(x_1, x), \ldots, K(x_m, x))^\top$ for some $x \in \mathcal{X}$, which requires $O(\kappa m)$, and the inner product with $\boldsymbol{\alpha}$, which is in $O(m)$.

Thus, in both cases, the main step for computing the solution is a matrix inversion, which takes $O(N^3)$ in the primal case, $O(m^3)$ in the dual case. When the dimension of the feature space is relatively small, solving the primal problem is advantageous, while for high-dimensional spaces and medium-sized training sets, solving the dual is preferable. Note that for relatively large matrices, the space complexity could also be an issue: the size of relatively large matrices could be prohibitive for memory storage and the use of external memory could significantly affect the running time of the algorithm.

For sparse matrices, there exist several techniques for faster computations of the matrix inversion. This can be useful in the primal case where the features can be relatively sparse. On the other hand, the kernel matrix $\mathbf{K}$ is typically dense; thus, there is less hope for benefiting from such techniques in the dual case. In such cases, or, more generally, to deal with the time and space complexity issues arising when m and N are large, approximation methods using low-rank approximations via the Nyström method or the partial Cholesky decomposition can be used very effectively.

The KRR algorithm admits several advantages: it benefits from favorable theoretical guarantees since it can be derived directly from the generalization bound we

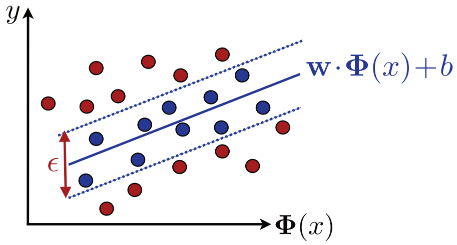

Figure 11.4
SVR attempts to fit a "tube" with width ϵ to the data. Training data within the "epsilon tube" (blue points) incur no loss.

presented; it admits a closed-form solution, which can make the analysis of many of its properties convenient; and it can be used with PDS kernels, which extends its use to non-linear regression solutions and more general features spaces. KRR also admits favorable stability properties that we discuss in chapter 14.

The algorithm can be generalized to learning a mapping from $\mathcal{X}$ to $\mathbb{R}^p$, $p > 1$. This can be done by formulating the problem as p independent regression problems, each consisting of predicting one of the p target components. Remarkably, the computation of the solution for this generalized algorithm requires only a single matrix inversion, e.g., $(\mathbf{K} + \lambda\mathbf{I})^{-1}$ in the dual case, regardless of the value of p.

One drawback of the KRR algorithm, in addition to the computational issues for determining the solution for relatively large matrices, is the fact that the solution it returns is typically not sparse. The next two sections present two sparse algorithms for linear regression.

11.3.3 Support vector regression

In this section, we present the *support vector regression* (SVR) algorithm, which is inspired by the SVM algorithm presented for classification in chapter 5. The main idea of the algorithm consists of fitting a tube of width $\epsilon > 0$ to the data, as illustrated by figure 11.4. As in binary classification, this defines two sets of points: those falling inside the tube, which are ϵ-close to the function predicted and thus not penalized, and those falling outside, which are penalized based on their distance to the predicted function, in a way that is similar to the penalization used by SVMs in classification.

Using a hypothesis set $\mathcal{H}$ of linear functions: $\mathcal{H} = \{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x) + b \colon \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}$, where $\mathbf{\Phi}$ is the feature mapping corresponding some PDS kernel K, the optimization problem for SVR can be written as follows:

$$\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \big|y_i - (\mathbf{w}\cdot\mathbf{\Phi}(x_i) + b)\big|_{\epsilon}, \tag{11.20}$$

where $|\cdot|_\epsilon$ denotes the *ϵ-insensitive loss*:

$$\forall y, y' \in \mathcal{Y}, \quad |y' - y|_\epsilon = \max(0, |y' - y| - \epsilon). \tag{11.21}$$

The use of this loss function leads to sparse solutions with a relatively small number of support vectors. Using slack variables $\xi_i \geq 0$ and $\xi_i' \geq 0$, $i \in [m]$, the optimization problem can be equivalently written as

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}'} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i') \\ \text{subject to } & (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b) - y_i \leq \epsilon + \xi_i \\ & y_i - (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b) \leq \epsilon + \xi_i' \\ & \xi_i \geq 0, \xi_i' \geq 0, \ \forall i \in [m]. \end{aligned} \tag{11.22}$$

This is a convex quadratic program (QP) with affine constraints. Introducing the Lagrangian and applying the KKT conditions leads to the following equivalent dual problem in terms of the kernel matrix $\mathbf{K}$:

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \boldsymbol{\alpha}'} \ & -\epsilon(\boldsymbol{\alpha}' + \boldsymbol{\alpha})^\top \mathbf{1} + (\boldsymbol{\alpha}' - \boldsymbol{\alpha})^\top \mathbf{y} - \frac{1}{2}(\boldsymbol{\alpha}' - \boldsymbol{\alpha})^\top \mathbf{K} (\boldsymbol{\alpha}' - \boldsymbol{\alpha}) \\ \text{subject to: } & (\mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C}) \wedge (\mathbf{0} \leq \boldsymbol{\alpha}' \leq \mathbf{C}) \wedge ((\boldsymbol{\alpha}' - \boldsymbol{\alpha})^\top \mathbf{1} = 0) \,. \end{aligned} \tag{11.23}$$

Any PDS kernel K can be used with SVR, which extends the algorithm to non-linear regression solutions. Problem (11.23) is a convex QP similar to the dual problem of SVMs and can be solved using similar optimization techniques. The solutions $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ define the hypothesis h returned by SVR as follows:

$$\forall x \in \mathcal{X}, \quad h(x) = \sum_{i=1}^{m} (\alpha_i' - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b \,, \tag{11.24}$$

where the offset b can be obtained from a point x_j with $0 < \alpha_j < C$ by

$$b = -\sum_{i=1}^{m} (\alpha_i' - \alpha_i) K(x_i, x_j) + y_j + \epsilon, \tag{11.25}$$

or from a point x_j with $0 < \alpha_j' < C$ via

$$b = -\sum_{i=1}^{m} (\alpha_i' - \alpha_i) K(x_i, x_j) + y_j - \epsilon. \tag{11.26}$$

By the complementarity conditions, for all $i \in [m]$, the following equalities hold:

$$\begin{aligned} & \alpha_i \big((\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b) - y_i - \epsilon - \xi_i \big) = 0 \\ & \alpha_i' \big((\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b) - y_i + \epsilon + \xi_i' \big) = 0. \end{aligned}$$

Thus, if $\alpha_i \neq 0$ or $\alpha'_i \neq 0$, that is if x_i is a support vector, then, either $(\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b) - y_i - \epsilon = \xi_i$ holds or $y_i - (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b) - \epsilon = \xi'_i$. This shows that support vectors points lying outside the ϵ-tube. Of course, at most one of α_i or α'_i is non-zero for any point x_i: the hypothesis either overestimates or underestimates the true label by more than ϵ. For the points within the ϵ-tube, we have $\alpha_j = \alpha'_j = 0$; thus, these points do not contribute to the definition of the hypothesis returned by SVR. Thus, when the number of points inside the tube is relatively large, the hypothesis returned by SVR is relatively sparse. The choice of the parameter ϵ determines a trade-off between sparsity and accuracy: larger ϵ values provide sparser solutions, since more points can fall within the ϵ-tube, but may ignore too many key points for determining an accurate solution.

The following generalization bounds hold for the ϵ-insensitive loss and kernel-based hypotheses and thus for the SVR algorithm. We denote by $\mathcal{D}$ the distribution according to which sample points are drawn and by $\widehat{\mathcal{D}}$ the empirical distribution defined by a training sample of size m.

Theorem 11.13 *Let $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PDS kernel, let $\Phi\colon \mathcal{X} \to \mathbb{H}$ be a feature mapping associated to K and let $\mathcal{H} = \{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x)\colon \|\mathbf{w}\|_{\mathbb{H}} \leq \Lambda\}$. Assume that there exists $r > 0$ such that $K(x, x) \leq r^2$ and $M > 0$ such that $|h(x) - y| \leq M$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Fix $\epsilon > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, each of the following inequalities holds for all $h \in \mathcal{H}$,*

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\big[|h(x) - y|_\epsilon\big] \leq \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}\big[|h(x) - y|_\epsilon\big] + 2\sqrt{\frac{r^2\Lambda^2}{m}} + M\sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\big[|h(x) - y|_\epsilon\big] \leq \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}\big[|h(x) - y|_\epsilon\big] + \frac{2\Lambda\sqrt{\mathrm{Tr}[\mathbf{K}]}}{m} + 3M\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Proof: Since for any $y' \in \mathcal{Y}$, the function $y \mapsto |y - y'|_\epsilon$ is 1-Lipschitz, the result follows Theorem 11.3 and the bound on the empirical Rademacher complexity of $\mathcal{H}$. □

These results provide theoretical guarantees for the SVR algorithm. Notice, however, that the theorem does not provide guarantees for the expected loss of the hypotheses in terms of the squared loss. For $0 < \epsilon < 1/4$, the inequality $|x|^2 \leq |x|_\epsilon$ holds for all x in $[-\eta'_\epsilon, -\eta_\epsilon] \cup [\eta_\epsilon, \eta'_\epsilon]$ with $\eta_\epsilon = \frac{1-\sqrt{1-4\epsilon}}{2}$ and $\eta'_\epsilon = \frac{1+\sqrt{1-4\epsilon}}{2}$. For small values of ϵ, $\eta_\epsilon \approx 0$ and $\eta'_\epsilon \approx 1$, thus, if $M = 2r\lambda \leq 1$, then, the squared loss can be upper bounded by the ϵ-insensitive loss for almost all values of $(h(x) - y)$ in $[-1, 1]$ and the theorem can be used to derive a useful generalization bound for the squared loss.

More generally, if the objective is to achieve a small squared loss, then, SVR can be modified by using the *quadratic ϵ-insensitive loss*, that is the square of the ϵ-insensitive loss, which also leads to a convex QP. We will refer by *quadratic SVR* to this version of the algorithm. Introducing the Lagrangian and applying the KKT conditions leads to the following equivalent dual optimization problem for quadratic SVR in terms of the kernel matrix $\mathbf{K}$:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} \; -\epsilon(\boldsymbol{\alpha}'+\boldsymbol{\alpha})^\top \mathbf{1} + (\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top \mathbf{y} - \frac{1}{2}(\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top \Big(\mathbf{K}+\frac{1}{C}\mathbf{I}\Big)(\boldsymbol{\alpha}'-\boldsymbol{\alpha}) \tag{11.27}$$

$$\text{subject to: } (\boldsymbol{\alpha} \geq \mathbf{0}) \wedge (\boldsymbol{\alpha}' \geq \mathbf{0}) \wedge (\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top \mathbf{1} = 0)\,.$$

Any PDS kernel K can be used with quadratic SVR, which extends the algorithm to non-linear regression solutions. Problem (11.27) is a convex QP similar to the dual problem of SVMs in the separable case and can be solved using similar optimization techniques. The solutions $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ define the hypothesis h returned by SVR as follows:

$$h(x) = \sum_{i=1}^{m} (\alpha_i' - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b\,, \tag{11.28}$$

where the offset b can be obtained from a point x_j with $0 < \alpha_j < C$ or $0 < \alpha_j' < C$ exactly as in the case of SVR with (non-quadratic) ϵ-insensitive loss. Note that for $\epsilon = 0$, the quadratic SVR algorithm coincides with KRR as can be seen from the dual optimization problem (the additional constraint $(\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top \mathbf{1} = 0$ appears here due to use of an offset b). The following generalization bound holds for quadratic SVR. It can be shown in a way that is similar to the proof of theorem 11.13 using the fact that the quadratic ϵ-insensitive function $x \mapsto |x|_\epsilon^2$ is $2M$-Lipschitz over the interval $[-M, +M]$.

Theorem 11.14 *Let $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PDS kernel, let $\Phi\colon \mathcal{X} \to \mathbb{H}$ be a feature mapping associated to K and let $\mathcal{H} = \{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x)\colon \|\mathbf{w}\|_{\mathbb{H}} \leq \Lambda\}$. Assume that there exists $r > 0$ such that $K(x,x) \leq r^2$ and $M > 0$ such that $|h(x) - y| \leq M$ for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$. Fix $\epsilon > 0$. Then, for any $\delta > 0$, with probability at least $1-\delta$, each of the following inequalities holds for all $h \in \mathcal{H}$,*

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\big[|h(x)-y|_\epsilon^2\big] \leq \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}\big[|h(x)-y|_\epsilon^2\big] + 4M\sqrt{\frac{r^2\Lambda^2}{m}} + M^2\sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\big[|h(x)-y|_\epsilon^2\big] \leq \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}}\big[|h(x)-y|_\epsilon^2\big] + \frac{4M\Lambda\sqrt{\operatorname{Tr}[\mathbf{K}]}}{m} + 3M^2\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

This theorem provides a strong justification for the quadratic SVR algorithm. Alternative convex loss functions can be used to define regression algorithms, in particular the *Huber loss* (see figure 11.5), which penalizes smaller errors quadratically and larger ones only linearly.

SVR admits several advantages: the algorithm is based on solid theoretical guarantees, the solution returned is sparse, and it allows a natural use of PDS kernels,

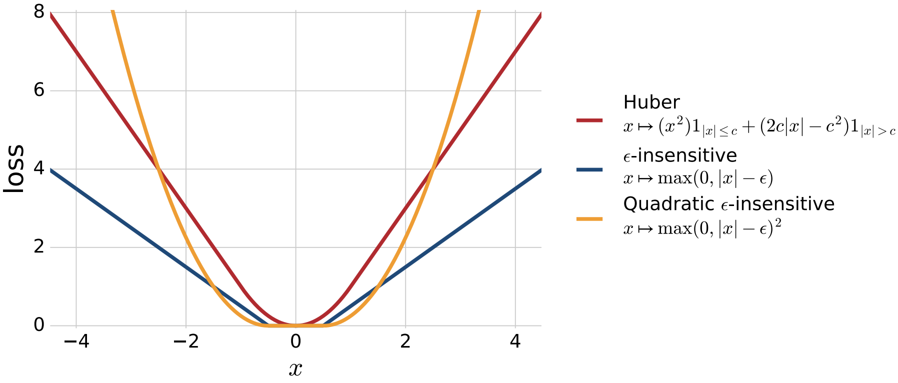

Figure 11.5
Alternative loss functions that can be used in conjunction with SVR.

which extend the algorithm to non-linear regression solutions. SVR also admits favorable stability properties that we discuss in chapter 14. However, one drawback of the algorithm is that it requires the selection of two parameters, C and ϵ. These can be selected via cross-validation, as in the case of SVMs, but this requires a relatively larger validation set. Some heuristics are often used to guide the search for their values: C is searched near the maximum value of the labels in the absence of an offset ($b = 0$) and for a normalized kernel, and ϵ is chosen close to the average difference of the labels. As already discussed, the value of ϵ determines the number of support vectors and the sparsity of the solution. Another drawback of SVR is that, as in the case of SVMs or KRR, it may be computationally expensive when dealing with large training sets. One effective solution in such cases, as for KRR, consists of approximating the kernel matrix using low-rank approximations via the Nyström method or the partial Cholesky decomposition. In the next section, we discuss an alternative sparse algorithm for regression.

11.3.4 Lasso

Unlike the KRR and SVR algorithms, the Lasso (least absolute shrinkage and selection operator) algorithm does not admit a natural use of PDS kernels. Thus, here, we assume that the input space $\mathcal{X}$ is a subset of $\mathbb{R}^N$ and consider a family of linear hypotheses $\mathcal{H} = \{x \mapsto \mathbf{w} \cdot \mathbf{x} + b \colon \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}$.

Let $S = \big((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\big) \in (\mathcal{X} \times \mathcal{Y})^m$ be a labeled training sample. Lasso is based on the minimization of the empirical squared error on S with a regularization term depending on the norm of the weight vector, as in the case of the ridge regression, but using the L_1 norm instead of the L_2 norm and without squaring the

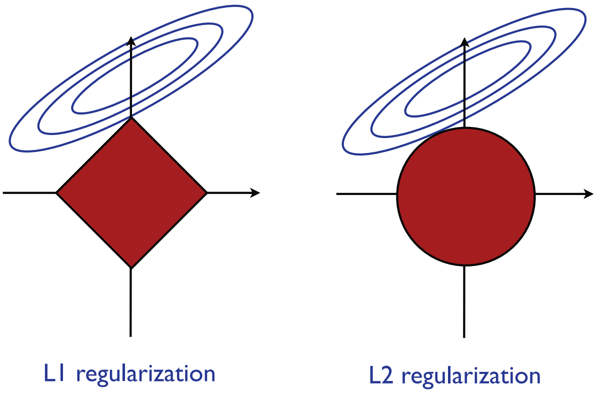

Figure 11.6
Comparison of the Lasso and ridge regression solutions.

norm:

$$\min_{\mathbf{w},b} F(\mathbf{w},b) = \lambda\|\mathbf{w}\|_1 + \sum_{i=1}^{m} (\mathbf{w}\cdot\mathbf{x}_i + b - y_i)^2 \ . \tag{11.29}$$

Here λ denotes a positive parameter as for ridge regression. This is a convex optimization problem, since $\|\cdot\|_1$ is convex as with all norms and since the empirical error term is convex, as already discussed for linear regression. The optimization for Lasso can be written equivalently as

$$\min_{\mathbf{w},b} \sum_{i=1}^{m} (\mathbf{w}\cdot\mathbf{x}_i + b - y_i)^2 \quad \text{subject to: } \|\mathbf{w}\|_1 \leq \Lambda_1, \tag{11.30}$$

where Λ_1 is a positive parameter.

The key property of Lasso as in the case of other algorithms using the L_1 norm constraint is that it leads to a sparse solution $\mathbf{w}$, that is one with few non-zero components. Figure 11.6 illustrates the difference between the L_1 and L_2 regularizations in dimension two. The objective function of (11.30) is a quadratic function, thus its contours are ellipsoids, as illustrated by the figure (in blue). The areas corresponding to L_1 and L_2 balls of a fixed radius Λ_1 are also shown in the left and right panel (in red). The Lasso solution is the point of intersection of the contours with the L_1 ball. As can be seen form the figure, this can typically occur at a corner of the L_1 ball where some coordinates are zero. In contrast, the ridge regression solution is at the point of intersection of the contours and the L_2 ball, where none of the coordinates is typically zero.

The following results show that Lasso also benefits from strong theoretical guarantees. We first give a general upper bound on the empirical Rademacher complexity of L_1 norm-constrained linear hypotheses .

Theorem 11.15 (Rademacher complexity of linear hypotheses with bounded L_1 norm) *Let $\mathcal{X} \subseteq \mathbb{R}^N$ and let $S = \big((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\big) \in (\mathcal{X} \times \mathcal{Y})^m$ be a sample of size m. Assume that for all $i \in [m]$, $\|\mathbf{x}_i\|_\infty \leq r_\infty$ for some $r_\infty > 0$, and let $\mathcal{H} = \{\mathbf{x} \in \mathcal{X} \mapsto \mathbf{w} \cdot \mathbf{x} \colon \|\mathbf{w}\|_1 \leq \Lambda_1\}$. Then, the empirical Rademacher complexity of $\mathcal{H}$ can be bounded as follows:*

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) \leq \sqrt{\frac{2r_\infty^2 \Lambda_1^2 \log(2N)}{m}}. \tag{11.31}$$

Proof: For any $i \in [m]$ we denote by x_{ij} the jth component of $\mathbf{x}_i$.

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(\mathcal{H}) &= \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\|\mathbf{w}\|_1 \leq \Lambda_1} \sum_{i=1}^m \sigma_i \mathbf{w} \cdot \mathbf{x}_i \right] \\
&= \frac{\Lambda_1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\Big\| \sum_{i=1}^m \sigma_i \mathbf{x}_i \Big\|_\infty \right] && \text{(by definition of the dual norm)} \\
&= \frac{\Lambda_1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\max_{j \in [N]} \left| \sum_{i=1}^m \sigma_i x_{ij} \right| \right] && \text{(by definition of } \|\cdot\|_\infty\text{)} \\
&= \frac{\Lambda_1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\max_{j \in [N]} \max_{s \in \{-1,+1\}} s \sum_{i=1}^m \sigma_i x_{ij} \right] && \text{(by definition of } \|\cdot\|_\infty\text{)} \\
&= \frac{\Lambda_1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{z} \in A} \sum_{i=1}^m \sigma_i z_i \right],
\end{aligned}$$

where A denotes the set of N vectors $\{s(x_{1j}, \ldots, x_{mj})^\top : j \in [N], s \in \{-1, +1\}\}$. For any $\mathbf{z} \in A$, we have $\|\mathbf{z}\|_2 \leq \sqrt{mr_\infty^2} = r_\infty\sqrt{m}$. Thus, by Massart's lemma (theorem 3.7), since A contains at most $2N$ elements, the following inequality holds:

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) \leq \Lambda_1 r_\infty \sqrt{m} \frac{\sqrt{2\log(2N)}}{m} = r_\infty \Lambda_1 \sqrt{\frac{2\log(2N)}{m}},$$

which concludes the proof. □

Note that dependence of the bound on the dimension N is only logarithmic, which suggests that using very high-dimensional feature spaces does not significantly affect generalization.

Combining the Rademacher complexity bound just proven and the general result of Theorem 11.3 yields the following generalization bound for the hypothesis set used by Lasso, using the squared loss.

Theorem 11.16 *Let $\mathcal{X} \subseteq \mathbb{R}^N$ and $\mathcal{H} = \{\mathbf{x} \in \mathcal{X} \mapsto \mathbf{w} \cdot \mathbf{x} \colon \|\mathbf{w}\|_1 \leq \Lambda_1\}$. Assume that there exists $r_\infty > 0$ such for all $\mathbf{x} \in \mathcal{X}$, $\|\mathbf{x}\|_\infty \leq r_\infty$ and $M > 0$ such that*

$|h(x) - y| \leq M$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, each of the following inequalities holds for all $h \in \mathcal{H}$:

$$R(h) \leq \widehat{R}_S(h) + 2r_\infty \Lambda_1 M \sqrt{\frac{2\log(2N)}{m}} + M^2 \sqrt{\frac{\log \frac{1}{\delta}}{2m}}. \tag{11.32}$$

As in the case of ridge regression, we observe that the objective function minimized by Lasso has the same form as the right-hand side of this generalization bound.

There exist a variety of different methods for solving the optimization problem of Lasso, including an efficient algorithm (LARS) for computing the entire *regularization path* of solutions, that is, the Lasso solutions for all values of the regularization parameter λ, and other on-line solutions that apply more generally to optimization problems with an L_1 norm constraint.

Here, we show that the Lasso problems (11.29) or (11.30) are equivalent to a quadratic program (QP), and therefore that any QP solver can be used to compute the solution. Observe that any weight vector $\mathbf{w}$ can be written as $\mathbf{w} = \mathbf{w}^+ - \mathbf{w}^-$, with $\mathbf{w}^+ \geq 0$, $\mathbf{w}^- \geq 0$, and $w_j^+ = 0$ or $w_j^- = 0$ for any $j \in [N]$, which implies $\|\mathbf{w}\|_1 = \sum_{j=1}^N w_j^+ + w_j^-$. This can be done by defining the jth component of $\mathbf{w}^+$ as w_j if $w_j \geq 0$, 0 otherwise, and similarly the jth component of $\mathbf{w}^-$ as $-w_j$ if $w_j \leq 0$, 0 otherwise, for any $j \in [N]$. With the replacement $\mathbf{w} = \mathbf{w}^+ - \mathbf{w}^-$, with $\mathbf{w}^+ \geq 0$, $\mathbf{w}^- \geq 0$, and $\|\mathbf{w}\|_1 = \sum_{j=1}^N w_j^+ + w_j^-$, the Lasso problem (11.29) becomes

$$\min_{\mathbf{w}^+ \geq 0, \mathbf{w}^- \geq 0, b} \lambda \sum_{j=1}^N (w_j^+ + w_j^-) + \sum_{i=1}^m \big((\mathbf{w}^+ - \mathbf{w}^-) \cdot \mathbf{x}_i + b - y_i\big)^2 . \tag{11.33}$$

Conversely, a solution $\mathbf{w} = \mathbf{w}^+ - \mathbf{w}^-$ of (11.33) verifies the condition $w_j^+ = 0$ or $w_j^- = 0$ for any $j \in [N]$, thus $w_j = w_j^+$ when $w_j \geq 0$ and $w_j = -w_j^-$ when $w_j \leq 0$. This is because if $\delta_j = \min(w_j^+, w_j^-) > 0$ for some $j \in [N]$, replacing w_j^+ with $(w_j^+ - \delta_j)$ and w_j^- with $(w_j^- - \delta_j)$ would not affect $w_j^+ - w_j^- = (w_j^+ - \delta) - (w_j^- - \delta)$, but would reduce the term $(w_j^+ + w_j^-)$ in the objective function by $2\delta_j > 0$ and provide a better solution. In view of this analysis, problems (11.29) and (11.33) admit the same optimal solution and are equivalent. Problem (11.33) is a QP since the objective function is quadratic in $\mathbf{w}^+$, $\mathbf{w}^-$, and b, and since the constraints are affine. With this formulation, the problem can be straightforwardly shown to admit a natural online algorithmic solution (exercise 11.10).[20]

Thus, Lasso has several advantages: it benefits from strong theoretical guarantees and returns a sparse solution, which is advantageous when there are accurate solutions based on few features. The sparsity of the solution is also computationally

[20] The technique we described to avoid absolute values in the objective function can be used similarly in other optimization problems.

attractive; sparse feature representations of the weight vector can be used to make the inner product with a new vector more efficient. The algorithm's sparsity can also be used for feature selection. The main drawback of the algorithm is that it does not admit a natural use of PDS kernels and thus an extension to non-linear regression, unlike KRR and SVR. One solution is then to use empirical kernel maps, as discussed in chapter 6. Also, Lasso's solution does not admit a closed-form solution. This is not a critical property from the optimization point of view but one that can make some mathematical analyses very convenient.

11.3.5 Group norm regression algorithms

Other types of regularization aside from the L_1 or L_2 norm can be used to define regression algorithms. For instance, in some situations, the feature space may be naturally partitioned into subsets, and it may be desirable to find a sparse solution that selects or omits entire subsets of features. A natural norm in this setting is the group or mixed norm $L_{2,1}$, which is a combination of the L_1 and L_2 norms. Imagine that we partition $\mathbf{w} \in \mathbb{R}^N$ as $\mathbf{w}_1, \ldots, \mathbf{w}_k$, where $\mathbf{w}_j \in \mathbb{R}^{N_j}$ for $1 \leq j \leq k$ and $\sum_j N_j = N$, and define $\mathbf{W} = (\mathbf{w}_1^\top, \ldots, \mathbf{w}_k^\top)^\top$. Then the $L_{2,1}$ norm of $\mathbf{W}$ is defined as

$$\|\mathbf{W}\|_{2,1} = \sum_{j=1}^{k} \|\mathbf{w}_j\| .$$

Combining the $L_{2,1}$ norm with the empirical mean squared error leads to the *Group Lasso* formulation. More generally, an $L_{q,p}$ group norm regularization can be used for $q, p \geq 1$ (see appendix A for the definition of group norms).

11.3.6 On-line regression algorithms

The regression algorithms presented in the previous sections admit natural online versions. Here, we briefly present two examples of these algorithms. These algorithms are particularly useful for applications to very large data sets for which a batch solution can be computationally too costly to derive and more generally in all of the on-line learning settings discussed in chapter 8.

Our first example is known as the *Widrow-Hoff algorithm* and coincides with the application of stochastic gradient descent techniques to the linear regression objective function. Figure 11.7 gives the pseudocode of the algorithm. A similar algorithm can be derived by applying the stochastic gradient technique to ridge regression. At each round, the weight vector is augmented with a quantity that depends on the prediction error $(\mathbf{w}_t \cdot \mathbf{x}_t - y_t)$.

Our second example is an online version of the SVR algorithm, which is obtained by application of stochastic gradient descent to the dual objective function of SVR. Figure 11.8 gives the pseudocode of the algorithm for an arbitrary PDS kernel K

```
WidrowHoff($\mathbf{w}_0$)
  $\mathbf{w}_1 \leftarrow \mathbf{w}_0$        ▷ typically $\mathbf{w}_0 = \mathbf{0}$
  for $t \leftarrow 1$ to $T$ do
      Receive($\mathbf{x}_t$)
      $\widehat{y}_t \leftarrow \mathbf{w}_t \cdot \mathbf{x}_t$
      Receive($y_t$)
      $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + 2\eta(\mathbf{w}_t \cdot \mathbf{x}_t - y_t)\mathbf{x}_t$    ▷ learning rate $\eta > 0$.
  return $\mathbf{w}_{T+1}$
```

Figure 11.7
The Widrow-Hoff algorithm.

in the absence of any offset ($b = 0$). Another on-line regression algorithm is given by exercise 11.10 for Lasso.

11.4 Chapter notes

The generalization bounds presented in this chapter are for bounded regression problems. When $\{x \mapsto L(h(x), y)\colon h \in \mathcal{H}\}$, the family of losses of the hypotheses, is not bounded, a single function can take arbitrarily large values with arbitrarily small probabilities. This is the main issue for deriving uniform convergence bounds for unbounded losses. This problem can be avoided either by assuming the existence of an *envelope*, that is a single non-negative function with a finite expectation lying above the absolute value of the loss of every function in the hypothesis set [Dudley, 1984, Pollard, 1984, Dudley, 1987, Pollard, 1989, Haussler, 1992], or by assuming that some moment of the loss functions is bounded [Vapnik, 1998, 2006]. Cortes, Greenberg, and Mohri [2013] (see also [Cortes et al., 2010a]) give two-sided generalization bounds for unbounded losses with finite second moments. The one-sided version of their bounds coincides with that of Vapnik [1998, 2006] modulo a constant factor, but the proofs given by Vapnik in both books seem to be incomplete and incorrect.

The notion of pseudo-dimension is due to Pollard [1984]. Its equivalent definition in terms of VC-dimension is discussed by Vapnik [2000]. The notion of fat-shattering was introduced by Kearns and Schapire [1990]. The linear regression algorithm is a classical algorithm in statistics that dates back at least to the nine-

```
OnLineDualSVR()
$\boldsymbol{\alpha} \leftarrow \mathbf{0}$
$\boldsymbol{\alpha}' \leftarrow \mathbf{0}$
for $t \leftarrow 1$ to $T$ do
    Receive($x_t$)
    $\widehat{y}_t \leftarrow \sum_{s=1}^{t} (\alpha'_s - \alpha_s) K(x_s, x_t)$
    Receive($y_t$)
    $\alpha'_{t+1} \leftarrow \alpha'_t + \min(\max(\eta(y_t - \widehat{y}_t - \epsilon), -\alpha'_t), C - \alpha'_t)$
    $\alpha_{t+1} \leftarrow \alpha_t + \min(\max(\eta(\widehat{y}_t - y_t - \epsilon), -\alpha_t), C - \alpha_t)$
return $\sum_{t=1}^{T} (\alpha'_t - \alpha_t) K(x_t, \cdot)$
```

Figure 11.8
An on-line version of dual SVR.

teenth century. The ridge regression algorithm is due to Hoerl and Kennard [1970]. Its kernelized version (KRR) was introduced and discussed by Saunders, Gammerman, and Vovk [1998]. An extension of KRR to outputs in $\mathbb{R}^p$ with $p > 1$ with possible constraints on the regression is presented and analyzed by Cortes, Mohri, and Weston [2007c]. The support vector regression (SVR) algorithm is discussed in Vapnik [2000]. Lasso was introduced by Tibshirani [1996]. The LARS algorithm for solving its optimization problem was later presented by Efron et al. [2004]. The Widrow-Hoff on-line algorithm is due to Widrow and Hoff [1988]. The dual on-line SVR algorithm was first introduced and analyzed by Vijayakumar and Wu [1999]. The kernel stability analysis of exercise 10.3 is from Cortes et al. [2010b].

For large-scale problems where a straightforward batch optimization of a primal or dual objective function is intractable, general iterative stochastic gradient descent methods similar to those presented in section 11.3.6, or quasi-Newton methods such as the limited-memory BFGS (Broyden-Fletcher-Goldfard-Shanno) algorithm [Nocedal, 1980] can be practical alternatives in practice.

In addition to the linear regression algorithms presented in this chapter and their kernel-based non-linear extensions, there exist many other algorithms for regression, including decision trees for regression (see chapter 9), boosting trees for regression, and artificial neural networks.

11.5 Exercises

11.1 Pseudo-dimension and monotonic functions.

Assume that ϕ is a strictly monotonic function and let $\phi \circ \mathcal{H}$ be the family of functions defined by $\phi \circ \mathcal{H} = \{\phi(h(\cdot)) : h \in \mathcal{H}\}$, where $\mathcal{H}$ is some set of real-valued functions. Show that $\text{Pdim}(\phi \circ \mathcal{H}) = \text{Pdim}(\mathcal{H})$.

11.2 Pseudo-dimension of linear functions. Let $\mathcal{H}$ be the set of all linear functions in dimension d, i.e. $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ for some $\mathbf{w} \in \mathbb{R}^d$. Show that $\text{Pdim}(\mathcal{H}) = d$.

11.3 Linear regression.

(a) What condition is required on the data $\mathbf{X}$ in order to guarantee that $\mathbf{X}\mathbf{X}^\top$ is invertible?

(b) Assume the problem is under-determined. Then, we can choose a solution $\mathbf{w}$ such that the equality $\mathbf{X}^\top \mathbf{w} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^\dagger \mathbf{X}\mathbf{y}$ (which can be shown to equal $\mathbf{X}^\dagger \mathbf{X}\mathbf{y}$) holds. One particular choice that satisfies this equality is $\mathbf{w}^* = (\mathbf{X}\mathbf{X}^\top)^\dagger \mathbf{X}\mathbf{y}$. However, this is not the unique solution. As a function of $\mathbf{w}^*$, characterize all choices of $\mathbf{w}$ that satisfy $\mathbf{X}^\top \mathbf{w} = \mathbf{X}^\dagger \mathbf{X}\mathbf{y}$ (*Hint*: use the fact that $\mathbf{X}\mathbf{X}^\dagger \mathbf{X} = \mathbf{X}$).

11.4 Perturbed kernels. Suppose two different kernel matrices, $\mathbf{K}$ and $\mathbf{K}'$, are used to train two kernel ridge regression hypothesis with the same regularization parameter λ. In this problem, we will show that the difference in the optimal dual variables, $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ respectively, is bounded by a quantity that depends on $\|\mathbf{K}' - \mathbf{K}\|_2$.

(a) Show $\boldsymbol{\alpha}' - \boldsymbol{\alpha} = \big((\mathbf{K}' + \lambda \mathbf{I})^{-1}(\mathbf{K}' - \mathbf{K})(\mathbf{K} + \lambda \mathbf{I})^{-1}\big)\mathbf{y}$. (*Hint*: Show that for any invertible matrix $\mathbf{M}$, $\mathbf{M}'^{1} - \mathbf{M}^{1} = -\mathbf{M}'^{-1}(\mathbf{M}' - \mathbf{M})\mathbf{M}^{-1}$.)

(b) Assuming $\forall y \in \mathcal{Y}, |y| \leq M$, show that

$$\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\| \leq \frac{\sqrt{m} M \|\mathbf{K}' - \mathbf{K}\|_2}{\lambda^2}.$$

11.5 Huber loss. Derive the primal and dual optimization problem used to solve the SVR problem with the Huber loss:

$$L_c(\xi_i) = \begin{cases} \frac{1}{2}\xi_i^2, & \text{if } |\xi_i| \leq c \\ c\xi_i - \frac{1}{2}c^2, & \text{otherwise} \end{cases},$$

where $\xi_i = \mathbf{w} \cdot \Phi(\mathbf{x}_i) + b - y_i$.

```
OnLineLasso($\mathbf{w}_0^+, \mathbf{w}_0^-$)
$\mathbf{w}_1^+ \leftarrow \mathbf{w}_0^+$        $\triangleright\ \mathbf{w}_0^+ \geq 0$
$\mathbf{w}_1^- \leftarrow \mathbf{w}_0^-$        $\triangleright\ \mathbf{w}_0^- \geq 0$
for $t \leftarrow 1$ to $T$ do
    Receive($\mathbf{x}_t, y_t$)
    for $j \leftarrow 1$ to $N$ do
        $w_{t+1j}^+ \leftarrow \max\Big(0, w_{tj}^+ - \eta\Big[\lambda - \big[y_t - (\mathbf{w}_t^+ - \mathbf{w}_t^-) \cdot \mathbf{x}_t\big]\mathbf{x}_{tj}\Big]\Big)$
        $w_{t+1j}^- \leftarrow \max\Big(0, w_{tj}^- - \eta\Big[\lambda + \big[y_t - (\mathbf{w}_t^+ - \mathbf{w}_t^-) \cdot \mathbf{x}_t\big]\mathbf{x}_{tj}\Big]\Big)$
return $\mathbf{w}_{T+1}^+ - \mathbf{w}_{T+1}^-$
```

Figure 11.9
On-line algorithm for Lasso.

11.6 SVR and squared loss. Assuming that $2r\Lambda \leq 1$, use theorem 11.13 to derive a generalization bound for the squared loss.

11.7 SVR dual formulations. Give a detailed and carefully justified derivation of the dual formulations of the SVR algorithm both for the ϵ-insensitive loss and the quadratic ϵ-insensitive loss.

11.8 Optimal kernel matrix. Suppose in addition to optimizing the dual variables $\alpha \in \mathbb{R}^m$, as in (11.16), we also wish to optimize over the entries of the PDS kernel matrix $\mathbf{K} \in \mathbb{R}^{m \times m}$.

$$\min_{\mathbf{K} \succeq 0} \max_{\boldsymbol{\alpha}} -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y}\,, \quad \text{s.t.} \quad \|\mathbf{K}\|_2 \leq 1$$

(a) What is the closed-form solution for the optimal $\mathbf{K}$ for the joint optimization?

(b) Optimizing over the choice of kernel matrix will provide a better value of the objective function. Explain, however, why the resulting kernel matrix is not useful in practice.

11.9 Leave-one-out error. In general, the computation of the leave-one-out error can be very costly since, for a sample of size m, it requires training the algorithm m times. The objective of this problem is to show that, remarkably, in the case of kernel ridge regression, the leave-one-out error can be computed efficiently by training the algorithm only once.

Let $S = ((x_1, y_1), \ldots, (x_m, y_m))$ denote a training sample of size m and for any $i \in [m]$, let S_i denote the sample of size $m - 1$ obtained from S by removing (x_i, y_i): $S_i = S - \{(x_i, y_i)\}$. For any sample T, let h_T denote a hypothesis obtained by training T. By definition (see definition 5.2), for the squared loss, the leave-one-out error with respect to S is defined by

$$\widehat{R}_{\text{LOO}}(\text{KRR}) = \frac{1}{m} \sum_{i=1}^{m} (h_{S_i}(x_i) - y_i)^2 .$$

(a) Let $S'_i = ((x_1, y_1), \ldots, (x_i, h_{S_i}(y_i)), \ldots, (x_m, y_m))$. Show that $h_{S_i} = h_{S'_i}$.

(b) Define $\mathbf{y}_i = \mathbf{y} - y_i \boldsymbol{e}_i + h_{S_i}(x_i)\boldsymbol{e}_i$, that is the vector of labels with the ith component replaced with $h_{S_i}(x_i)$. Prove that for KRR $h_{S_i}(x_i) = \mathbf{y}_i^\top (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K} \boldsymbol{e}_i$.

(c) Prove that the leave-one-out error admits the following simple expression in terms of h_S:

$$\widehat{R}_{\text{LOO}}(\text{KRR}) = \frac{1}{m} \sum_{i=1}^{m} \left[\frac{h_S(x_i) - y_i}{\boldsymbol{e}_i^\top (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K} \boldsymbol{e}_i} \right]^2 . \tag{11.34}$$

(d) Suppose that the diagonal entries of matrix $\mathbf{M} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K}$ are all equal to γ. How do the empirical error $\widehat{R}_S$ of the algorithm and the leave-one-out error $\widehat{R}_{\text{LOO}}$ relate? Is there any value of γ for which the two errors coincide?

11.10 On-line Lasso. Use the formulation (11.33) of the optimization problem of Lasso and stochastic gradient descent (see section 8.3.1) to show that the problem can be solved using the on-line algorithm of figure 11.9.

11.11 On-line quadratic SVR. Derive an on-line algorithm for the quadratic SVR algorithm (provide the full pseudocode).