Christian Igel

Machine Learning: Kernel-based Methods

Lecture Notes

Version 0.2.7

Department of Computer Science
University of Copenhagen

Preface

"When I grow old I will be a script!"

This text is supplementary material for the courses "Introduction to the Mathematics of Supervised Learning" and "Machine Learning: Kernel-based Methods". However, please note that it does not fully cover the lecture. In particular, the units 2 and 3 are missing almost completely and many proofs are left out. On the other hand, there is some advanced material that is not discussed in the lecture. The main and definite resources for preparing for the "Introduction to the Mathematics of Supervised Learning" exam are the handouts.

My goal is to revise and extend this document over time. Please do not hesitate to give me feedback (christian.igel@ini.rub.de), in particular, please point out errors (even small ones) and sections or paragraphs that are not clear or should be extended.

This script builds on existing textbooks, especially those by Schölkopf and Smola [2002], Shawe-Taylor and Cristianini [2004], and Lange [2004], which are highly recommended.

Contents

1

Introduction

We strive for computer systems that can deal autonomously and flexibly with our needs. Such systems must work in situations that have not been fully specified a priori. Incomplete descriptions of application scenarios are inevitable because we need software for domains where the designer's knowledge is not perfect, the solutions to particular problems are simply unknown, and/or the sheer complexity and variability of the task and the environment precludes a sufficiently accurate description of the domain. Although such systems are in general too complex to be designed manually, large amounts of data describing the task and the environment are often available or can be autonomously obtained. To take proper advantage of this available information, we need to develop systems that self-adapt and automatically improve based on sample data—systems that learn.

Accordingly, machine learning, which can be defined as the scientific field devoted to answering the fundamental question of how we can "build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes" [Mitchell, 2006], has become an important research area in computer science.

In order to build powerful learning machines, an interdisciplinary approach is needed. Machine learning is clearly rooted in computer science as well as in statistics and optimization theory, because learning from experience means analyzing data and improving with respect to some measure of performance can be cast as an optimization problem. Still, I regard understanding the principles of how humans learn as the royal road to creating flexible, autonomous machines. This goes without saying that machine learning need—and should—not work exactly like the learning in biological systems. Different constraints require and allow for different implementations of the same fundamental principles.

Machine learning algorithms are already an integral part of today's computing systems. Highly specialized technical solutions for restricted task domains exist that have reached superhuman performance, usually for problems that require extensive computation or handling large datasets. State-of-the-art

real-world applications of systems relying at least partly on machine learning include pattern recognition in biometrics, text and speech recognition, internet search engines, and data mining in bioinformatics.

2

Supervised Learning

This chapter summarizes basic concepts of supervised learning and pattern recognition.

2.1 The Learning Problem

Let us consider the standard supervised learning scenario, in which we want to infer a relation between elements of an input space $\mathcal{X}$ and an output space $\mathcal{Y}$. The learning is driven by sample data $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ with input patterns $x_i \in \mathcal{X}$ and labels $y_i \in \mathcal{Y}$ for $1 \leq i \leq \ell$. In accordance with the standard literature, S is referred to as training set, although S is better defined as a multiset and in many scenarios even as an ordered sequence. A training set is called *non-trivial* if it contains patterns with different labels.

In the following, the same symbol p is used for both probability density functions as well as probability mass functions. These specify continuous and discrete distributions of random variables, respectively, and therefore a distribution is often directly identified by p and with an abuse of notation the symbol p is also used for the underlying probability measures. If not stated otherwise, it is assumed that all data are identically, independently distributed (i.i.d.), that is, each pattern is an independent realization of a joint random variable $Z = (X, Y)$ with a stationary distribution p over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. We further assume that p can be factorized according to $p(x, y) = p_X(x) \cdot p_{Y|X}(y|x)$, where $p_X(x)$ and $p_{Y|X}(y|x)$ are distributions over $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Supervised learning is often divided into *classification*, where the goal is to assign an input to one of a finite set of classes $\mathcal{Y} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$, $2 \leq m < \infty$; *regression*, where we want to predict an output in $\mathcal{Y} \subseteq \mathbb{R}^m$, $1 \leq m < \infty$, given an input pattern; and *density estimation*, where the task is to predict the probability distribution $p(y|x)$ on $\mathcal{Y}$ given an input x [Vapnik, 1998].

In the following, the focus is on classification, the most basic machine learning task, but most considerations apply to regression problems as well. Only the inference of static relations is considered, although arguably the most

interesting learning systems are situated in time. However, solving the static problem is a prerequisite for solving the more complex time dependent one, especially because one can cast many time dependent problems into static ones.

The goal of a learning machine is to infer an appropriate *hypothesis* from data. A hypothesis $h : \mathcal{X} \to \mathcal{Y}$ is a map returning an output given an input. A *hypothesis class* $\mathcal{H}$ is a set of those functions. Thus, given $\mathcal{H}$ and sample data S, a *learning algorithm* can be defined as a function a returning a hypothesis in $\mathcal{H}$:

$$a : \{((x_1, y_1), \ldots, (x_\ell, y_\ell)) \,|\, 1 \leq i \leq \ell < \infty; x_i \in \mathcal{X}, y_i \in \mathcal{Y}\} \to \mathcal{H}$$

This definition is not fully appropriate for *online* and *active* learning.

The quality of the prediction of a hypothesis is quantified based on a task-dependent *loss function*. A loss function can generally be defined as a map $L : \mathcal{Y} \times \mathcal{Y} \to [0, \infty[$ with the property $L(y, y) = 0$, where $L(y, \hat{y})$ represents the price we pay by predicting $\hat{y}$ in place of y.

Now we can formally state the goal of supervised machine learning, namely to come up with a hypothesis h showing good average performance in terms of L on patterns $(x_i, y_i) \sim p$ based on sample data. That is, we want to find h minimizing the *risk*

$$\mathcal{R}_p(h) = \int L(y, h(x)) \mathrm{d}p(x, y) \ .$$

In general, we cannot compute the risk, especially because we usually do not know p. What can be determined is the *empirical risk* $\mathcal{R}_S(h)$ of h on sample data S ($|S| = \ell$) by computing

$$\mathcal{R}_S(h) = \frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, h(x_i)) \ .$$

A typical loss function for classification is the *0-1 loss* $L(y, \hat{y}) = \mathbf{1}\{y \neq \hat{y}\}$, where the indicator function $\mathbf{1}\{\cdot\}$ is 1 if its argument is true and 0 otherwise. When considering the 0-1 loss, the risk of a hypothesis h is the probability of error, $\mathcal{R}_p(h) = \mathbb{E}_p\{\mathbf{1}\{h(X) \neq Y\}\}$, and the empirical risk on sample data S ($|S| = \ell$) basically counts mistakes, $\mathcal{R}_S(h) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbf{1}\{h(x_i) \neq y_i\}$. For regression, the squared error $L(y, \hat{y}) = (y - \hat{y})^2$ is a common choice. Then the empirical risk corresponds to the mean-squared error. The squared error can be motivated as follows. Given a hypothesis h it is assumed that given an input $x \in \mathcal{X}$ the observed label $\mathbf{y} \in \mathbb{R}^m$ is distributed normally around h. That is, we have $p(\mathbf{y} \,|\, x; h) \sim \mathcal{N}(h(x), \sigma^2 \mathbf{I})$ with covariance matrix $\sigma^2 \mathbf{I}$, where $\mathbf{I}$ denotes the m-dimensional unit matrix and σ^2 a scalar variance parameter. Under these assumptions, minimizing the mean squared error corresponds to choosing the hypothesis h that maximizes the *likelihood function*

$$\mathcal{L}(h, S) = \prod_{i=1}^{\ell} p(\mathbf{y}_i \,|\, x_i; h)$$

w.r.t. $S = \{(x_1, \mathbf{y}_1), \ldots, (x_\ell, \mathbf{y}_\ell)\}$. The likelihood function $\mathcal{L}(h, S)$ is the probability to observe the patterns in S under the model h. The learning strategy to maximize $\mathcal{L}$ is known as *maximum likelihood* inference.

The best risk value we can achieve is the *Bayes risk* $\mathcal{R}_p^{\text{Bayes}}$, which is defined as the minimum risk over all measurable functions

$$\mathcal{R}_p^{\text{Bayes}} = \mathcal{R}_p(h^{\text{Bayes}}) = \inf_h \mathcal{R}_p(h)$$

and usually differs from what can be achieved by a hypothesis in $\mathcal{H}$ (i.e., from $\inf_{h \in \mathcal{H}} \mathcal{R}_p(h)$). For $\mathcal{Y} = \{-1, 1\}$ we have

$$h^{\text{Bayes}}(x) = \operatorname{sgn}\left[\mathbb{E}_{p_{Y|X}}(y|x)\right] \quad .$$

Here we define $\operatorname{sgn}(x) = 2\mathbf{1}\{x \geq 0\} - 1$. If $p_{Y|X}(y|x)$ is not a Dirac delta function for all possible input patterns $x \in \mathcal{X}$ then the Bayes risk is not zero. A hypothesis attaining the Bayes risk is called *Bayes optimal*. An algorithm a is *consistent* if $\lim_{\ell \to \infty} \mathcal{R}_S(a(S)) = \mathcal{R}_p^{\text{Bayes}}$ almost surely. This is a highly desired property, because it tells us that we can expect the learning algorithm to converge to an optimal hypothesis with increasing sample size.

2.2 Generalization, Complexity, and Regularization

The goal of learning is not just to memorize the training data by heart, which corresponds to minimizing $\mathcal{R}_S(h)$, but to find a hypothesis that fits the underlying distribution and therefore generalizes well. That is, the algorithm should output a hypothesis making accurate predictions given inputs drawn according to p_X, regardless whether these input patterns were provided in the training data or not.

We talk of *overfitting* when a hypothesis does not generalize well because it faithfully reflects aspects of the sample data to the extent that idiosyncrasies of these data, rather than merely of the underlying distribution, shape the hypothesis. An example is shown in Fig. 2.1. Overfitting can occur when the hypothesis class $\mathcal{H}$ is too rich or too complex and the learning algorithm a determines the hypothesis using empirical risk minimization given $\mathcal{H}$ ("learning by heart restricted to $\mathcal{H}$"):

$$a(S) = h_S = \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{R}_S(h)$$

Now, what does complexity or richness of a hypothesis class mean? There are different ways to quantify the complexity of a hypothesis class, for example in terms of the *VC dimension* (Vapnik-Chervonenkis dimension, Vapnik and

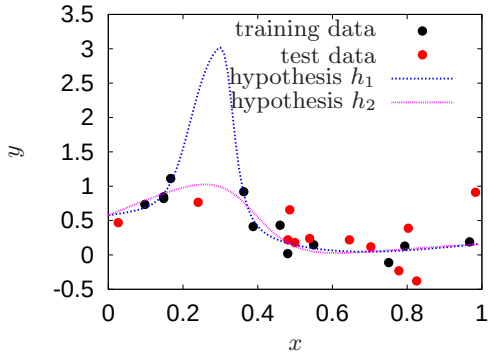


Fig. 2.1. The dots show i.i.d. sample data, the curves h_1 and h_2 correspond to hypotheses generated by a neural network [e.g., see Bishop, 1995, 2006] with and without early-stopping, respectively. Without early-stopping overfitting can be observed.

Chervonenkis, 1971) or the *Rademacher complexity*, which is discussed in more detail below. These measures of richness or capacity of a hypothesis class $\mathcal{H}$ have an intuitive interpretation in terms of the ability to find a function in $\mathcal{H}$ fitting arbitrarily labeled data. But more important, they allow to bound the largest difference between true and empirical loss for functions in $\mathcal{H}$ with a high probability. A typical bound can be expressed like this: With probability of at least $1-\delta \in]0,1[$ it holds $\mathcal{R}_p(h_S) \leq \mathcal{R}_S(h_S) + B(\ell, \mathcal{H}, \delta)$. The value of B increases with the complexity of $\mathcal{H}$ and decreases with the number of training samples ℓ and the uncertainty δ.

Given some notion of complexity of a hypothesis class, we can apply a learning strategy termed *structural risk minimization*. Given a nested sequence of hypothesis spaces $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \mathcal{H}_3 \ldots$ with non-decreasing complexity, the learning algorithms selects a hypothesis according to:

$$h_S = \operatorname*{argmin}_{h \in \mathcal{H}_d, d \in \mathbb{N}} \left[\mathcal{R}_S(h) + \text{penalty}(\ell, \mathcal{H}_d) \right]$$

This strategy falls in the general *regularization* framework, where given $\mathcal{H}$ one tries to prevent overfitting by selecting a hypothesis according to

$$h_S = \operatorname*{argmin}_{h \in \mathcal{H}} \left[\mathcal{R}_S(h) + \text{penalty}(\ell, h) \right] \quad .$$

A very important example of regularization, applicable when $\mathcal{H}$ is a normed space, is

$$h_S = \operatorname*{argmin}_{h \in \mathcal{H}} \left[\mathcal{R}_S(h) + \vartheta \|h\|_{\mathcal{H}} \right]$$

with regularization parameter $\vartheta > 0$ controlling the trade-off between empirical risk minimization and keeping the norm of the hypothesis small. It

is interesting to note that this is related to Bayesian approaches to machine learning, where a likelihood $\propto \exp(-\vartheta \mathcal{R}_S)$ and an a priori probability of a hypothesis described by $p_{\mathcal{H}}$ yields

$$h_S = \operatorname*{argmin}_{h \in \mathcal{H}} \left[\vartheta \mathcal{R}_S(h) + \log p_{\mathcal{H}}(h) \right]$$

as maximum a posteriori (MAP) estimate of the hypothesis.

There are other ways of avoiding overfitting that do not fall in the general regularization framework outlined above. An important example is *early-stopping* [Bishop, 2006], see Fig. 2.1. The learning algorithm partitions the sample S into training S_{train} and validation S_{val} data. Then it iteratively produces a sequence of hypotheses $h_1, h_2, h_3, \ldots$ based on S_{train}, ideally corresponding to a nested sequence of hypothesis spaces $\mathcal{H}_1 \subseteq \mathcal{H}_2 \ldots$ with non-decreasing complexity and $h_i \in \mathcal{H}_i$ and decreasing empirical risk $\mathcal{R}_{S_{\text{train}}}(h_i) > \mathcal{R}_{S_{\text{train}}}(h_{i+1})$ measured on the training data. The empirical risk $\mathcal{R}_{S_{\text{val}}}(h_i)$ measured on the validation data is also monitored, and the algorithm stops if this value, usually smoothed over time, increases considerably. The algorithm finally outputs the hypothesis h_i minimizing $\mathcal{R}_{S_{\text{val}}}(h_i)$.

2.2.1 Rademacher Complexity

In this section we consider the Rademacher complexity as an example for a measure of the richness or capacity of a hypothesis class [Bartlett and Mendelson, 2002, Shawe-Taylor and Cristianini, 2004, Ambroladze et al., 2007].

For a sample $S = \{z_1, \ldots, z_\ell\}$ generated by a distribution p over a set $\mathcal{Z}$ and a real-valued function class $\mathcal{H}$ with domain $\mathcal{X}$, the *empirical Rademacher complexity* of $\mathcal{H}$ is the random variable

$$\hat{R}_\ell(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}} \left\{ \sup_{f \in \mathcal{H}} \left| \frac{2}{\ell} \sum_{i=1}^{\ell} \sigma_i f(x_i) \right| \,\middle|\, x_1, \ldots, x_\ell \right\} \ ,$$

where $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_\ell\}^T$ are independent uniform $\{\pm 1\}$-valued (Rademacher) random variables and $\mathbb{E}_{\boldsymbol{\sigma}}$ denotes the expectation w.r.t. $\boldsymbol{\sigma}$. The *Rademacher complexity* of $\mathcal{H}$ is

$$R_\ell(\mathcal{H}) = \mathbb{E}_S\{\hat{R}_\ell(\mathcal{H})\} = \mathbb{E}_{S\boldsymbol{\sigma}} \left\{ \sup_{f \in \mathcal{H}} \left| \frac{2}{\ell} \sum_{i=1}^{\ell} \sigma_i f(x_i) \right| \right\} \ .$$

Here, $\mathbb{E}_S$ denotes the expectation w.r.t. S drawn according to p^ℓ. A proof of this theorem is given by Ambroladze et al. [2007] (see also Shawe-Taylor and Cristianini, 2004).

The Rademacher complexity measures how well noise (i.e., realization of the Rademacher random variables, which can be interpreted as random labels of the training patterns) can be fitted by functions from the class $\mathcal{H}$, where the term $\frac{2}{\ell} \sum_{i=1}^{\ell} \sigma_i f(x_i)$ measures the quality of the fit similar to a correlation.

The following theorem lists properties of the Rademacher complexity (see Ambroladze et al., 2007 and Shawe-Taylor and Cristianini, 2004 for proofs).

Theorem 2.1. *For classes of real functions $\mathcal{F}, \mathcal{F}_1, \ldots, \mathcal{F}_n$ and $\mathcal{G}$:*

1. *If $\mathcal{F} \subseteq \mathcal{G}$, then $\hat{R}_\ell(\mathcal{F}) \leq \hat{R}_\ell(\mathcal{G})$;*
2. *$\hat{R}_\ell(\mathcal{F}) = \hat{R}_\ell(\operatorname{conv} \mathcal{F})$, where $\operatorname{conv}(F)$ denotes the set of convex combinations of elements of a vector space F;*
3. *for every $c \in \mathbb{R}$, $\hat{R}_\ell(c\mathcal{F}) = |c|\hat{R}_\ell(\mathcal{F})$*
4. *If $\mathcal{A} : \mathbb{R} \to \mathbb{R}$, is Lipschitz with constant L and satisfies $\mathcal{A}(0) = 0$, then $\hat{R}_\ell(\mathcal{A} \circ \mathcal{F}) \leq 2L\hat{R}_\ell(\mathcal{F})$;*
5. *For any function h, $\hat{R}_\ell(\mathcal{F} + h) \leq \hat{R}_\ell(\mathcal{F}) + 2\sqrt{\frac{\hat{\mathbb{E}}\{h^2\}}{\ell}}$;*
6. *For any $1 \leq q < \infty$, let $\mathcal{L}_{\mathcal{F},h,q} = \{|f - h|^q | f \in \mathcal{F}\}$. If $\|f - h\|_\infty \leq 1$ for every $f \in \mathcal{F}$, then $\hat{R}_\ell(\mathcal{L}_{\mathcal{F},h,q}\} \leq 2q\left(\hat{R}_\ell(\mathcal{F}) + 2\sqrt{\frac{\mathbb{E}\{h^2\}}{\ell}}\right)$;*
7. *$\hat{R}_\ell\big(\bigcup_{i=1}^n \mathcal{F}_i\big) \leq \sum_{i=1}^n \hat{R}_\ell(\mathcal{F}_i)$.*

Here $\hat{\mathbb{E}}$ denoted the empirical expectation (i.e., the average computed over the sample data).

The Rademacher complexity can be bounded using the empirical Rademacher complexity.

Lemma 2.1. *For classes of real functions $\mathcal{F}$ mapping to $[0,1]$ and a sample $S = \{x_1, \ldots, x_\ell\}$ we have with probability of at least $1 - \delta$*

$$R_\ell(\mathcal{F}) \leq \hat{R}_\ell(\mathcal{F}) + 2\sqrt{\frac{-\ln \delta}{2\ell}} \quad .$$

Proof. This bound follows from applying McDiarmid's (see Theorem A.1) inequality with $c_i = 2/\ell, i = 1, \ldots, \ell$.

If we apply the Rademacher complexity not just to a class of hypotheses $\mathcal{H}$ but to a loss function applied to these hypotheses, we can bound the largest difference between true and empirical loss for functions in $\mathcal{H}$. This can be done using the following fundamental result on the Rademacher complexity:

Theorem 2.2. *Fix $\delta \in]0,1[$ and let $\mathcal{F}$ be a class of functions mapping from $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ to $[0,1]$. Let $S = (z_1, \ldots, z_\ell)$ be drawn independently according to a probability distribution p. Then with probability at least $1 - \delta$ over random draws of samples of size ℓ, every $f \in \mathcal{F}$ satisfies*

$$\begin{aligned} \mathbb{E}_p f &\leq \hat{\mathbb{E}}_S f + R_\ell(\mathcal{F}) + \sqrt{\frac{\ln(2/\delta)}{2\ell}} \\ &\leq \hat{\mathbb{E}}_S f + \hat{R}_\ell(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}} \end{aligned}$$

with $\mathbb{E}_p f = \int f(z) dp(z)$ and $\hat{\mathbb{E}}_S f = \frac{1}{\ell} \sum_{z' \in S} f(z')$.

Proof. We follow the proof by Shawe-Taylor and Cristianini [2004]. For a fixed $f \in \mathcal{F}$ we have

$$\mathbb{E}_p f \leq \hat{\mathbb{E}}_S f + \sup_{h \in \mathcal{F}} \left(\mathbb{E}_p h - \hat{\mathbb{E}}_S h \right) \ .$$

McDiarmid's inequality with $c_i = 1/\ell$ gives with probability of at least $1 - \delta/2$

$$\mathbb{E}_p f \leq \hat{\mathbb{E}}_S f + \mathbb{E}_S \sup_{h \in \mathcal{F}} \left(\mathbb{E}_p h - \hat{\mathbb{E}}_S h \right) + \sqrt{\frac{\ln(2/\delta)}{2\ell}} \ .$$

Now we introduce a second sample, a *ghost sample*, $\tilde{S}$ and get

$$\begin{aligned}
\mathbb{E}_S \sup_{h \in \mathcal{F}} \left(\mathbb{E}_p h - \hat{\mathbb{E}}_S h \right) &= \mathbb{E}_S \sup_{h \in \mathcal{F}} \mathbb{E}_{\tilde{S}} \left[\frac{1}{\ell} \sum_{i=1}^{\ell} h(\tilde{x}_i) - \frac{1}{\ell} \sum_{i=1}^{\ell} h(x_i) \,\middle|\, S \right] \\
&\leq \mathbb{E}_S \mathbb{E}_{\tilde{S}} \left[\sup_{h \in \mathcal{F}} \frac{1}{\ell} \sum_{i=1}^{\ell} (h(\tilde{x}_i) - h(x_i)) \right] \\
&= \mathbb{E}_{\boldsymbol{\sigma} S \tilde{S}} \left[\sup_{h \in \mathcal{F}} \frac{1}{\ell} \sum_{i=1}^{\ell} \sigma_i (h(\tilde{x}_i) - h(x_i)) \right] \\
&\leq 2 \mathbb{E}_{\boldsymbol{\sigma} S} \left[\sup_{h \in \mathcal{F}} \left| \sum_{i=1}^{\ell} \frac{1}{\ell} \sigma_i h(\tilde{x}_i) \right| \right] = R_\ell(\mathcal{F}) \ .
\end{aligned}$$

From the first to the second line we used the fact that $\sup \mathbb{E} f \leq \mathbb{E} \sup f$ for any function f and for the last inequality the general rule $a - b \leq |a| + |b|$.

This result can be used to derive bounds for the generalization error. To this end, we consider a loss function L mapping to $[0, 1]$ (e.g., the 0-1 loss), define the *loss class* $\mathcal{F} = \{f : (x, y) \mapsto L(y, h(x)) \,|\, h \in \mathcal{H}\}$, and bound $\hat{R}_\ell(\mathcal{F})$ in terms of $\hat{R}_\ell(\mathcal{H})$ [Ambroladze et al., 2007]. Then $\mathbb{E}_p f$ and $\hat{\mathbb{E}}_S f$ in Theorem 2.2 correspond to the true and empirical risk, respectively. This procedure can be used to prove Theorem 4.4.

The link to the VC dimension $\mathrm{VCdim}(\mathcal{F})$ of the class is established by $\hat{R}_\ell(\mathcal{F}) \in O\left(\sqrt{\frac{\mathrm{VCdim}(\mathcal{F})}{\ell}} \right)$ [Bousquet et al., 2004].

2.2.2 No-free-lunch for Learning

It is not only intuitive, but also provable that it is not possible to design an universal learning machine that outperforms other systems across all possible problems. This is formally expressed by the no-free-lunch (NFL) theorems [Wolpert, 1996, Devroye and Gyorfi, 1997, Bousquet et al., 2004]. Coarsely speaking, the NFL theorems for learning state that if there is no assumption how the training data ("the past") is related to test data ("the future"), prediction is impossible. In other words, if there is no a priori restriction on the

possible phenomena that are expected, it is impossible to achieve generalization and thus no algorithm is superior to another. Even worse, any consistent algorithm (i.e., any algorithm converging to the Bayes optimal classifier almost surely when the number of training patterns, drawn independently from the distribution describing the problem, approaches infinity) can have arbitrarily poor behavior when given a finite, incomplete training set.

Thus, generalization is only possible if we have additional knowledge about the underlying distribution, as pointed out by Bousquet et al. [2004]: "Generalization = Data + Knowledge".

2.3 Bibliographical Remarks

For further reading, the excellent introduction to learning theory by Bousquet et al. [2004] is highly recommended. Good chapters on statistical learning theory can be found in the textbooks by Anthony and Bartlett [1999], Schölkopf and Smola [2002], and Shawe-Taylor and Cristianini [2004]. For further information on applied statistical learning the textbooks by Hastie et al. [2001] and Bishop [2006] are recommended.

3

Optimization

Machine learning is concerned with changing a system to the better. Thus, there is a close link between machine learning and optimization. In this chapter a few selected topics from gradient-based optimization are considered, which will later be used in the context of supervised learning.

3.1 Basic Definitions

Let $f : M \to \mathbb{R}$ with $M \subseteq \mathbb{R}^n$ be an *objective function* to be optimized. Because maximizing a function $f(\mathbf{x})$ for $\mathbf{x} \in M$ corresponds to minimizing $-f(\mathbf{x})$ for $\mathbf{x} \in M$, we only consider minimization (e.g., of error or risk) in the following without loss of generalization. Points $\mathbf{x}^* \in M$ minimizing f are called (global) minima and the optimal value $f(\mathbf{x}^*)$ of the objective function is called the value of the optimization problem.

In the following, gradient-based optimization is considered. The objective function f is assumed to differentiable. The gradient of f at point $\mathbf{x}$ is denoted by $\boldsymbol{\nabla} f(\mathbf{x}) = (\partial f(\mathbf{x})/\partial x_1, \ldots, \partial f(\mathbf{x})/\partial x_n)^{\mathrm{T}}$.

3.2 Constraint Optimization

Now we add constraints restricting the space of feasible solutions. For functions f, g_i, h_j ($i = 1, \ldots, k$ and $j = 1, \ldots, m$) defined on an open set $M \subset \mathbb{R}^n$, the *primal optimization problem* in standard form is defined as follows:

Definition 3.1 (Primal Optimization Problem). *Given functions f, g_i, h_j ($i = 1, \ldots, k$ and $j = 1, \ldots, m$) defined on an open set $M \subset \mathbb{R}^n$, the* primal optimization problem *is given by*

$$\begin{array}{lll} \textit{minimize} & f(\mathbf{w}) & \mathbf{w} \in M \\ \textit{subject to} & g_i(\mathbf{w}) \leq 0 & i = 1, \ldots, k \\ & h_i(\mathbf{w}) = 0 & i = 1, \ldots, m \end{array} ,$$

where $f(\mathbf{w})$ is called the objective function, $g_i(\mathbf{w})$ the inequality constraints, and $h_i(\mathbf{w})$ the equality constraints. The optimal value $f(\mathbf{x^*})$ of the objective function is called the value of the optimization problem. The feasible region is defined by all $\mathbf{w} \in M$ for which $\forall i = 1, \ldots, k : g_i(\mathbf{w}) \leq 0 \wedge \forall j = 1, \ldots, m : h_j(\mathbf{w})$.

If for an inequality constraint $g_i(\mathbf{w}) = 0$, then the constraint g_i is called active, if $g_i(\mathbf{w}) < 0$, then g_i is called inactive.

Because maximizing a function $f(\mathbf{w})$ for $\mathbf{w} \in M$ corresponds to minimizing $-f(\mathbf{w})$ for $\mathbf{w} \in M$, we just consider minimization in the following. inequality constraint!active inequality constraints!inactive

In the following, we first look at necessary conditions for a minimum of the above optimization problem. That is, we derive properties that hold for any optimal point (but may also be true for other, non-optimal points). Then, sufficient conditions for a point being a minimum are derived (not every condition which is sufficient for a point being optimal has to be fulfilled for every optimal point).

3.2.1 Necessary Conditions for a Minimum

Theorem 3.1 (Karush-Kuhn-Tucker Lagrange multiplier rule). *Suppose the objective function $f(\mathbf{x})$ of the constraint optimization problem has a local minimum at the feasible point $\mathbf{x}^*$. If $f(\mathbf{x})$ and the various constraint functions are continuously differentiable near $\mathbf{x}^*$, then there exists a unit vector of Lagrange multipliers $\lambda_0, \ldots, \lambda_m, \mu_1, \ldots, \mu_k$ such that*

$$\lambda_0 \nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{j=1}^{k} \mu_j \nabla g_j(\mathbf{x}^*) = \mathbf{0} \ .$$

Each of the multipliers $\lambda_0, \mu_1, \ldots, \mu_k$ is nonnegative and $\mu_j = 0$ if $g_j(\mathbf{x}^*) < 0$.

Proof. Without loss of generality, we assume $\mathbf{x} = \mathbf{0}$, $f(\mathbf{0}) = 0$ and that the first r inequality constraints are active and that the other $k - r$ are inactive at $\mathbf{x}$. For the inactive constraints we set $\mu_j = 0$, $k \geq j > r$, and thus most of the following sums run over the active constraints only.

For our proof, we consider subset of M around $\mathbf{x}^*$ with convenient properties. We pick $\delta > 0$ such that

1. the closed ball
$$C(\mathbf{0}, \delta) = \{\|\mathbf{u}\| \leq \delta \,|\, \mathbf{u} \in \mathbb{R}^n\}$$
is contained in M,
2. objective function and constraints are continuously differentiable in $C(\mathbf{0}, \delta)$,
3. inactive constraints in $\mathbf{0}$ are inactive throughout $C(\mathbf{0}, \delta)$.

Such a δ exists as M is an open set and the objective function and the constraints are continuously differentiable near the optimum by assumption. From the continuity of the inequality constraints ensures a δ that fulfills the third property.

The trick is to turn the inequality constraints into penalty functions by defining

$$\tilde{g}_j(\mathbf{u}) = \max\{g_j(\mathbf{u}), 0\} \ .$$

The proof can be divided into three major steps. First, we prove that for each $0 < \epsilon \leq \delta$ there is a $c > 0$ such that

$$F(\mathbf{u}) = f(\mathbf{u}) + \|\mathbf{u}\|^2 + c\sum_{i=1}^{m} h_i(\mathbf{u})^2 + c\sum_{j=1}^{r} \tilde{g}_j(\mathbf{u})^2 > 0$$

for all $\mathbf{u}$ with $\|\mathbf{u}\| = \epsilon$ (note that $f(\mathbf{u})$ need not be nonnegative in $C(\mathbf{0}, \delta)$ outside the feasible region). If $\mathbf{u}$ is feasible, $h_i(\mathbf{u})^2 = \tilde{g}_j(\mathbf{u})^2 = 0$ for all i and j and the above inequality implies $f(\mathbf{u}) > -\epsilon^2$.

Assume the claim is false. That is, there exists an $\epsilon \leq \delta$ such that there is no $c > 0$ such that

$$F(\mathbf{u}) = f(\mathbf{u}) + \|\mathbf{u}\|^2 + c\sum_{i=1}^{m} h_i(\mathbf{u})^2 + c\sum_{j=1}^{r} \tilde{g}_j(\mathbf{u})^2 > 0$$

for all $\mathbf{u}$ with $\|\mathbf{u}\| = \epsilon$. Then there exists a sequence of points $\mathbf{u}_1, \mathbf{u}_2, \ldots$ with $\|\mathbf{u}_n\| = \epsilon$ and a corresponding sequence of numbers c_n tending to ∞ such that

$$f(\mathbf{u}_n) + \|\mathbf{u}_n\|^2 \leq -c_n \sum_{i=1}^{m} h_j(\mathbf{u}_n)^2 - c_n \sum_{j=1}^{r} \tilde{g}_j(\mathbf{u}_n)^2 \ ,$$

which implies $f(\mathbf{u}_n) \leq -\epsilon^2$ if $\mathbf{u}_n$ is feasible. Because $C(\mathbf{0}, \delta)$ is compact, the sequence $\mathbf{u}_n$ has a convergent subsequence with limit $\mathbf{z}$, w. l. o. g. say the sequence itself. We have $\|\mathbf{z}\| = \epsilon$. Taking the limit

$$\lim_{n\to\infty} \left(\frac{f(\mathbf{u}_n) + \|\mathbf{u}\|^2}{-c_n} \geq \sum_{i=1}^{m} h_i(\mathbf{u}_n)^2 + \sum_{j=1}^{r} \tilde{g}_j(\mathbf{u}_n)^2 \right)$$

implies

$$\sum_{i=1}^{m} h_i(\mathbf{z})^2 + \sum_{j=1}^{r} \tilde{g}_j(\mathbf{z})^2 = 0 \ .$$

This implies that $\mathbf{z}$ is feasible and thus $f(\mathbf{z}) \geq f(\mathbf{0}) = 0$. This contradicts the assumption.

The next step is to prove an intermediate multiplier rule, namely that there exists a point $\mathbf{u}$ and a vector $(\lambda_0, \lambda_1, \ldots, \lambda_m, \mu_1, \ldots, \mu_r)^{\mathrm{T}}$ such that

1. $\|\mathbf{u}\| < \epsilon$,

2. $\lambda_0, \mu_1, \ldots, \mu_r \geq 0$,
3. $\|(\lambda_0, \mu_1, \ldots, \mu_r)^{\mathrm{T}}\| = 1$,
4. $\lambda_0[\nabla f(\mathbf{u}) + 2\mathbf{u}] + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{u}) + \sum_{i=1}^{r} \mu_i \nabla g_i(\mathbf{u}) = \mathbf{0}$.

We consider $F(\mathbf{u})$ with c picked as described above. There is a point $\mathbf{u}$ giving the unconstrained minimum of $F(\mathbf{u})$ on the compact set $C(\mathbf{0}, \epsilon)$. Because $F(\mathbf{u}) \leq F(\mathbf{0}) = 0$, it is impossible that $\|\mathbf{u}\| = \epsilon$ as shown above. Thus $\mathbf{u}$ falls in the interior of $C(\mathbf{0}, \epsilon)$ and we have $\nabla F(\mathbf{u}) = \mathbf{0}$, which means that

$$\nabla f(\mathbf{u}) + 2\mathbf{u} + c \sum_{i=1}^{m} 2h_i(\mathbf{u}) \nabla h_i(\mathbf{u}) + c \sum_{i=1}^{r} 2g_i(\mathbf{u}) \nabla \tilde{g}_i(\mathbf{u}) = \mathbf{0} \ .$$

Normalization and renaming of the multipliers finishes our second step.

Finally, we choose a sequence $\epsilon_n > 0$ tending to 0 and corresponding points $\mathbf{u}_n$ where the intermediate multiplier rule holds. The corresponding sequence of multiplier unit vectors has a convergent subsequence with limit

$$(\lambda_0, \lambda_1, \ldots, \lambda_m, \mu_1, \ldots, \mu_r)^{\mathrm{T}}$$

that is also a unit vector. Now we consider the sequence $\mathbf{u}_{i_n}$, the subsequence of $\mathbf{u}_n$ corresponding to the convergent subsequence of multiplier unit vectors. This sequence converges to 0 because $\|\mathbf{u}_{i_n}\| \leq \epsilon_{n_l}$. Now we can take limits along $\mathbf{u}_{i_n}$ in the intermediate multiplier rule completing the proof.

Lemma 3.1 (Constraint qualification & Kuhn-Tucker condition). *By definition, the* Mangasarin-Fromowitz constraint qualification *holds at a feasible point* $\mathbf{x}$ *if the differentials* $dh_i(\mathbf{x})$ *are linearly independent and there exists a vector* $\mathbf{v}$ *such that* $dh_i(\mathbf{x})\mathbf{v} = 0$ *and* $dg_i(\mathbf{x})\mathbf{v} < 0$ *for all inequality constraints* $g_i(\mathbf{x})$ *active at* $\mathbf{x}$.

The Kuhn-Tucker condition *holds provided that the differentials* $dh_i(\mathbf{x})$ *of the equality constraints and the differentials* $dg_i(\mathbf{x})$ *of the active inequality constraints are linearly independent at* $\mathbf{x}$.

The Kuhn-Tucker condition implies Mangasarin-Fromowitz constraint qualification.

Proof. Given the Kuhn-Tucker condition holds. For r active inequality constraints we define

$$\mathbf{A} = (\nabla h_1(\mathbf{x}), \ldots, \nabla h_m(\mathbf{x}), \nabla g_1(\mathbf{x}), \ldots, \nabla g_r(\mathbf{x}), \mathbf{b}_1, \ldots, \mathbf{b}_{n-m-r}) \in \mathbb{R}^{n \times n}$$

with $b_1, \ldots, b_{n-m-r}$ chosen such that $\mathbf{A}$ has full rank. Then we can solve

$$\mathbf{A}^{\mathrm{T}}\mathbf{v} = -(\underbrace{0, \ldots, 0}_{m \text{ times}}, \underbrace{1, \ldots, 1}_{r \text{ times}}, \underbrace{0, \ldots, 0}_{n-m-r \text{ times}})^{\mathrm{T}}$$

yielding a vector $\mathbf{v}$ fulfilling the Mangasarin-Fromowitz constraint qualification.

An infinitesimal motion from $\mathbf{x}$ along the vector $\mathbf{v}$ fulfilling the Mangasarin-Fromowitz constraint qualification stays within the feasible region.

Lemma 3.2. *If the constraint functions satisfy the constraint qualification at the local minimum $\mathbf{x}^*$, then the Lagrange multiplier in the Karush-Kuhn-Tucker Lagrange multiplier rule can be rescaled such that $\lambda_0 = 1$.*

Proof. Suppose that the constraint qualification holds at the local minimum $\mathbf{x}^* = \mathbf{0}$ and $\lambda_0 = 0$. First we show that at least one of the nonnegative multipliers μ_j is not 0. Assume this claim is false. Because there is a unit vector of multipliers fulfilling the Lagrange multiplier rule, there is at least one of the λ_i that is not 0 and we have

$$\sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{0}) = \mathbf{0} \ .$$

This contradicts the linear independence of the $\mathrm{d}h_i(\mathbf{x})$.

The constraint qualification implies the existence of $\mathbf{v}$ with

$$\sum_{i=1}^{k} \mu_i \mathrm{d}g_i(\mathbf{0})\mathbf{v} < 0 \text{ and } \sum_{i=1}^{m} \lambda_i \mathrm{d}h_i(\mathbf{0})\mathbf{v} = 0 \ .$$

This implies $\lambda_0 \neq 0$, because multiplying both sides of the Lagrange multiplier rule with $\mathbf{v}$ yields $\lambda_0 \mathrm{d}f(\mathbf{0})\mathbf{v} + \sum_{i=1}^{k} \mu_i \mathrm{d}g_i(\mathbf{0})\mathbf{v} = \mathbf{0}\mathbf{v} = \mathbf{0}$, and thus the vector of multipliers can be rescaled by dividing by λ_0 to prove the lemma.

3.2.2 Sufficient Conditions for a Minimum

Lemma 3.3. *Let f be a convex function on a convex set $M \subset \mathbb{R}^m$. If $\mathbf{x}^*$ is a local minimum of f, then it is a global minimum of f and the set $\{\mathbf{u} \in M \,|\, f(\mathbf{u}) = f(\mathbf{x}^*)\}$ is convex. If f is strictly convex, $\mathbf{x}^*$ is a single global minimum.*

Proof. If $f(\mathbf{u}) \leq f(\mathbf{x}^*)$ and $f(\mathbf{z}) \leq f(\mathbf{x}^*)$ then

$$f(\theta\mathbf{u} + (1-\theta)\mathbf{z}) \leq \theta f(\mathbf{u}) + (1-\theta) f(\mathbf{z}) \leq f(\mathbf{x}^*)$$

for any $\theta \in [0,1]$ and thus $\{\mathbf{u} \in M \,|\, f(\mathbf{u}) \leq f(\mathbf{x}^*)\}$ is convex.

Assume that $f(\mathbf{u}) < f(\mathbf{x}^*)$. Then for $\theta > 0$ the inequality above is also strict. Setting $\mathbf{z} = \mathbf{x}^*$ and letting θ tend to 0 contradicts that $\mathbf{x}^*$ is a local optimum and thus $\mathbf{x}^*$ is a global optimum.

Lemma 3.4. *Let f be a differentiable function on a convex set $M \subset \mathbb{R}^m$. If $\mathbf{x}^* \in M$ satisfies $df(\mathbf{x}^*)(\mathbf{z} - \mathbf{x}^*) \geq 0$ for every point $\mathbf{z} \in M$, then $\mathbf{x}^*$ is a global minimum of f.*

Proof. This follows immediately from Lemma 3.3.

Theorem 3.2. *Suppose the functions h_i are affine and the functions g_j and f are convex and differentiable. Then a feasible point $\mathbf{x}$ satisfying the Lagrange multiplier rule with $\lambda_0 = 1$ furnishes a global minimum of f.*

Proof. The feasible region is convex (see Lemma A.1) and thus we can apply Lemma 3.4. Let $\mathbf{z}$ be another feasible point, then the vector $\mathbf{v} = \mathbf{z} - \mathbf{x}^*$ satisfies

$$\mathrm{d}h_i(\mathbf{x}^*)\mathbf{v} = h_i(\mathbf{z}) - h_i(\mathbf{x}^*) = 0$$

for all equality constraints h_i. Further, we have

$$\mathrm{d}g_i(\mathbf{x}^*)\mathbf{v} \leq g_i(\mathbf{z}) - g_i(\mathbf{x}^*) \leq 0$$

for all inequality constraints g_i active at $\mathbf{x}^*$, because of Lemma A.2 and $g_i(\mathbf{x}^*) = 0$.

Transposing the multiplier equality and taking the inner product on both sides with $\mathbf{v}$,

$$\left(\nabla f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(\mathbf{x}^*) + \sum_{i=1}^{k} \mu_i \nabla g_i(\mathbf{x}^*)\right)^{\mathrm{T}} \mathbf{v} = \mathbf{0}^{\mathrm{T}}\mathbf{v} \ ,$$

leads to

$$\mathrm{d}f(\mathbf{x}^*)\mathbf{v} = -\sum_{i=1}^{k} \mu_i \mathrm{d}g_i(\mathbf{x}^*)\mathbf{v} \ ,$$

which is the sufficient condition of Lemma 3.4.

Definition 3.2 (Lagrangian Function). *For a given primal optimization problem, the* Lagrangian function *is defined as*

$$L(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{w}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{w}) + \sum_{i=1}^{k} \mu_i g_i(\mathbf{w}) \ .$$

We summarize Theorem 3.1, Lemma 3.2, and Theorem 3.2 in the following theorem:

Theorem 3.3 (Kuhn-Tucker). *Given an optimization problem*

$$\begin{array}{lll} \textit{minimize} & f(\mathbf{w}) & \mathbf{w} \in M \\ \textit{subject to} & g_i(\mathbf{w}) \leq 0 & i = 1, \ldots, k \\ & h_i(\mathbf{w}) = 0 & i = 1, \ldots, m \end{array}$$

with differentiable convex functions f, g_i, and affine functions h_j ($i = 1, \ldots, k$ and $j = 1, \ldots, m$) defined on a convex set $M \subset \mathbb{R}^n$.

Suppose the differentials $dh_i(\mathbf{x}^*)$ of the equality constraints and the differentials $dg_i(\mathbf{x}^*)$ of the active inequality constraints are linearly independent at $\mathbf{x}^* \in M$.

Sufficient and necessary conditions for $\mathbf{x}^*$ *to be the global minimum is the existence of* $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ *and* $\boldsymbol{\mu} \in \mathbb{R}^k$ *with*

$$\begin{aligned} \frac{\partial L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)}{\partial \mathbf{w}} &= \mathbf{0} \ , \\ \frac{\partial L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)}{\partial \boldsymbol{\lambda}} &= \mathbf{0} \ , \\ \mu_i^* g_i(\mathbf{x}^*) &= 0 \ \textit{for } i = 1, \ldots, k \ , \\ g_i(\mathbf{x}^*) &\leq 0 \ \textit{for } i = 1, \ldots, k \ , \\ \mu_i^* &\geq 0 \ \textit{for } i = 1, \ldots, k \ . \end{aligned}$$

3.2.3 Dual Problems

Lemma 3.5 (Saddle Point Inequalities). *Suppose that the constraint qualification at a global minimum* $\mathbf{x}^*$ *of a primal optimization problem holds, the functions* h_i *are affine, and the functions* g_j *and* f *are convex and differentiable.*

For an optimum $\mathbf{x}^*$ *of the primal optimization problem and corresponding Lagrange multipliers* $\boldsymbol{\lambda}^*$ *and* $\boldsymbol{\mu}^*$ *(for* $\lambda_0 = 1$*) it holds*

$$L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) \leq L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \leq L(\mathbf{w}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$$

for all $\mathbf{w} \in M$ *and* $\boldsymbol{\lambda} \in \mathbb{R}^m, \mu_1, \ldots, \mu_k \in \mathbb{R}_0^+$.

Proof. The Lagrange multiplier rule shows that the derivative of $L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ with respect to $\mathbf{w}$ is zero. Thus, the point $\mathbf{x}^*$ is also a global minimum of $L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$. This proves the right inequality.

Because $h_i(\mathbf{x}^*) = 0$ for all i and $\mu_j^* g_j(\mathbf{x}^*) = 0$ for all j we have

$$L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = -\sum_{i=1}^{k} \mu_i g_i(\mathbf{x}^*) \geq 0 \ .$$

This proves the left inequality.

Definition 3.3 (Dual Optimization Problem). *Given a primal optimization problem with Lagrangian function* L*, the* dual optimization problem *is defined by*

$$\begin{array}{lll} \textit{maximize} & \vartheta(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\mathbf{w} \in M} L(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) & \boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\mu} \in \mathbb{R}^k \\ \textit{subject to} & \mu_i \geq 0 & i = 1, \ldots, k \ . \end{array}$$

Lemma 3.6. *Suppose that the constraint qualification at a global minimum* $\mathbf{x}^*$ *of a primal optimization problem holds, the functions* h_i *are affine, and the functions* g_j *and* f *are convex and differentiable.*

Then the duality gap

$$f(\mathbf{x}^*) - \sup_{\boldsymbol{\lambda}\in\mathbb{R}^m,\mu_1,\ldots,\mu_k\in\mathbb{R}_0^+} \left[\inf_{\mathbf{w}\in M} L(\mathbf{w},\boldsymbol{\lambda},\boldsymbol{\mu})\right]$$

is zero.

Proof. The left saddle point inequality directly implies

$$\sup_{\boldsymbol{\lambda}\in\mathbb{R}^m,\mu_1,\ldots,\mu_k\in\mathbb{R}_0^+} \left[\inf_{\mathbf{w}\in M} L(\mathbf{w},\boldsymbol{\lambda},\boldsymbol{\mu})\right] \leq \sup_{\boldsymbol{\lambda}\in\mathbb{R}^m,\mu_1,\ldots,\mu_k\in\mathbb{R}_0^+} L(\mathbf{x}^*,\boldsymbol{\lambda},\boldsymbol{\mu}) \leq L(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) = f(\mathbf{x}^*)$$

and the right one

$$f(\mathbf{x}^*) = L(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) \leq \inf_{\mathbf{w}\in M} L(\mathbf{w},\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) \leq \sup_{\boldsymbol{\lambda}\in\mathbb{R}^m,\mu_1,\ldots,\mu_k\in\mathbb{R}_0^+} \left[\inf_{\mathbf{w}\in M} L(\mathbf{w},\boldsymbol{\lambda},\boldsymbol{\mu})\right] \quad .$$

3.3 Bibliographical Remarks

A thorough introduction to gradient-based optimization is beyond the scope of this script. The material in this chapter on constraint optimization is taken from the recommend textbook by Lange [2004].

4

Support Vector Machines

This chapter introduces support vector machines (SVMs, also referred to as support vector networks, Boser et al., 1992, Cortes and Vapnik, 1995). Support vector machines are state-of-the-art in machine learning for pattern recognition, in particular for binary classification. They follow a well thought out strategy to construct a hypothesis. First, the input data is transformed into a (often high dimensional) feature space, which is a Hilbert space and therefore possesses convenient mathematical properties. In this space, the data points are then linearly separated by a hyperplane. If the data are separable, a hard margin SVM chooses the hypothesis that has maximum margin w.r.t. the training data points in the sense that the closest distance of a training data point to the hyperplane is maximized. Choosing this hyperplane is not only intuitive, but also well founded by results from statistical learning theory. Soft margin SVMs are originally also rooted in the concept of large margin separation. They implement regularized risk minimization, penalizing hyperplanes with large norm in feature space. The SVM learning can be cast into a constrained convex quadratic optimization problem (a *quadratic program*) free from suboptimal local extrema. Last but not least, SVMs make use of the kernel trick. All operations in feature space can be expressed by scalar products in that space, and the feature space is indirectly defined via a kernel function, which is a real-valued function of two elements of the input space that corresponds to the dot product of its arguments mapped to the feature space. As the kernel function can usually be computed very efficiently, all operations in the feature space are carried out using the kernel. Kernel-based algorithms are highly flexible because apart from hyperparameters (such as learning rates and regularization parameters) only the kernel function has to be changed when the problem domain changes.

4.1 Basic Concepts

In the following, basic concepts underlying support vector machines are presented. First, we have a look at linear classification before kernels and the corresponding feature spaces are introduced.

4.1.1 Linear Classification

Binary classification is often performed according to the sign of a scalar discrimination function $f : \mathcal{X} \to \mathbb{R}$. Given a discrimination function, the corresponding hypothesis mapping an input $x \in \mathcal{X}$ to one of two classes with labels $\mathcal{Y} = \{-1, 1\}$ is $h_f(x) = \operatorname{sgn}(f(x))$.

Let us consider the case where $\mathcal{X} \subseteq \mathbb{R}^n$ and the decision function is affine linear, so that we have $f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle + b$. The parameters $(\mathbf{w}, b) \in \mathbb{R}^n \times \mathbb{R}$ determine (non-uniquely) the linear decision function and the corresponding hypothesis. They define a hyperplane (i.e., an $n-1$-dimensional affine subspace) of the input space by $\ker(f) = \{\mathbf{x} \in \mathcal{X} \mid f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle + b = 0\}$. The hyperplane divides the input space into two half-spaces corresponding to the two classes.

There is an *inherent degree of freedom* in the choice of the parameters $(\mathbf{w}, b)$ because all $(c\mathbf{w}, cb)$ lead to the same hyperplane and the same decision boundary for $c \in \mathbb{R}^+$. The absolute value of $f(\mathbf{x}_0)/\|\mathbf{w}\|$ is the Euclidean distance of $\mathbf{x}_0$ from the hyperplane, and the sign indicates on which side (i.e., in which half-space) $\mathbf{x}_0$ lies, see Fig. 4.1.

The normal vector $\mathbf{w}$ is also often referred to as weight vector in analogy to the weights of a *Perceptron*, an abstract model for the computations in neurons [Rosenblatt, 1958].

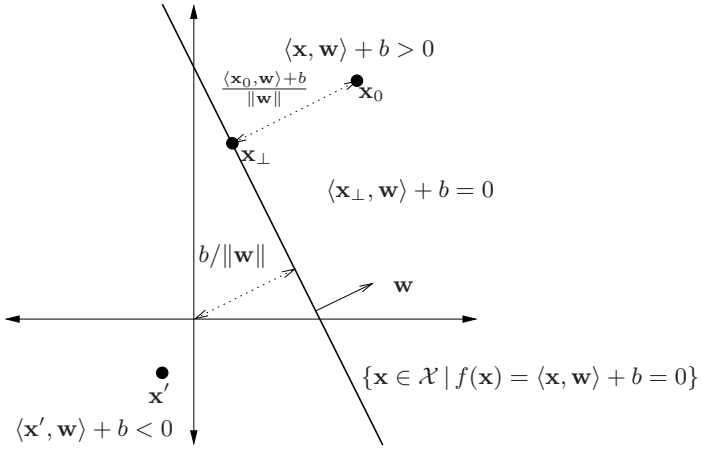


Fig. 4.1. Linear classification according to a decision function $f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle + b$.

Margins. The information about the distance of an input pattern $\mathbf{x}_i$ with label y_i from a hyperplane and whether $\mathbf{x}_i$ lies on the correct side of the hyperplane is reflected by the *margin* of the pattern. The *functional margin* of an example $(\mathbf{x}_i, y_i)$ with respect to a hyperplane $(\mathbf{w}, b)$ is

$$\gamma_i = y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \ .$$

The *geometric margin* of an example $(\mathbf{x}_i, y_i)$ with respect to a hyperplane $(\mathbf{w}, b)$ is

$$\rho_i = y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)/\|\mathbf{w}\| = \gamma_i/\|\mathbf{w}\| \ .$$

A positive margin implies correct classification. The absolute value of the geometric margin of an input pattern corresponds to its distance from the hyperplane. If a finite data set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$ is linearly separable then there exist $(\mathbf{w}, b)$ such that for all $1 \leq i \leq \ell$

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0 \ .$$

This implies

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \gamma$$

for some $\gamma > 0$.

Based on the margin definition for a single pattern, the margin of a hyperplane with respect to a training set S can be defined. The geometric margin ρ_S of a hyperplane $(\mathbf{w}, b)$ with respect to S is $\min_{1 \leq i \leq \ell} \rho_i$ and the functional margin γ_S of the hyperplane with respect to S is $\min_{1 \leq i \leq \ell} \gamma_i$. Given a data set, the hyperplane having maximum margin is called *maximum margin hyperplane* accordingly. Figure 4.2 shows two hyperplanes that correspond to two hypotheses that classify the patterns from a linearly separable set S correctly.

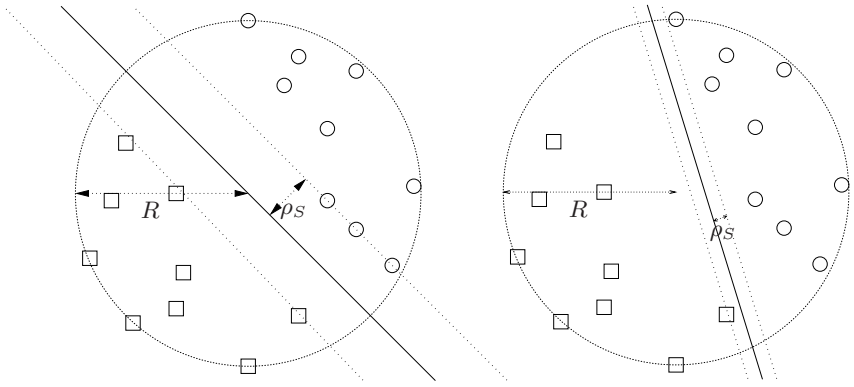


Fig. 4.2. Linear classifiers, the hyperplane in the left figure classifies with maximum margin. The geometric margin with respect to the data set S is denoted by ρ_S, and R stands for the radius of the smallest ball containing S.

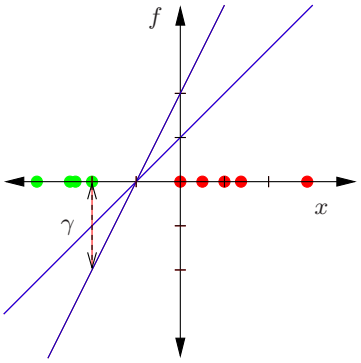


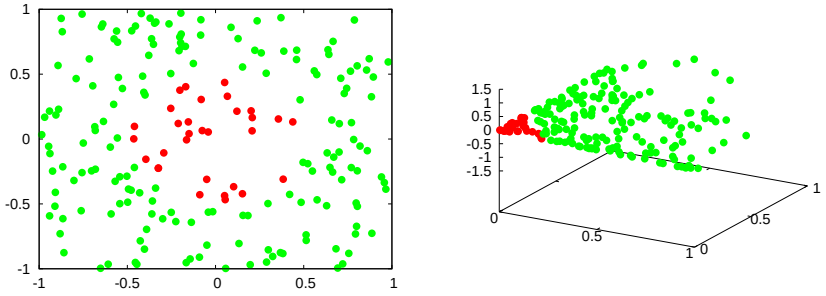
Fig. 4.3. Two decision functions corresponding to the same separating hyperplane but having different functional margins.

4.1.2 Kernels and Reproducing Kernel Hilbert Spaces

In order to apply a machine learning algorithm, it is necessary to encode the input such that it can be handled by the algorithm. To achieve generalization, additionally some notion of similarity and/or dissimilarity or at least neighborhood between patterns is needed. In the following, we assume that the input data are mapped into a metric space (this mapping may be the identity and need not be performed explicitly) and that similarity between patterns is defined according to a metric in that space, which we call metric *feature space*. The classification or regression algorithms operate in this feature space. The mapping to the feature space corresponds to a change in *representation* of the data. The choice of the representation is fundamental for the performance of the learning machine, and our goal is to find a representation that fosters learning and generalization.

Let $\mathcal{X}$ denote the input space, $\mathcal{F}$ the feature space, and Φ: $\mathcal{X} \to \mathcal{F}$ the *feature map*. As a standard example, we consider a polynomial classifier. Let us suppose the n-dimensional input vectors $\mathbf{x} \in \mathcal{X} = \mathbb{R}^n$ are best represented by the dth order products (monomials) of the components x_j of $\mathbf{x}$, that is, by the $x_{j_1} \cdot x_{j_2} \cdot \ldots \cdot x_{j_d}$, for $j_1, \ldots, j_d \in \{1, \ldots, n\}$. If we consider, for example, second order monomials, the feature map is given by $\tilde{\Phi} : \mathbb{R}^2 \to \mathbb{R}^4$ with $\tilde{\Phi}((x_1, x_2)) = (x_1^2, x_2^2, x_1x_2, x_2x_1)$. Alternatively, we can use the feature map $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ with $\Phi((x_1, x_2)) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$, where the order of monomials is not considered and a weighting factor, the role of which will become clear below, is used. Figure 4.4 illustrates an example where Φ makes input data that is not linearly separable in $\mathbb{R}^2$ linearly separable in the feature space. In general, however, both increasing and reducing the dimensionality by applying Φ can be reasonable.

By increasing the dimensionality we can make input patterns linearly separable. However, increasing the dimensionality comes with increasing compu-



(x_1, x_2) $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$

Fig. 4.4. Example of an embedding into a feature space turning linearly non-separable to separable data. The colors indicate different class labels. The feature map $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ changes the representation of input patterns (x_1, x_2) to $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$.

tational costs when doing computations in the feature space. For example, for n dimensions there exist more than $((d+n-1)/d)^d$ monomials of order d. However, many algorithms only require the computation of dot products $\langle \Phi(x), \Phi(x') \rangle$ for $x, x' \in \mathcal{X}$ in feature space. Therefore, the idea is to efficiently compute the dot product by a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the property

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

for all $x, x' \in \mathcal{X}$.

Coming back to the example, for second order monomials we have $k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^2$ and for dth order monomials $k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^d$. This shows that the feature space for a given function k is not unique because the canonical dot product in the feature spaces induced by the two maps $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ with $\Phi((x_1, x_2)) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ and $\tilde{\Phi} : \mathbb{R}^2 \to \mathbb{R}^4$ with $\tilde{\Phi}((x_1, x_2)) = (x_1^2, x_2^2, x_1x_2, x_2x_1)$ can be computed by the same function

$$k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2 = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle = \left\langle \tilde{\Phi}(\mathbf{x}), \tilde{\Phi}(\mathbf{z}) \right\rangle \ .$$

Two questions arise:

- Given some function k, can we construct a feature space $\mathcal{F}$ and a feature map Φ such that k computes the dot product $k(\cdot, \cdot) = \langle \Phi(\cdot), \Phi(\cdot) \rangle$ in $\mathcal{F}$?
- Given a mapping Φ into a feature space $\mathcal{F}$, can we find a function k computing the dot product $k(\cdot, \cdot) = \langle \Phi(\cdot), \Phi(\cdot) \rangle$ in $\mathcal{F}$?

In the following, both questions are answered positively for *positive definite kernel functions* k.

Positive definite kernels

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \neq \emptyset$ that for all $m \in \mathbb{N}$ and all $x_1, \ldots, x_m \in \mathcal{X}$ gives rise to a positive definite *Gram matrix* is called a *positive definite kernel* or *positive definite kernel function*, where the attribute "positive definite" is usually omitted. The Gram or kernel matrix of k with respect to $x_1, \ldots, x_m$ is the $m \times m$ matrix $\mathbf{K}$ with elements $K_{ij} = k(x_i, x_j)$. A real symmetric $m \times m$ matrix $\mathbf{K}$ satisfying

$$\forall c_1, \ldots, c_m \in \mathbb{R} : \sum_{i,j=1}^{m} c_i c_j K_{ij} \geq 0 \qquad \left(\forall \mathbf{x} \in \mathbb{R}^m : \mathbf{x}^T \mathbf{K} \mathbf{x} \geq 0\right)$$

is called positive definite. If the inequality is replaced by a strict inequality for $\mathcal{X} \neq \mathbf{0}$, we talk of strictly positive definite kernels and matrices.[1]

Now we show that for any feature map $\Phi : \mathbf{x} \to \mathcal{F}$ we can indeed find a kernel function computing the dot product in $\mathcal{F}$ and that for any kernel function k we can find a feature space $\mathcal{F}$ such that k computes the dot product in $\mathcal{F}$.

Feature map to kernel. Finding a kernel for a feature map is straightforward. Given $\Phi : \mathcal{X} \to \mathcal{F}$ we define

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle \quad ,$$

which is positive definite for all $m \in \mathbb{N}, c_i \in \mathbb{R}, x_i \in \mathcal{X}$, $i = 1, \ldots, m$, because

$$\sum_{i,j=1}^{m} c_i c_j k(x_i, x_j) = \left\langle \sum_{i=1}^{m} c_i \Phi(x_i), \sum_{j=1}^{m} c_j \Phi(x_j) \right\rangle = \left\| \sum_{i=1}^{m} c_i \Phi(x_i) \right\|^2 \geq 0 \quad .$$

Kernel to feature map. For every input space $\mathcal{X} \neq \emptyset$ and positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a canonical feature space can be constructed, which is a function space (i.e., each function in this space can be thought of as a point). We proceed in four steps [Schölkopf and Smola, 2002], and define a map Φ given a kernel k, turn the image of Φ into a vector space, define a dot product in that space, and finally show that $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$.

The canonical feature map is given by

$$\begin{aligned} \Phi : \quad & \mathcal{X} \to \mathbb{R}^{\mathcal{X}} = \{f : \mathcal{X} \to \mathbb{R}\} \\ & \Phi(x)(\cdot) = k(\cdot, x) \end{aligned}$$

and the corresponding feature space by

$$\mathcal{S}_k = \operatorname{span}\{k(x, \cdot) \,|\, x \in \mathcal{X}\} \quad .$$

[1] Sometimes the strict inequality is required already for positive definiteness, and the weaker form is referred to as positive semi-definiteness.

It consists of all functions

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$$

for any $m \in \mathbb{N}$ and $x_1, \ldots, x_m \in \mathcal{X}$, $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$. This canonical mapping into a function space is illustrated in Fig. 4.5. There is a canonical dot product

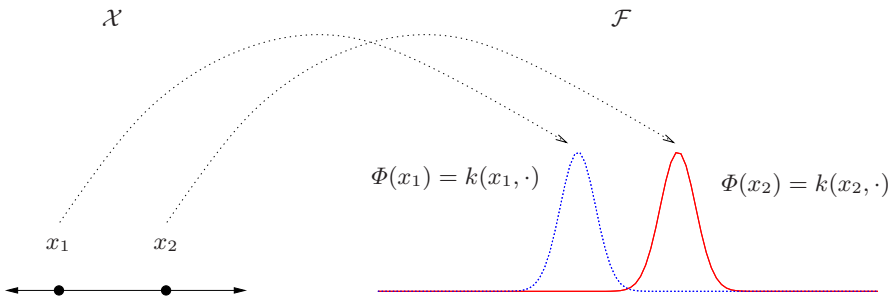


Fig. 4.5. Canonical feature map.

in this space, defined as

$$\langle f, g \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j) = \sum_{j=1}^{m'} \beta_j f(x'_j) = \sum_{i=1}^{m} \alpha_i g(x_i)$$

for two elements

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \qquad \text{and} \qquad g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j) \ ,$$

which is clearly symmetric, bilinear, and positive definite. It has the *reproducing property*

$$\langle k(\cdot, x), f \rangle = f(x)$$

implying $\langle \Phi(x), \Phi(x') \rangle = \langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x')$ and is well defined. The latter can be seen from

$$\begin{aligned} \langle f, g \rangle = \left\langle \sum_{i=1}^{m} \alpha_i k(\cdot, x_i), \sum_{j=1}^{m'} \beta_j k(\cdot, x'_j) \right\rangle &= \sum_{i=1}^{m} \alpha_i \sum_{j=1}^{m'} \beta_j \langle k(x_i, \cdot), k(x_j, \cdot) \rangle \\ &= \sum_{i=1}^{m} \alpha_i g(x_i) \ , \end{aligned}$$

which shows the independence of $\langle f, g\rangle$ from the representation of g (and of f by symmetry).

We can turn our canonical feature space $\mathcal{S}_k$ into a Hilbert space, which has several convenient mathematical properties. In particular, orthonormal bases and projections onto closed subspaces exist in Hilbert spaces. The function space $\mathcal{S}_k$ is already an inner-product space with norm $\|f\| = \sqrt{\langle f, f\rangle}$ for $f \in \mathcal{S}_k$. For a Hilbert space completeness is required, which means that every Cauchy sequence has to be convergent. To achieve this property, we simply add all limit points of Cauchy sequences in $\mathcal{S}_k$ to the feature space resulting in the completion $\overline{\mathcal{S}_k}$. The space $\mathcal{F}_k = \overline{\mathcal{S}_k}$ is the *reproducing kernel Hilbert space* (RKHS) induced by the kernel function k [Aronszajn, 1950]. The RKHS uniquely determines k because the reproducing property and symmetry imply $\langle k(x,\cdot), k'(z,\cdot)\rangle = k(x,z) = k'(x,z)$ for any $x, z \in \mathcal{X}$.

Orthogonal projections. Let $\mathcal{F}$ be a Hilbert space and M a closed subspace. Then every $\mathbf{x} \in \mathcal{F}$ can be written uniquely as $\mathbf{x} = \mathbf{z} + \mathbf{z}_\perp$, where $\mathbf{z} \in M$ and $\langle \mathbf{z}_\perp, \mathbf{t}\rangle = 0$ for all $\mathbf{t} \in M$. The vector $\mathbf{z}$ is the unique element of M minimizing $\|\mathbf{x} - \mathbf{z}\|$; it is called the (orthogonal) *projection* of $\mathbf{x}$ onto M. In a RKHS $\mathcal{F}$ with kernel k on $\mathcal{X}$, the projection of $\Phi(x) = k(x,\cdot)$, $x \in \mathcal{X}$, onto $\operatorname{span}\{\mathbf{w}\}$ for $\mathbf{w} \in \mathcal{F}$ is given by

$$\frac{\langle \mathbf{w}, \Phi(x)\rangle}{\|\mathbf{w}\|^2}\mathbf{w} \quad .$$

Universal kernels. Universality is an important property of kernel functions (see section 4.2.3). A continuous kernel k on a compact metric space $\mathcal{X}$ is called *universal* if the space of all functions induced by the kernel is dense in the space of all continuous functions on $\mathcal{X}$ [Steinwart, 2002b, Steinwart and Christmann, 2008]. That is, for every $\epsilon > 0$ and every continuous function g on $\mathcal{X}$, there exists an element $\mathbf{w}$ of the RKHS induced by k such that $\sup_{x\in\mathcal{X}} |\langle \mathbf{w}, \Phi(x)\rangle - g(x)| < \epsilon$.

Gaussian kernels. The most frequently used kernel functions if $\mathcal{X} = \mathbb{R}^n$ are Gaussian kernels, especially when little is known about the problem at hand. General Gaussian kernels can be defined as

$$k(\mathbf{x}, \mathbf{z}) = e^{-(\mathbf{x}-\mathbf{z})^T\mathbf{M}(\mathbf{x}-\mathbf{z})} \quad ,$$

where $\mathbf{M}$ is a positive definite matrix. The standard choice is $\mathbf{M} = \frac{1}{2\sigma^2}\mathbf{I}$, with unit matrix $\mathbf{I}$ and scalar parameter $\sigma \in \mathbb{R}^+$.

Gaussian kernels are universal [Steinwart, 2002a] (for compact $\mathcal{X} \subseteq \mathbb{R}^n$). The radial Gaussian kernel

$$k(\mathbf{x}, \mathbf{z}) = \exp\left(-\|\mathbf{x} - \mathbf{z}\|^2/(2\sigma^2)\right)$$

has the properties that the image of each element of $\mathcal{X}$ has unit length (i.e., $k(\mathbf{x}, \mathbf{x}) = 1$ for $\mathbf{x} \in \mathcal{X}$), the images of the elements of $\mathcal{X}$ lie in the same orthant (we have $\cos(\angle(\Phi(\mathbf{x}), \Phi(\mathbf{z}))) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z})\rangle = k(\mathbf{x}, \mathbf{z}) > 0$), and the induced Gram matrices have full rank. The latter means that for any set of distinct points $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ and any $\sigma \in \mathbb{R}^+$ the matrix $\mathbf{K}$ with $K_{ij} = \exp\left(-(\|\mathbf{x}_i - \mathbf{x}_j\|^2)/(2\sigma^2)\right)$ has full rank.

Making kernels from kernels. In order to design a valid kernel—and therefore a representation—for a particular class of problems, it is useful to know how to derive new positive definite kernels from other positive definite kernels Let $k_1, k_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $k_3 : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ be valid kernel functions and $a \in \mathbb{R}^+$, $f : \mathcal{X} \to \mathbb{R}$, and $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^m$. Then the following functions are positive definite kernels:

1. $k(x, z) = ak_1(x, z)$,
2. $k(x, z) = k_1(x, z) + k_2(x, z)$,
3. $k(x, z) = k_1(x, z)k_2(x, z)$,
4. $k(x, z) = e^{k_1(x,z)}$,
5. $k(x, z) = f(x)f(z)$,
6. $k(x, z) = k_3(\boldsymbol{\phi}(x), \boldsymbol{\phi}(z))$,
7. $k(x, z) = k_1(x, z)/\sqrt{k_1(x, x)k_1(z, z)}$.

In the last case, $k(x, x) \neq 0$ is assumed for all $x \in \mathcal{X}$.

Kernel trick

Using kernels allows for an efficient formulation of nonlinear variants of any algorithm that can be expressed in terms of dot products. This is known as the *kernel trick*: "Given an algorithm formulated in terms of a positive definite kernel k, one can construct an alternative algorithm by replacing k by an alternative kernel" [Schölkopf and Smola, 2002].

4.2 Support Vector Machines for Classification

Now we are in the position to derive SVMs [Boser et al., 1992, Cortes and Vapnik, 1995]. We restrict our considerations to machines for binary classification. First, linear and non-linear *hard margin* SVMs are discussed, which implement strict maximum margin separation. Then we turn to regularized *soft margin* SVMs. The section ends with a discussion of selected important properties of soft margin SVMs.

4.2.1 Hard Margin Support Vector Machines

Let us assume linearly separable, non-trivial training data $S \in (\mathbb{R}^n \times \{-1, 1\})^\ell$ and that we want to derive an affine decision function. There are infinitely many possible decision functions leading to zero empirical risk. Which one should we choose? It appears to be a reasonable strategy to pick a decision function that corresponds to the hyperplane having maximum margin w.r.t. S, see Fig. 4.2. A large safety margin promises robust classification and good generalization. As we will see later, there are good, less hand-waving reasons for that choice.

Linear Hard Margin SVM

Computing the parameters of the maximum margin hyperplane is exactly what a *linear hard margin SVM* does. Training this machine means solving the following constraint optimization problem:

$$\begin{aligned} \text{maximize}_{\mathbf{w},b} \quad & \rho = \gamma/\|\mathbf{w}\| \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \gamma \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

For $\gamma > 0$ the constraints ensure positive margins of the patterns in S and therefore that the training patterns are classified correctly. We get rid of the inherent degree of freedom (see section 4.1.1) by fixing $\gamma = 1$ (alternatively we could instead fix $\|\mathbf{w}\| = 1$) and maximize $\rho = 1/\|\mathbf{w}\|$ subject to $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$, $1, \ldots, \ell$, which is equal to

$$\begin{aligned} \text{minimize}_{\mathbf{w},b} \quad & \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \ , \ \ i = 1, \ldots, \ell \ . \end{aligned}$$

This is the primal form of the optimization problem underlying linear hard margin SVM learning. The hyperplane $(\mathbf{w}^*, b^*)$ solving this optimization problem has the maximum margin $\rho = 1/\|\mathbf{w}^*\|$.

In practice, we solve the dual form of this problem (see section 3), which is given by

$$\begin{aligned} \text{maximize}_{\boldsymbol{\alpha}} \quad & \inf_{\mathbf{w},b} \quad L(\mathbf{w}, b, \boldsymbol{\alpha}) \\ \text{subject to} \quad & \alpha_i \geq 0 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

with Lagrangian:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i [y_i(\langle \mathbf{w}, x_i \rangle + b) - 1]$$

The Karush-Kuhn-Tucker Theorem requires

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \qquad \frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0$$

yielding

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{w} - \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i \quad \text{and} \quad \frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i y_i \ .$$

This implies

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i \quad \text{and} \quad 0 = \sum_{i=1}^{\ell} \alpha_i y_i \ .$$

The first equality shows that $\mathbf{w}$ can be written as expansion in terms of training input patters. We substitute both equations, which are valid at an optimum, into the Lagrangian. The result is the following dual form of the linear hard margin SVM learning algorithm. For linearly separable data $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$ the solution of

$$\begin{aligned} \text{maximize}_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

defines the decision function $f(\mathbf{x}) = \langle \mathbf{w}^*, \mathbf{x} \rangle + b^*$ by

$$\begin{aligned} \mathbf{w}^* &= \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i \\ b^* &= -\frac{\max_{y_i=-1}(\langle \mathbf{w}^*, \mathbf{x}_i \rangle) + \min_{y_i=1}(\langle \mathbf{w}^*, \mathbf{x}_i \rangle)}{2} \ , \end{aligned}$$

which corresponds to the maximum margin hyperplane with geometric margin $\rho = 1/\|\mathbf{w}^*\|$. Accordingly, an input pattern $\mathbf{x}$ is assigned to one of the two classes by the decision rule $\operatorname{sgn} f(\mathbf{x})$.

The *Karush-Kuhn-Tucker (KKT) complementarity condition* (see Theorem 3.3) requires

$$\alpha_i^* [y_i (\langle \mathbf{w}^*, \mathbf{x}_i \rangle + b^*) - 1] = 0$$

for $i = 1, \ldots, \ell$. This condition, which can be used to derive b^* once $\mathbf{w}^*$ has been computed, shows that the solution is sparse because for every data point not having exactly a functional margin of one the corresponding α-coefficient must be zero. Therefore we can write

$$f(\mathbf{x}, \boldsymbol{\alpha}^*, b^*) = \sum_{i=1}^{\ell} y_i \alpha_i^* \langle \mathbf{x}_i, \mathbf{x} \rangle + b^* = \sum_{\mathbf{x}_i \in \mathrm{SV}} y_i \alpha_i^* \langle \mathbf{x}_i, \mathbf{x} \rangle + b^* \ ,$$

where we defined $\mathrm{SV} = \{\mathbf{x}_i \,|\, \alpha_i^* \neq 0\}$, the set of *support vectors*. That is, the sum only runs over a subset of training patterns, the support vectors, which determine the decision function. In general, however, the number of support vectors still grows with the number of training patterns, see Theorem 4.6.

For each support vector $\mathbf{x}_j \in \mathrm{SV}$ it follows that

$$y_j f(\mathbf{x}_j, \boldsymbol{\alpha}^*, b^*) = y_j \left(\sum_{\mathbf{x}_i \in \mathrm{SV}} y_i \alpha_i^* \langle \mathbf{x}_i, \mathbf{x}_j \rangle + b^* \right) = 1$$

and therefore the norm of the weight vector is given by

$$\langle \mathbf{w}^*, \mathbf{w}^* \rangle = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$
$$= \sum_{\mathbf{x}_j \in \mathrm{SV}} \alpha_j^* y_j \sum_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i^* y_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{\mathbf{x}_j \in \mathrm{SV}} \alpha_j^* (1 - y_j b^*) = \sum_{\mathbf{x}_j \in \mathrm{SV}} \alpha_j^* \ .$$

Thus, the geometric margin of a successfully trained linear hard margin SVM is $\rho = \left(\sqrt{\sum_{\mathbf{x}_j \in \mathrm{SV}} \alpha_j^*} \right)^{-1}$.

Non-linear Hard Margin SVM

The input patterns enter the linear SVM learning only in form of scalar products. To turn the linear hard margin SVM algorithm into a non-linear pattern recognition method, we can therefore simply apply the kernel trick (see section 4.1.2).

Given a kernel function k on an arbitrary input space $\mathcal{X}$, the *non-linear hard margin SVM* learning algorithms works as follows. For training data $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ linearly separable in the feature space $\mathcal{F}_k$ induced by k the solution $\boldsymbol{\alpha}^*$, b^* of

$$\begin{aligned} \text{maximize}_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \ , \ \ \alpha_i \geq 0 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

leads to the decision rule $\mathrm{sgn}(f(x))$ with $f(x) = \sum_{i=1}^{\ell} y_i \alpha_i^* k(x_i, x) + b^*$ that is equivalent to the maximum margin hyperplane in $\mathcal{F}_k$ with margin $\rho = 1/\|\mathbf{w}^*\| = \left(\sqrt{\sum_{x_j \in \mathrm{SV}} \alpha_j^*} \right)^{-1}$.

Generalization Bound for Hard Margin SVMs

So far, intuitive arguments were given why classification according to the maximum margin hyperplane is a reasonable approach. These arguments can be supported by results from statistical learning theory. For example, based on the Rademacher complexities of the involved function classes (see section 2.2.1), bounds on the generalization error can be derived depending on the margin of the separating hyperplane.

For a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, a sample $S = \{x_1, \ldots, x_\ell\}$ of points from $\mathcal{X}$ leading to Gram matrix $\mathbf{K}$, and a positive constant B, let

$$\mathcal{F}_B = \left\{ x \mapsto \sum_{i=1}^{\ell} \alpha_i k(x_i, x) \;\middle|\; \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \leq B^2 \right\}$$
$$= \{ x \mapsto \langle \mathbf{w}, \Phi(x) \rangle \;|\; \mathbf{w} = \sum_{i=1}^{\ell} \alpha_i k(x_i, \cdot)\,,\, \|\mathbf{w}\| \leq B \}$$

denote the class of bounded linear functions in the span of the samples. It holds:

Theorem 4.1. *If $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel and $S = \{x_1, \ldots, x_\ell\}$ is a sample of points from $\mathcal{X}$, then the empirical Rademacher complexity of the class $\mathcal{F}_B$ satisfies*

$$\hat{R}_\ell(\mathcal{F}_B) \leq \frac{2B}{\ell} \sqrt{\sum_{i=1}^{\ell} k(x_i, x_i)} = \frac{2B}{\ell} \sqrt{\mathrm{tr}(\mathbf{K})} \;.$$

Here $\mathrm{tr}(\mathbf{K})$ denotes the trace of the Gram matrix. A proof of this theorem can for example be found in the textbook by Shawe-Taylor and Cristianini [2004, proof of Theorem 4.15]. Hard margin SVMs without bias parameter b produce decision functions in $\mathcal{F}_B$ with corresponding hypotheses $\mathrm{sgn} \circ \mathcal{F}_B = \{\mathrm{sgn} \circ g \,|\, g \in \mathcal{F}_B\}$.

Now this result is extended to bound the 0-1 loss. To this end, functions over $\mathcal{Z} = \mathcal{X} \times \{-1, 1\}$ are considered. The 0-1 loss of a decision function $g : \mathcal{X} \to \mathbb{R}$ can be expressed by the binary function $(x, y) \mapsto \mathrm{sgn}(-yg(x))$. In the following, a bound on the empirical Rademacher complexity of these functions under the assumption that $g \in \mathcal{F}_B$ is derived. Let each pattern in the set $S = \{(x_1, y_1) \ldots, (x_\ell, y_\ell)\}$ be drawn i.i.d. from some distribution over $\mathcal{Z}$. Because for any function class $\mathcal{F}$ and constant $c \in \mathbb{R}$ it holds $\hat{R}_\ell(c\mathcal{F}) = |c| \hat{R}_\ell(\mathcal{F})$, the empirical Rademacher complexity of the function class $\mathcal{F} = \{-yg(x) \,|\, g \in \mathcal{F}_B\}$ has the same upper bound as $\mathcal{F}_B$,

$$\hat{R}_\ell(\mathcal{F}) \leq \frac{2B}{\ell} \sqrt{\sum_{i=1}^{\ell} k(x_i, x_i)} = \frac{2B}{\ell} \sqrt{\mathrm{tr}(\mathbf{K})} \;.$$

Under the assumption of separation with a margin of ρ, the empirical Rademacher complexity of $\mathrm{sgn} \circ \mathcal{F}$ can be bounded by $\hat{R}_\ell(\mathrm{sgn} \circ \mathcal{F}) \leq 2\hat{R}_\ell(\mathcal{F})/\rho$. Combined with Theorem 2.2, this implies:

Theorem 4.2. *Fix $\rho, \delta > 0$. Assume a training set $S = \{(x_1, y_1) \ldots, (x_\ell, y_\ell)\}$ drawn i.i.d. according to some distribution over $\mathcal{Z}$. The generalization error of a hard margin SVM trained on S that linearly separates all training data points with a geometric margin $\left(\sqrt{\sum_{\mathbf{x}_j \in SV} \alpha_j^*}\right)^{-1} > \rho$ is bounded by*

$$\frac{4}{\ell\rho} \sqrt{\mathrm{tr}(\mathbf{K})} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}}$$

with probability $1 - \delta$.

A proof of this theorem is given by Shawe-Taylor and Cristianini [2004, see Theorem 7.26].

4.2.2 Soft Margin Support Vector Machines

It is always possible to ensure by a considerate choice of the kernel function k that any finite training data set (without contradictory samples) can be correctly linearly separated in the RKHS induced by k. If the kernel function is universal (see section 4.1.2), there exists a feasible solution for the hard margin SVM optimization problem [e.g., Steinwart and Christmann, 2008]. For the very same reason, hard margin SVMs are rarely used in practice, because forcing a learning machine to correctly separate all training data is in general not reasonable. This is easy to see, because if the optimal risk value that can be achieved by a hypothesis in the considered hypothesis space (see section 2.1) is not zero, the learning machine should be allowed to make mistakes on the training data. In other words, hard margin SVMs succumb to overfitting. Therefore, *soft margin SVMs* were invented, which incorporate an additional regularization term for controlling the complexity of the classifier in the training process.

Linear Soft Margin SVM

To "soften" the SVM algorithm, we allow training patterns to have a functional margin smaller than one and to even be misclassified. To measure the amount by which a margin is violated, we introduce a *margin slack variable* for every training pattern. Let us first consider $\mathcal{X} = \mathbb{R}^d$. For a fixed value $\gamma > 0$, we can define the margin slack variable ξ_i of an example $(\mathbf{x}_i, y_i)$ with respect to the hyperplane $(\mathbf{w}, b)$ and target margin γ as

$$\xi((\mathbf{x}_i, y_i), (\mathbf{w}, b), \gamma) = \xi_i = \max(0, \gamma - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)) \ .$$

In our SVM formulation the target margin is one. Therefore, the variable ξ_i is zero if $\mathbf{x}_i$ is correctly classified by the decision rule $h(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$ and has at least a distance of $1/\|\mathbf{w}\|$ from the hyperplane $(\mathbf{w}, b)$. If it is closer to the hyperplane but still on the correct side, ξ_i is positive and smaller than one. If ξ_i is larger than one then $\mathbf{x}_i$ is classified wrongly, see Fig. 4.6.

Using the concept of margin slack variables, we turn the primal form of the hard margin SVM algorithms, which minimizes $\frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle$ subject to $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$, into the *linear soft-margin 2-norm SVM* problem

$$\begin{aligned} \text{minimize}_{\boldsymbol{\xi}, \mathbf{w}, b} \quad & \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + \frac{C}{2} \sum_{i=1}^{\ell} \xi_i^2 \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

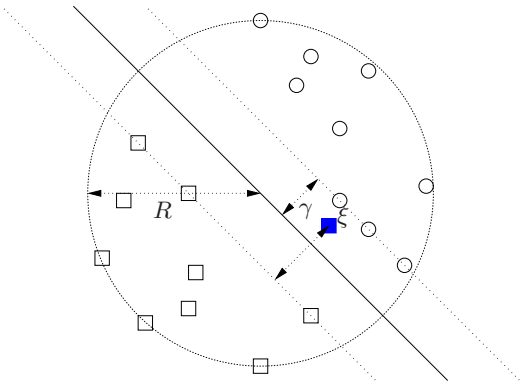


Fig. 4.6. Soft margin classification. The filled square indicates a wrongly classified pattern with slack variable ξ larger than γ.

or the *linear soft-margin 1-norm SVM* problem

$$\begin{aligned} \text{minimize}_{\boldsymbol{\xi},\mathbf{w},b} \quad & \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + C\sum_{i=1}^{\ell}\xi_i \\ \text{subject to} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i \ , \ \ i = 1, \ldots, \ell \\ & \xi_i \geq 0 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

with positive regularization parameter C. In the first formulation, the regularization term is quadratic in the margin violations, while in the second problem the regularization depends linearly on the ξ_i.

2-norm soft margin SVM. The dual form of the linear 2-norm soft margin SVM has the Lagrangian

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{\ell}\xi_i^2 - \sum_{i=1}^{\ell}\alpha_i(y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) - 1 + \xi_i)$$

leading to the KKT conditions

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{\ell}\alpha_i y_i \mathbf{x}_i = 0 \qquad \rightarrow \mathbf{w} = \sum_{i=1}^{\ell}\alpha_i y_i \mathbf{x}_i \ , \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \boldsymbol{\xi}} &= C\boldsymbol{\xi} - \boldsymbol{\alpha} = \mathbf{0} \ , \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial b} &= \sum_{i=1}^{\ell}\alpha_i y_i = 0 \ , i = 1, \ldots, \ell \ , \end{aligned}$$

with the complementarity condition:

$$\alpha_i^*[y_i(\langle \mathbf{w}^*, \mathbf{x}_i\rangle + b^*) - 1 + \xi_i^*] = 0 \ , \ \ i = 1, \ldots, \ell \ .$$

As for the hard margin SVM, the KKT complementary condition indicates the sparseness of the resulting kernel classifier. With these conditions, the Lagrangian can be rewritten as

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) &= \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{\ell}\xi_i^2 - \sum_{i=1}^{\ell}\alpha_i(y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) - 1 + \xi_i) \\ &= \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle - \frac{1}{2C}\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle \\ &= \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \left(\langle \mathbf{x}_i, \mathbf{x}_j\rangle + \frac{1}{C}\delta_{ij}\right) \ . \end{aligned}$$

The last equations shows that the effect of the regularization amounts to adding $1/C$ to the eigenvalues of Gram matrix. Clearly, for C approaching infinity, which corresponds to an infinitely strong penalty for margin violations, we retrieve the hard margin SVM.

1-norm soft margin SVM. Now we derive the optimization problem for 1-norm soft margin SVMs in the same way. For a linear 1-norm soft margin SVM, the Lagrangian is given by

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\alpha_i[y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) - 1 + \xi_i] - \sum_{i=1}^{\ell}\beta_i\xi_i$$

under the constraints $\alpha_i, \beta_i \geq 0, i = 1, \ldots, \ell$. The KKT conditions are

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{w}}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathbf{w} - \sum_{i=1}^{\ell}\alpha_i y_i x_i = 0 \ , \\ &\frac{\partial L}{\partial \xi_i}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = C - \alpha_i - \beta_i = 0 \ \ i = 1, \ldots, \ell \ , \\ &\frac{\partial L}{\partial b}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{\ell}\alpha_i y_i = 0 \ , \end{aligned}$$

with KKT complementarity conditions

$$\begin{aligned} &\alpha_i[y_i(\langle w, x_i\rangle + b) - 1 + \xi_i] = 0 \ , \\ &\beta_i\xi_i = 0 \ , \\ &\xi_i(\alpha_i - C) = 0 \quad (\text{using } \beta_i = C - \alpha_i \text{ and } \beta_i\xi_i = 0) \end{aligned}$$

for all $i = 1, \ldots, \ell$. The complementarity conditions show that the regularization parameter C imposes box constraints on the dual variables α_i. From $\beta_i = C - \alpha_i$ we get $0 \leq \alpha_i \leq C$ for all i. From the condition $\xi_i(\alpha_i - C) = 0$ we directly see that if x_i violates the margin then the box constraint is active and $\alpha_i = C$.

Non-linear Soft Margin SVM

As in the case of the hard margin formulation, we turn the algorithms into non-linear methods using the kernel trick. Figure 4.7 shows a schema of the resulting soft margin SVM learning.

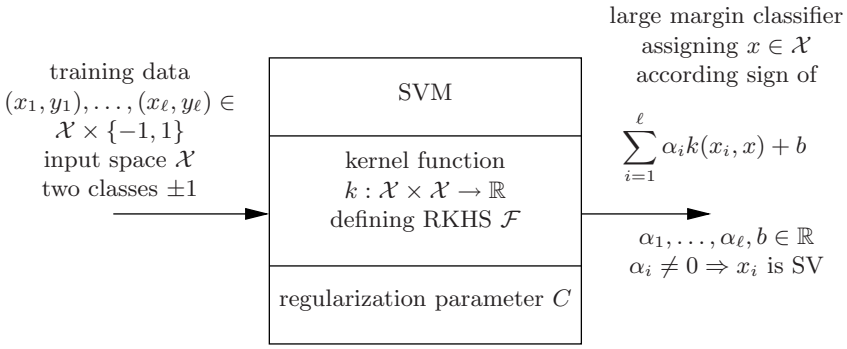


Fig. 4.7. Schema of a soft-margin support vector machine.

Non-linear 2-norm soft margin SVM. We get the following optimization problem for the non-linear 2-norm soft margin SVM. Given a training data set $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ and a kernel k, the solution $\boldsymbol{\alpha}^*$, b^* of

$$\begin{aligned} \text{maximize}_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \left(k(x_i, x_j) + \frac{1}{C} \delta_{ij} \right) \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \ , \ \ \alpha_i \geq 0 \ , \ \ i = 1, \ldots, \ell \end{aligned}$$

leads to the decision rule $\operatorname{sgn}(f(x))$ with $f(x) = \sum_{i=1}^{\ell} y_i \alpha_i^* k(x_i, x) + b^*$. The offset b^* is chosen such that $y_i f(x_i) = 1 - \alpha_i^*/C$ for some i with $\alpha_i \neq 0$. The slack variables w.r.t. the hyperplane $(\mathbf{w}^*, b^*)$ are defined relative to the *geometric* margin

$$\rho = \frac{1}{\|\mathbf{w}^*\|} = \left(\sqrt{\sum_{x_j \in \mathrm{SV}} \alpha_j^* - \frac{1}{C} \langle \boldsymbol{\alpha}^*, \boldsymbol{\alpha}^* \rangle} \right)^{-1} \ .$$

Non-linear 1-norm soft margin SVM. Non-linear 1-norm soft margin SVMs are derived in a similar way as in the 2-norm case. For a training data set $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ and a kernel k, the solution $\boldsymbol{\alpha}^*$, b^* of

$$\begin{aligned} \text{maximize}_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \ , \ C \geq \alpha_i \geq 0 \ , \ i = 1, \ldots, \ell \end{aligned}$$

leads to the decision rule $\text{sgn}(f(x))$ with $f(x) = \sum_{i=1}^{\ell} y_i \alpha_i^* k(x_i, x) + b^*$, where b^* is chosen so that $y_i f(x_i) = 1$ for any i with $C > \alpha_i > 0$ and the slack variables of the "corresponding hyperplane" in the feature space defined by kernel k are defined relative to the *functional margin* with value one and the *geometric* margin $\rho = 1/\|\mathbf{w}^*\| = 1/\sqrt{\sum_{x_j \in \text{SV}} \alpha_j^*}$.

Again, we see that for C going to infinity (i.e., no upper bound on the dual variables) we retrieve the hard margin SVM algorithm. The 1-norm regularization is often preferred to the 2-norm regularization because it leads to sparser solutions.

4.2.3 More on the Soft Margin Support Vector Machines Approach to Pattern Recognition

For the hard margin SVM, both intuitive as well as theoretical arguments that support this approach to pattern recognition were given. These arguments do not directly carry over to soft margin machines. In the following, however, strong theoretical support for soft-margin SVMs for pattern recognition is provided. First, an alternative derivation of SVMs from the viewpoint of regularized risk minimization is presented and the Representer Theorem is given, which justifies to restrict the space of candidate hypotheses in the RKHS to the span of the support vectors. Then the fundamental result on the consistency of SVMs is stated. Finally, a generalization bound for the soft margin SVMs is quoted.

SVMs and Regularization

Soft margin SVMs lack the sound interpretation as strict maximum margin classifiers because they allow non-zero slack variables. In fact, they are better motivated from the viewpoint of regularization. Let us consider the primal, w.l.o.g. 1-norm, soft margin SVM optimization problem

$$\begin{aligned} \text{minimize}_{\boldsymbol{\xi}, \mathbf{w}, b} \quad & \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{\ell} \xi_i \\ \text{subject to} \quad & y_i (\langle \mathbf{w}, \Phi(x_i) \rangle + b) \geq 1 - \xi_i \ , \ i = 1, \ldots, \ell \\ & \xi_i \geq 0 \ , \ i = 1, \ldots, \ell \ . \end{aligned}$$

For a fixed $\mathbf{w}$, the optimal slack variables are

$$\xi_i = \max(0, 1 - y_i(\langle \mathbf{w}, \Phi(x_i)\rangle + b)) \ .$$

Solving this quadratic program can be reformulated as regularized risk minimization. To this end, we define our loss function to be the *hinge loss* $L(y, \hat{y}) = \max(0, 1 - y\hat{y})$ and consider classes $\mathcal{H}_k$ (the RKHS induced by k) and $\mathcal{H}_k^b = \{f(x) = \langle \mathbf{w}, \Phi(x)\rangle + b \,|\, \mathbf{w} \in \mathcal{H}_k, b \in \mathbb{R}\}$ with $\Phi(x) = k(x, \cdot)$ for $x \in \mathcal{X}$. The norm $\|.\|_k$ on $\mathcal{H}_k$ can be inherited to $\mathcal{H}_k^b$, where it is only a semi-norm. If we assume that $\mathbf{w}$ is of the form $\sum_{i=1}^{\ell} \alpha_i k(x_i, \cdot)$, training the SVM corresponds to solving the regularized risk minimization problem

$$\text{minimize}_{f \in H_k^b} \quad \frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, f(x_i)) + \vartheta_\ell \|f\|_k^2$$

with $\vartheta_\ell = (2\ell C)^{-1}$. But why should we assume that $\mathbf{w}$ is in the span of the training examples? That this is indeed reasonable follows from the Representer Theorem, here stated for the hypothesis space $\mathcal{H}_k$:

Theorem 4.3 (Representer Theorem). *Let $\Omega : [0, \infty[\to \mathbb{R}$ denote a strictly monotonically increasing function, $\mathcal{H}_k$ a RKHS with kernel k on $\mathcal{X}$, and $L : \bigcup_{\ell=1}^{\infty} (\mathcal{X} \times \mathbb{R})^\ell \to [0, \infty[$ a loss function. Then each minimizer $f \in \mathcal{H}_k$ of the regularized loss*

$$L\left((y_1, f(x_1)), \ldots, (y_\ell, f(x_\ell))\right) + \Omega(\|f\|_k^2)$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{\ell} \alpha_i k(x_i, x) \ .$$

This means, if only hypotheses in the RKHS are considered, there is always one best hypothesis, which minimizes an empirical risk term and a regularization term monotonically increasing in the norm of the decision function, that can be written as an expansion in terms of the training examples.

For $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i k(x_i, \cdot)$ and $f(x) = \langle \mathbf{w}, \Phi(x)\rangle + b$ it holds $\|f\|_k^2 = \langle \mathbf{w}, \mathbf{w}\rangle$. The question arises, why is it reasonable to penalize the norm of the hypothesis' normal (weight) vector? To see this, let us consider a hypothesis $h(\cdot) = \langle \mathbf{w}, \Phi(\cdot)\rangle + b \in \mathcal{H}_k^b$. Assume $\|k(x, \cdot) - k(z, \cdot)\|_k \leq d_{\mathcal{X}}(x, z) c_k$ for $x, z \in \mathcal{X}$ and some metric $d_{\mathcal{X}}$ on $\mathcal{X}$, that is, the kernel is Lipschitz continuous with constant $c_k \in \mathbb{R}^+$. We have $\|h(x) - h(z)\|_k = \|\langle \mathbf{w}, k(x, \cdot) - k(z, \cdot)\rangle\|_k \leq \|\mathbf{w}\|_k \|k(x, \cdot) - k(z, \cdot)\|_k$, and therefore

$$\|h(x) - h(z)\|_k \leq \|\mathbf{w}\|_k d_{\mathcal{X}}(x, z) c_k \ .$$

That is, the hypothesis is Lipschitz with a constant proportional to the norm of the weight vector. Hence, the smaller the norm the smoother the hypothesis. A much more rigorous discussion of SVMs and regularization can be found in the textbook by Schölkopf and Smola [2002].

Consistency of Support Vector Machines

Steinwart [2002a,b] proved that 1-norm soft margin SVMs are consistent provided that the regularization parameter is chosen in a distinct manner depending on the training sample size and that the kernel is universal (see section 4.1.2). In other words, for any universal kernel we can find parameters that guarantee asymptotic convergence of the SVM solutions to the best possible solution with increasing number of i.i.d. training samples.

In particular, for Gaussian kernels the following result can be derived [Steinwart and Christmann, 2008, p. 238]. Considering the SVM in the regularized risk minimization formulation given above in section 4.2.3 with Gaussian kernels, the SVM learning algorithm is consistent if the regularization parameter ϑ_ℓ is chosen dependent on the sample size ℓ such that $\lim_{\ell\to\infty} \vartheta_\ell = 0$ and $\lim_{\ell\to\infty} \vartheta_\ell^2 \ell = \infty$.

Generalization Bounds

This section is concluded with an extension of the generalization bound presented in section 4.2.1 to soft margin SVMs. It can be proved using the techniques presented in the book by Shawe-Taylor and Cristianini [2004].

Theorem 4.4. *Fix $\rho, \delta > 0$. Assume examples $S = \{(x_1, y_1) \ldots, (x_\ell, y_\ell)\}$ drawn i.i.d. according to some distribution over $\mathcal{Z} = \mathcal{X} \times \{-1, 1\}$. Let g be the linear discriminant function in a kernel-defined feature space solving the 1-norm soft margin SVM optimization problem (without bias), and let the geometric margin of g w.r.t. S be at least ρ. Then with a probability of at least $1 - \delta$ over the samples of size ℓ it holds*

$$\begin{aligned}\Pr_p(y \neq \operatorname{sgn}(g(x))) &= \mathbb{E}_p\{\mathbf{1}\{-yg(x) > 0\}\} \\ &\leq \frac{1}{\ell}\sum_{i=1}^{\ell} \xi_i + \frac{4}{\ell\rho}\sqrt{\operatorname{tr}(\mathbf{K})} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}} \ ,\end{aligned}$$

where $\mathbf{K}$ is the kernel matrix for the training set and $\xi_i = \max(0, 1 - y_i g(x_i))$ is the target margin violation of the example (x_i, y_i).

Before we can proove this theorem, we derive a result about the Rademacher complexity of the considered hypothesis class. For a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, a sample $S = \{x_1, \ldots, x_\ell\}$ of points from $\mathcal{X}$, and a positive constant B let

$$\begin{aligned}\mathcal{F}_B &= \left\{ x \mapsto \sum_{i=1}^{\ell} \alpha_i k(x_i, x) \;\middle|\; \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \leq B^2 \right\} \\ &= \{x \mapsto \langle \mathbf{w}, \Phi(x) \rangle \mid \|\mathbf{w}\| \leq B\}\end{aligned}$$

denote the class of bounded linear functions in the span of the samples. Here, Φ is a feature map corresponding to the kernel k.

Theorem 4.5. *If $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel and $S = \{x_1, \ldots, x_\ell\}$ is a sample of points from $\mathcal{X}$, then the empirical Rademacher complexity of the class $\mathcal{F}_B$ satisfies*

$$\hat{R}_\ell(\mathcal{F}_B) \leq \frac{2B}{\ell}\sqrt{\sum_{i=1}^{\ell} k(x_i, x_i)} = \frac{2B}{\ell}\sqrt{\mathrm{tr}(\mathbf{K})} \ .$$

Proof. We quote the proof by Shawe-Taylor and Cristianini [2004]:

$$\begin{aligned}\hat{R}_\ell(\mathcal{F}_B) &= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{F}_B}\left|\frac{2}{\ell}\sum_{i=1}^{\ell}\sigma_i h(x_i)\right|\right] \\ &= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\|\mathbf{w}\|\leq B}\left|\left\langle \mathbf{w}, \frac{2}{\ell}\sum_{i=1}^{\ell}\sigma_i \Phi(x_i)\right\rangle\right|\right]\end{aligned}$$

using the Cauchy-Schwarz inequality

$$\begin{aligned}&\leq \frac{2B}{\ell}\mathbb{E}_{\boldsymbol{\sigma}}\left[\left\|\sum_{i=1}^{\ell}\sigma_i\Phi(x_i)\right\|\right] \\ &= \frac{2B}{\ell}\mathbb{E}_{\boldsymbol{\sigma}}\left[\left(\left\langle\sum_{i=1}^{\ell}\sigma_i\Phi(x_i), \sum_{i=1}^{\ell}\sigma_i\Phi(x_i)\right\rangle\right)^{1/2}\right]\end{aligned}$$

by Jensen's inequality

$$\begin{aligned}&\leq \frac{2B}{\ell}\left(\mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{i,j=1}^{\ell}\sigma_i\sigma_j k(x_i, x_j)\right]\right)^{1/2} \\ &= \frac{2B}{\ell}\left(\sum_{i=1}^{\ell} k(x_i, x_i)\right)^{1/2} \ .\end{aligned}$$

Proof (of Theorem 4.4). First, we combine hypothesis class and loss function to a function class, for which we will compute the Rademacher complexity. We consider hypotheses $h : \mathcal{X} \to \{-1, 1\}$ based on a class $\mathcal{F}$ of discriminating functions $g : \mathcal{X} \to \mathbb{R}$ given by

$$h(x) = \mathrm{sgn}(g(x)) \in \{-1, 1\} \ .$$

Using the Heavyside function

$$\Theta(a) = \begin{cases} 1 & \text{if } a \geq 0, \\ 0 & \text{otherwise} \end{cases}$$

we express the 0-1-loss as

$$L(x, y, h(x)) = \Theta(-yg(x))$$

and the corresponding risk (i.e., the quantity for which we want to derive the bound) as

$$\Pr_p(y \neq \operatorname{sign}(g(x))) = \mathbb{E}_p[\Theta(-yg(x))] \ .$$

Now we define the auxiliary class

$$\tilde{\mathcal{F}} = \{(x, y) \mapsto -yg(x) \,|\, g \in \mathcal{F}\} \ .$$

Combining 0-1-loss and the class of discrimination functions thus corresponds to the functions

$$\Theta \circ \tilde{\mathcal{F}} = \left\{\Theta \circ f \,|\, f \in \tilde{\mathcal{F}}\right\} \ .$$

If we know the (empirical) Rademacher complexity of $\Theta \circ \tilde{\mathcal{F}}$ or at least an upper bound of it, we can derive a bound for the risk using Lemma 2.1 . The Heavyside function is not continuous, not even Lipschitz. In order to be able to apply inequality 4 of Theorem 2.1, we therefore bound the loss function by a Lipschitz continuous function. The function

$$\mathcal{A}(a) = \begin{cases} 1 & \text{if } a > 0, \\ 1 + a & \text{if } -1 \leq a \leq 0, \\ 0 & \text{otherwise} \end{cases}$$

dominates the Heavyside function

$$\mathcal{A}(a) \geq \Theta(a) = \begin{cases} 1 & \text{if } a \geq 0, \\ 0 & \text{otherwise} \end{cases}$$

and has the desired property of being Lipschitz with constant 1. We have

$$\mathbb{E}_p[\Theta(f(x, y))] \leq \mathbb{E}_p[\mathcal{A}(f(x, y))] \ .$$

Using this inequality and applying Lemma 2.1 to functions of the class $(\mathcal{A} - 1) \circ \tilde{\mathcal{F}} = \{(x, y) \mapsto \mathcal{A}(-yg(x)) - 1 \,|\, g \in \mathcal{F}\}$ shows that with a probability of at least $1 - \delta$

$$\begin{aligned} \mathbb{E}_p[\Theta(f(x, y)) - 1] &\leq \mathbb{E}_p[\mathcal{A}(f(x, y)) - 1] \\ &\leq \hat{\mathbb{E}}_S[\mathcal{A}(f(x, y)) - 1] + \hat{R}_\ell((\mathcal{A} - 1) \circ \tilde{\mathcal{F}}) + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}} \ . \end{aligned}$$

If we add 1 to both sides of the equation, we get the expectations of the quantities we are interested in. Further, we have $\mathcal{A}(-y_i g(x_i)) \leq \xi_i$ for all i and thus

$$\mathbb{E}_p[\Theta(f(x,y))] \leq \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i + \hat{R}_\ell((\mathcal{A}-1)\circ\tilde{\mathcal{F}}) + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}} \ .$$

Moreover, we have $(\mathcal{A}-1)(0)=0$ and therefore $\hat{R}_\ell((\mathcal{A}-1)\circ\tilde{\mathcal{F}}) \leq 2\hat{R}_\ell(\tilde{\mathcal{F}})$ (see 4 of Theorem 2.1). Next we derive the empirical Rademacher complexity of $\hat{R}_\ell(\tilde{\mathcal{F}})$ to complete the proof.

We consider discrimination function from class $\mathcal{F}_B$ with $B=\rho^{-1}$. Note that by assumption, the trainend SVM has a margin of at least ρ and therefore the SVM decision function is in $\mathcal{F}_B$. We have for the empirical Rademacher complexity of $\tilde{\mathcal{F}}$

$$\begin{aligned}
\hat{R}_\ell(\tilde{\mathcal{F}}) &= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f\in\tilde{\mathcal{F}}}\left|\frac{2}{\ell}\sum_{i=1}^{\ell}\sigma_i f(x_i)\right|\right] \\
&= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{F}_B}\left|\frac{2}{\ell}\sum_{i=1}^{\ell}\sigma_i y_i g(x_i)\right|\right] \\
&= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{F}_B}\left|\frac{2}{\ell}\sum_{i=1}^{\ell}\sigma_i g(x_i)\right|\right] \\
&= \hat{R}_\ell(\mathcal{F}_B)
\end{aligned}$$

by Theorem 4.5

$$= \frac{2B}{\ell}\sqrt{\mathrm{tr}(\mathbf{K})} = \frac{2}{\ell\rho}\sqrt{\mathrm{tr}(\mathbf{K})} \ .$$

Substituting $2\hat{R}_\ell(\tilde{\mathcal{F}}) = 4(\ell\rho)^{-1}\sqrt{\mathrm{tr}(\mathbf{K})}$ for $\hat{R}_\ell((\mathcal{A}-1)\circ\tilde{\mathcal{F}})$ in the bound for $\mathbb{E}_p[\Theta(f(x,y))]$ derived above yields the desired result.

The square root of the trace of the kernel matrix is upper bounded by ($\sqrt{\ell}$ times) the radius of the smallest ball in $\mathcal{H}_k$ centered at the origin containing all training samples. Therefore, bounds of the above type are often referred to as radius-margin bounds.

4.3 Training Support Vector Machines

The question arises how to solve the constraint convex optimization problems for training SVMs. Bottou and Lin [2007] give an overview over the different methods that have been proposed. In principle, these optimization problems could be solved using standard methods from quadratic programming. However, the special form of the SVM quadratic programs allows to devise specialized heuristics with better time and space complexity.

In the following, we concentrate on training 1-norm soft margin SVMs, because they are most popular and most frequently used in the later chapters.

From the variety of different approaches to SVM training, we restrict ourselves to *decomposition algorithms* [Osuna et al., 1997, Platt, 1999], which are known to be highly efficient.

4.3.1 Decomposition Algorithms

Arguably the most prominent algorithms for solving SVM optimization problems are decomposition methods [Osuna et al., 1997, Joachims, 1999, Platt, 1999, Lin, 2001, Keerthi and Gilbert, 2002, Hush and Scovel, 2003, Fan et al., 2005, Glasmachers and Igel, 2006, List, 2008, List and Simon, 2009]. These methods decompose the learning problem into subproblems. The strategy is to iteratively solve dual optimization problems restricted to a subset B, the so called *working set*, of variables. Algorithm 4.1 illustrates this procedure.

Decomposition Algorithm

```
1 α ← feasible starting point
2 repeat
3 |   select working set B
4 |   solve quadratic program restricted to B resulting in α̂
5 |   α ← α̂
6 until stopping criterion is met
```

Alg. 4.1: Canonical decomposition algorithm.

The 1-norm soft margin SVM dual optimization problem restricted to a working set $B \subset \{1, \ldots, \ell\}$ is defined as

$$\begin{aligned} \text{maximize}_{\hat{\boldsymbol{\alpha}}} \quad & \mathcal{D}(\hat{\boldsymbol{\alpha}}) = \sum_{i=1}^{\ell} \hat{\alpha}_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \hat{\alpha}_i \hat{\alpha}_j y_i y_j k(x_i, x_j) \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \hat{\alpha}_i y_i = 0 \\ & \forall i \in \{1, \ldots, \ell\} : 0 \leq \hat{\alpha}_i \leq C \\ & \forall i \notin B : \hat{\alpha}_i = \alpha_i \ . \end{aligned}$$

Figure 4.8 visualizes this restricted optimization problem for a working set $B = \{i, j\}$ of two variables. It is convenient to use the shortcut notation $K_{ij} = k(x_i, x_j)$ for Gram matrix entries and

$$g_i = \frac{\partial \mathcal{D}(\boldsymbol{\alpha}^*)}{\partial \alpha_i} = 1 - y_i \sum_{j=1}^{\ell} y_j \alpha_j K_{ij}$$

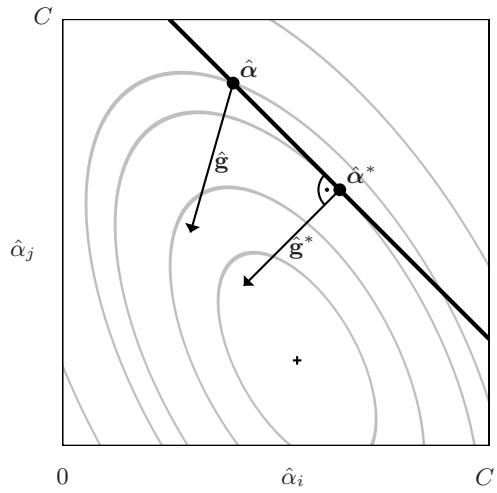


Fig. 4.8. Two-dimensional subproblem for a 1-norm soft margin SVM. The restriction of the feasible region to the line segment in the box is due to the equality constraint and the box constraints. The points $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\alpha}}^*$ are feasible solutions of the subproblem, $\hat{\boldsymbol{\alpha}}^*$ is optimal. The corresponding gradients are denoted by $\hat{\mathbf{g}}$ and $\hat{\mathbf{g}}^*$.

for the gradient of the problem. Further, the index sets

$$\begin{aligned} I_{\text{up}} &= \{i \,|\, y_i\alpha_i < b_i\} \qquad (y_i\alpha_i \text{ may increase}) \quad \text{and} \\ I_{\text{down}} &= \{i \,|\, y_i\alpha_i > a_i\} \qquad (y_i\alpha_i \text{ may decrease}) \end{aligned}$$

are defined with

$$[a_i, b_i] = \begin{cases} [0, C] & \text{if } y_i = +1 \\ [-C, 0] & \text{if } y_i = -1 \end{cases} .$$

A support vector x_i is called *bounded* if $\alpha_i = C$, otherwise it is called *free* (and $i \in I_{\text{up}} \wedge i \in I_{\text{down}}$).

Optimality and Stopping Criterion

The optimization problem is approximately solved by iterative algorithms, which require a *stopping criterion*. To define a meaningful termination condition, a "quantitative" optimality criterion is derived. For $i \in I_{\text{up}}$ and $j \in I_{\text{down}}$ we define $\boldsymbol{\alpha}^\epsilon \in \mathbb{R}^\ell$ for $\epsilon > 0$ to be

$$\alpha_k^\epsilon = \alpha_k + \epsilon[\mathbf{u}_{ij}]_k = \alpha_k + \begin{cases} +\epsilon y_k & \text{if } k = i \\ -\epsilon y_k & \text{if } k = j \\ 0 & \text{otherwise} \end{cases} ,$$

where $[\mathbf{u}_{ij}]_k$ denotes the kth component of the ℓ-dimensional vector $\mathbf{u}_{ij}$, which equals $\mathbf{0}$ except in its ith and jth component, where it is equal to y_k and $-y_k$, respectively. If $\boldsymbol{\alpha}^*$ is optimal, we have

$$\mathcal{D}(\boldsymbol{\alpha}^\epsilon) - \mathcal{D}(\boldsymbol{\alpha}^*) = \epsilon(y_i g_i^* - y_j g_j^*) + o(\epsilon) \ .$$

This leads to the necessary *optimality criterion*

$$\exists r \in \mathbb{R} : \max_{i \in I_{\text{up}}} y_i g_i^* \leq r \leq \min_{j \in I_{\text{down}}} y_j g_j^*$$

or equivalently

$$\exists r \in \mathbb{R} : \forall k : \begin{cases} \alpha_k^* = C & \text{if } g_k^* > y_k r \\ \alpha_k^* = 0 & \text{if } g_k^* < y_k r \end{cases} \ .$$

To show that the optimality criterion is also sufficient, let us consider some feasible solution $\boldsymbol{\alpha}^*$ of $\mathcal{D}$ and pick

$$\mathbf{w}^* = \sum_{i=1}^{\ell} y_i \alpha_i^* k(x_i, \cdot) \ , \ b^* = r \ , \ \xi_i^* = \max\{0, g_i^* - y_i r\} \ .$$

Now we compute the duality gap

$$\mathcal{P}(\boldsymbol{\xi}^*, \mathbf{w}^*, b^*) - \mathcal{D}(\boldsymbol{\alpha}^*) = C \sum_{i=1}^{\ell} \xi_i^* - \sum_{i=1}^{\ell} \alpha_i^* g_i^* = \sum_{i=1}^{\ell} (C\xi_i^* - \alpha_i^* g_i^*) \ ,$$

where $\mathcal{P}(\boldsymbol{\xi}^*, \mathbf{w}^*, b^*)$ stands for the corresponding primal optimization problem. Because $C\xi_i^* - \alpha_i^* g_i^* = -y_i \alpha_i^* r$ we get

$$\mathcal{P}(\boldsymbol{\xi}^*, \mathbf{w}^*, b^*) - \mathcal{D}(\boldsymbol{\alpha}^*) = -r \sum_{i=1}^{\ell} y_i \alpha_i^* = 0 \ .$$

Thus, the duality gap vanishes and $\boldsymbol{\alpha}^*$ is indeed optimal. As a secondary action, setting $b^* = r$ is a good way to determine b^*.

The optimality criterion can be efficiently computed by

$$\max_{i \in I_{\text{up}}} y_i g_i^* - \min_{j \in I_{\text{down}}} y_j g_j^* \leq 0 \ .$$

In practice, this condition is softened to the stopping criterion

$$\max_{i \in I_{\text{up}}} y_i g_i - \min_{j \in I_{\text{down}}} y_j g_j \leq \epsilon$$

for $\epsilon > 0$. This stopping condition is defined in terms of the dual optimization problem. That it is indeed a meaningful criterion w.r.t. the accuracy of the solution to the primal problem has been shown by List [2008], List and Simon [2009].

Recomputing Gradient and Stopping Criterion

The gradients play an important role in the SMO algorithm, for instance for computing the stopping criterion. It is convenient to start the algorithm at $\boldsymbol{\alpha} = \mathbf{0}$, because this is always a feasible point and the gradient at $\boldsymbol{\alpha} = \mathbf{0}$ can easily be computed, it always evaluates to $\mathbf{1}$. After solving the problem restricted to a working set B, the gradient vector of the full problem has to be adjusted. This can be done incrementally using

$$\forall k \in \{1, \ldots, \ell\} : g_k \leftarrow g_k - y_k \sum_{i \in B} y_i (\hat{\alpha}_i - \alpha_i) K_{ik} \ .$$

Sequential Minimal Optimization

For the standard SVM optimization problem the minimum working set size that allows for feasible steps is two because there is exactly one equality constraint. If we change some variable α_i by an amount of $\Delta\alpha_i$ we need to change another variable α_j by $-y_j y_i \Delta\alpha_i$ to get a feasible point. If we choose the minimal working set size, we talk of a SMO-type decomposition algorithm, where SMO stands for *sequential minimal optimization* [Platt, 1999]. In the following, we restrict ourselves to SMO-type algorithms. This scenario is depicted in Fig. 4.8.

The advantage of just considering two variables is that the restricted two-dimensional subproblem can easily be solved analytically. The working set $B = \{i, j\}$ is composed of $i \in I_{\text{up}}$ and $j \in I_{\text{down}}$. The idea is that α_i and α_j are changed such that $y_i \alpha_i$ increases in the SMO step, while $y_j \alpha_j$ decreases accordingly. We therefore assume that $y_i g_i > y_j g_j$, which is ensured by the working set selection algorithms discussed in section 4.3.1. Note that if there is no pair with $i \in I_{\text{up}}$, $j \in I_{\text{down}}$, and $y_i g_i > y_j g_j$ then the stopping condition defined in the previous section is met.

If w.l.o.g. $i < j$, the SMO search direction in the subproblem is given by

$$(0, \ldots, y_i, 0, \ldots, 0, -y_j, 0, \ldots, 0) = \mathbf{u}_{ij} \ .$$

Solving the subproblem in search direction $\mathbf{u}_{ij}$ ignoring box constraints corresponds to maximizing

$$\mathcal{D}(\boldsymbol{\alpha} + \lambda \mathbf{u}_{ij}) - \mathcal{D}(\boldsymbol{\alpha}) = \lambda(y_i g_i - y_j g_j) - \frac{\lambda^2}{2}(K_{ii} + K_{jj} - 2K_{ij})$$

w.r.t. λ. Computing the Newton step gives the optimal λ^*

$$\lambda^* = \frac{y_i g_i - y_j g_j}{K_{ii} + K_{jj} - 2K_{ij}}$$

which is then clipped to meet the box constraints

$$\lambda = \min \left\{ b_i - y_i \alpha_i, y_j \alpha_j - a_j, \frac{y_i g_i - y_j g_j}{K_{ii} + K_{jj} - 2K_{ij}} \right\}$$

and the new coefficients are given by $\boldsymbol{\alpha} + \lambda \mathbf{u}_{ij}$. The general SMO algorithm is sketched in Algorithm 4.2.

Sequential minimal optimization

$\boldsymbol{\alpha} \leftarrow \mathbf{0}$, $\mathbf{g} \leftarrow \mathbf{1}$
repeat
  select indices $i \in I_{\text{up}}$ and $j \in I_{\text{down}}$ with $y_i g_i > y_j g_j$
  $\lambda = \min \left\{ b_i - y_i \alpha_i, y_j \alpha_j - a_j, \dfrac{y_i g_i - y_j g_j}{K_{ii} + K_{jj} - 2K_{ij}} \right\}$
  $\forall k \in \{1, \ldots, \ell\} : g_k \leftarrow g_k - \lambda y_k K_{ik} + \lambda y_k K_{jk}$
  $\alpha_i \leftarrow \alpha_i + y_i \lambda$
  $\alpha_j \leftarrow \alpha_j - y_j \lambda$
until $\max_{i \in I_{\text{up}}} y_i g_i - \min_{j \in I_{\text{down}}} y_j g_j \leq \epsilon$

Alg. 4.2: Sequential minimal optimization.

Working Set Selection

Obviously, the performance of the decomposition algorithm crucially depends on the way the working set is selected in each iteration. Therefore, the key question is how to select the working set B such that

1. much progress is made and only few iterations are needed, and
2. few kernel evaluations are required?

Most Violating Pair Working Set Selection. If we just consider SMO, the most prominent algorithm in current textbooks is *most violating pair* working set selection. It chooses the indices $i \neq j$ according to the following rule (here we do not demand that $i < j$):

1. first index $i = \operatorname{argmax}_{k \in I_{\text{up}}} y_k g_k$
2. second index $j = \operatorname{argmin}_{k \in I_{\text{down}}} y_k g_k$

This realizes *first order working set selection*, in the sense that it maximizes the first-order approximation

$$\mathcal{D}(\boldsymbol{\alpha} + \lambda \mathbf{u}_{ij}) - \mathcal{D}(\boldsymbol{\alpha}) = \lambda(y_i g_i - y_j g_j) + o(\lambda)$$

of the progress in the dual. This algorithm is very efficient, it requires only $O(\ell)$ computations.

Maximum Gain Working Set Selection. As we can compute the progress we make in the dual, it appears to be reasonable to select i and j such that the gain

$$\mathcal{D}(\boldsymbol{\alpha} + \lambda^* \mathbf{u}_{ij}) - \mathcal{D}(\boldsymbol{\alpha}) = \frac{(y_i g_i - y_j g_j)^2}{2(K_{ii} + K_{jj} - 2K_{ij})}$$

is maximized [Fan et al., 2005, Glasmachers and Igel, 2006]. The box constraints may be ignored [Fan et al., 2005]. This has the effect that variables that end up as bounded SVs in the final solution are pushed against the

boundary earlier, making *shrinking* highly efficient (see section 4.3.2). However, checking all $\ell(\ell-1)/2$ index pairs is not feasible. Therefore, Fan et al. [2005] apply the following heuristic:

1. first index i is picked according to most violating pair heuristic
2. second index j is selected to maximize gain

This *second order working set selection* can be implemented such that it only requires $O(\ell)$ computations.

4.3.2 Caching and Shrinking

Gram matrix caching and *shrinking* significantly determine the learning speed in SVM implementations.

Caching Gram matrix entries avoids costly kernel evaluations. Because each iteration of the SMO algorithm works on complete rows/columns of the symmetric kernel matrix, complete rows/columns should be cached [Joachims, 1999]. Moreover, it is reasonable to invest $O(\ell)$ memory to store all diagonal entries K_{ii}, $1 \leq i \leq \ell$, which are needed for solving the restricted subproblem in the SMO algorithm (see line 2 in Algorithm 4.2) and for second-order working set selection.

Shrinking refers to the process of temporally freezing variables of the quadratic program. For example, it is a good heuristic to neglect α-coefficients that are at the boundary of the feasible region and where additionally the corresponding gradient component points strongly away from the feasible region. If the restricted subproblem is solved up to a predefined accuracy, the frozen variables are "unshrinked" and the complete problem has to be considered.

4.3.3 How Long Does Training an SVM Take?

How does SVM training scale with the number of training patterns? List and Simon [2009] derived bounds for the time needed to solve SVM optimization problems up to a certain accuracy when using decomposition algorithms. Here, we give two intuitive bounds [Bottou and Lin, 2007]:

- Let us assume that an oracle tells us the unbounded support vectors $F = \{x_i \,|\, 0 < \alpha_i < C\}$ and the bounded support vectors. Then all what is left to do for computing $\boldsymbol{\alpha}^*$ is solving an $|F|$-dimensional unconstrained optimization problem. This requires $O(|F|^3)$ computations.
- Checking the optimality condition by computing the gradient from scratch takes $O(\ell \cdot |\mathrm{SV}|)$, where $|\mathrm{SV}|$ denotes the number of support vectors. This gives us a strict lower bound.

These are lower bounds. Indeed, SVM training can be said to scale between quadratically and cubically in the number of training points. This is, for example, in accordance with the often cited empirical results obtained by Joachims [1999]. How long training takes in practice depends on the actual Gram matrix

(e.g., its condition number), the number of bounded support vectors (which depends on the choice of C), and the stopping criterion (however, $O(1/\epsilon)$ iterations are sufficient to achieve an accuracy of ϵ as shown by List and Simon, 2009). The dependence on the stopping criterion is crucial, but sometimes ignored in empirical studies. In practice, the training time may even depend on the order of the training samples in the training data set because this order may influence how ties are broken in the working set selection process.

4.3.4 Sparseness of SVMs

Trained SVMs have a slow run-time classification speed if the classification problem is noisy and the sample data set is large. The reason is that although the SVM hypothesis may be sparse in the sense that not all training points become support vectors, the number of support vectors usually grows with the number of training patterns [Steinwart, 2003]. This is intuitively clear, because if the Bayes risk is non-zero the number of wrongly classified training patterns should scale linearly with the number of training patterns, and each wrongly classified training pattern becomes a support vector. More precisely, it holds (see Steinwart and Christmann, 2008, Proposition 8.27):

Theorem 4.6. *Let the hypothesis h (exactly) solve the 1-norm SVM classification problem in the regularized risk minimization formulation stated in section 4.2.3 given a sample set with ℓ patterns. Then the number of support vectors SV is bounded from below by*

$$|SV| \geq \ell \left[2\vartheta_\ell \|h\|^2_{\mathcal{F}_k} + \mathcal{R}_S(h) \right] \quad .$$

When classifying a new input, evaluating the SVM hypothesis takes time linear in the number of support vectors. For each support vector, a kernel evaluation is needed, which can be computational expensive. Thus, with growing training set size, higher Bayes risk, and increasing complexity of kernel evaluations, the time to evaluate an SVM increases. In applications with time constraints, such as driver assistance or biometric systems that must work in real-time, this is a severe problem.

Approximating the kernel expansion after training offers a solution [Romdhani et al., 2004]. Given an SVM decision function

$$f(x) = \sum_{i=1}^{\ell} \alpha_i k(x_i, x) + b$$

the idea is to approximate this expression by a sparser decision function

$$f'(x) = \sum_{i=1}^{\ell'} \alpha'_i k(x'_i, x) + b'$$

with $\ell' \ll \ell$. This approximation is constructed iteratively starting with an approximation based on $\ell' = 1$ vector. In each iteration, the number ℓ' of vectors in the expansion is increased by one. If the input space has a differentiable structure (e.g., $\mathcal{X} = \mathbb{R}^n$) and the kernel function is differentiable, gradient-based optimization can be used to approximate f using f' by adjusting the x'_i together with the α_i and b.

4.4 Bibliographical Remarks

Large parts of this chapter are based on the textbooks by Schölkopf and Smola [2002], Cristianini and Shawe-Taylor [2000], and Shawe-Taylor and Cristianini [2004]. The section on SVM training follows Bottou and Lin [2007]. Steinwart and Christmann [2008] provide an excellent, mathematically rigorous treatment of SVMs.

Exercises

Exercise 4.1. For all types of SVMs presented in this chapter, show how inserting the KKT conditions into the Lagrangian leads to the final dual optimization problems.

Exercise 4.2. Show that if $\boldsymbol{\alpha}^*$ is optimal, it holds

$$\mathcal{D}(\boldsymbol{\alpha}^\epsilon) - \mathcal{D}(\boldsymbol{\alpha}^*) = \epsilon(y_i g_i^* - y_j g_j^*) + o(\epsilon) \ .$$

Exercise 4.3. Show that

$$\exists r \in \mathbb{R} : \max_{i \in I_{\text{up}}} y_i g_i^* \leq r \leq \min_{j \in I_{\text{down}}} y_j g_j^*$$

implies

$$\exists r \in \mathbb{R} : \forall k : \begin{cases} \alpha_k^* = C & \text{if } g_k^* > y_k r \\ \alpha_k^* = 0 & \text{if } g_k^* < y_k r \end{cases} \ .$$

Why does this together with $\xi_i^* = \max\{0, g_i^* - y_i r\}$ and $b^* = r$ imply $C\xi_i^* - \alpha_i^* g_i^* = -y_i \alpha_i^* r$?

A

Mathematical Background

A.1 Concentration Inequalities

Theorem A.1 (McDiarmid). *Let $X_1, \ldots, X_n$ be independent random variables taking values in a set A, and assume that $f : A^n \to \mathbb{R}$ satisfies*

$$\sup_{x_1,\ldots,x_n,\hat{x}_i \in A} |f(x_1, \ldots, x_n) - f(x_1, \ldots, \hat{x}_i, x_{i+1}, \ldots, x_n)| \leq c_i$$

for $1 \leq i \leq n$ and $c_1, \ldots, c_n \in \mathbb{R}_0^+, \sum_{i=1}^n c_i > 0$. Then for all $\epsilon > 0$

$$\Pr\{f(X_1, \ldots, X_n) - \mathbb{E}\{f(X_1, \ldots, X_n)\} \geq \epsilon\} \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n c_i^2}\right) \quad .$$

This implies that with probability of at least $1 - \delta$

$$f(X_1, \ldots, X_n) \leq \mathbb{E}\{f(X_1, \ldots, X_n)\} + \sqrt{-\ln \delta \frac{\sum_{i=1}^n c_i^2}{2}} \quad .$$

Hoeffding's inequality is a special case of McDiarmid's inequality with

$$f(X_1, \ldots, X_n) = \sum_{i=1}^n X_i \quad .$$

Theorem A.2 (Hoeffding's inequality). *If $X_1, \ldots, X_n$ are independent random variables satisfying $X_i \in [a_i, b_i]$, and if we define the random variable $S_n = \sum_{i=1}^n X_i$, then it follows that*

$$\Pr\{|S_n - \mathbb{E}\{S_n\}| \geq \epsilon\} \leq 2 \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

A.2 Convexity

Definition A.1 (Convexity). *A real-valued function $f(\mathbf{w})$ is called convex for $\mathbf{w} \in \mathbb{R}^n$ if for all $\mathbf{w}, \mathbf{u} \in \mathbb{R}^n$ and any $\theta \in]0,1[$*

$$f(\theta\mathbf{w} + (1-\theta)\mathbf{u}) \leq \theta f(\mathbf{w}) + (1-\theta) f(\mathbf{u}) \ .$$

A set M is convex if for all $\mathbf{w}, \mathbf{u} \in M$ and any $\theta \in]0,1[$ we have $(\theta\mathbf{w} + (1-\theta)\mathbf{u}) \in M$.

Lemma A.1. *If $f(\mathbf{x})$ is convex, then the sets $\{\mathbf{x} \,|\, f(\mathbf{x}) < c\}$ and $\{\mathbf{x} \,|\, f(\mathbf{x}) \leq c\}$ are convex. The intersection of convex sets is also convex. Affine functions are convex. Weighted sums of convex functions are convex if the weights are nonnegative.*

Proof (sketch). This follows almost directly from the definitions.

Theorem A.3. *A twice differentiable function f on a convex set $M \subseteq \mathbb{R}^n$ with positive definite Hessian $\mathbf{H} = d^2 f(\mathbf{w})$,*

$$[\mathbf{H}|_{\mathbf{w}}]_{ij} = \frac{\partial^2 f(\mathbf{w})}{\partial w_i \partial w_j} \ ,$$

is (strictly) convex.

Proof. First, we consider the case $n = 1$, where M is an interval. Suppose that $d^2 f(w)$ is strictly positive for all $w \in M$. Take any $a, b \in M$ with $a < b$ and any $\theta \in]0,1[$. Let $w_\theta = \theta a + (1-\theta)b$. By the fundamental Theorem of Calculus we have

$$f(w_\theta) - f(a) = \int_a^{w_\theta} \mathrm{d}f(w)\mathrm{d}w < \mathrm{d}f(w_\theta)(w_\theta - a)$$

and

$$f(b) - f(w_\theta) = \int_{w_\theta}^{b} \mathrm{d}f(w)\mathrm{d}w > \mathrm{d}f(w_\theta)(b - w_\theta) \ ,$$

because $d^2 f(w)$ is strictly positive means that $\mathrm{d}f(w)$ is strictly increasing and thus $\mathrm{d}f(w) < \mathrm{d}f(w_\theta)$ for $w \in [a, w_\theta[$ and $\mathrm{d}f(w) > \mathrm{d}f(w_\theta)$ for $w \in]w_\theta, b]$.

Because $w_\theta - a = (1-\theta)(b-a)$ and $b - w_\theta = \theta(b-a)$, we have

$$\begin{aligned} f(w_\theta) &< f(a) + \mathrm{d}f(w_\theta)(1-\theta)(b-a) \\ f(w_\theta) &< f(b) - \mathrm{d}f(w_\theta)\theta(b-a) \ . \end{aligned}$$

Multiplying both sides of the first inequality with θ and both sides of the second with $1-\theta$ and adding the two together shows convexity

$$\theta f(w_\theta) + (1-\theta) f(w_\theta) = f(w_0) < \theta f(a) + (1-\theta) f(b) \ .$$

Now we go for $n > 1$. Consider $\mathbf{a}, \mathbf{b} \in M$ with $\mathbf{a} \neq \mathbf{b}$ and any $\theta \in]0,1[$ and $\mathbf{w}_\theta = \theta\mathbf{a} + (1-\theta)\mathbf{b}$. It suffices to shown that f is strictly convex on the line segment joining $\mathbf{a}$ and $\mathbf{b}$. Therefore, we parameterize this line segment by $g : [0,1] \to \mathbb{R}$ defined as $g(r) = \mathbf{a} + r(\mathbf{b} - \mathbf{a})$. Now we define $\tilde{f}(r) = f(g(r))$ on $[0,1]$. It is sufficient to show that $\tilde{f}$ is strictly convex. As shown above, we just need to show that $d^2\tilde{f}(r) > 0$ for all $r \in [0,1]$. By the chain rule, letting $\mathbf{v} = \mathbf{b} - \mathbf{a}$ and $\mathbf{w} = \mathbf{a} + r(\mathbf{b} - \mathbf{a})$ we have

$$d^2\tilde{f}(r) = \mathbf{v}^T d^2 f(\mathbf{w})\mathbf{v} \ .$$

The right side is positive because the Hessian is positive definite.

Lemma A.2. *A differentiable function $f(\mathbf{x})$ on a convex open set $M \subset \mathbb{R}^n$ is convex if and only if $f(\mathbf{u}) \geq f(\mathbf{x}) + df(\mathbf{x})(\mathbf{u} - \mathbf{x})$ for all $\mathbf{x}, \mathbf{u} \in M$.*

Proof. Convexity

$$f(\theta\mathbf{w} + (1-\theta)\mathbf{u}) \leq \theta f(\mathbf{w}) + (1-\theta)f(\mathbf{u})$$

implies

$$\frac{f(\theta\mathbf{w} + (1-\theta)\mathbf{u}) - f(\mathbf{w})}{1-\theta} \leq f(\mathbf{u}) - f(\mathbf{w})$$

and thus

$$\frac{f(\mathbf{w} + (1-\theta)(\mathbf{u} - \mathbf{w})) - f(\mathbf{w})}{1-\theta} \leq f(\mathbf{u}) - f(\mathbf{w})$$

Letting θ tend to 1 proves the inequality.

To show the converse, let $\mathbf{z} = \theta\mathbf{w} + (1-\theta)f\mathbf{u}$. By assumption

$$\begin{aligned} f(\mathbf{x}) &\geq f(\mathbf{z}) + \mathrm{d}f(\mathbf{z})(\mathbf{x} - \mathbf{z}) \\ f(\mathbf{u}) &\geq f(\mathbf{z}) + \mathrm{d}f(\mathbf{z})(\mathbf{u} - \mathbf{z}) \ . \end{aligned}$$

Multiplying the first of these inequalities by θ and the second by $1-\theta$ yields and adding the results yield

$$\theta f(\mathbf{x}) + (1-\theta)f(\mathbf{u}) \geq f(\mathbf{z}) + \mathrm{d}f(\mathbf{z})(\mathbf{z} - \mathbf{z}) = f(\mathbf{z})$$

showing convexity.

References

A. Ambroladze, E. Parrado-Hernández, and J. Shawe-Taylor. Complexity of pattern classes and the Lipschitz property. *Theoretical Computer Science*, 382:232–246, 2007.

M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

P. Bartlett and S. Mendelson. Gaussian and Rademacher complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT 1992)*, pages 144–152. ACM, 1992.

L. Bottou and C.-J. Lin. Support vector machine solvers. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA., 2007.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In *Advanced Lectures in Machine Learning*, volume 3176 of *LNAI*, pages 169–207. Springer-Verlag, 2004.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

L. Devroye and L. Gyorfi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1997.

R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using the second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

T. Glasmachers and C. Igel. Maximum-gain working set selection for support vector machines. *Journal of Machine Learning Research*, 7:1437–1466, 2006.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

D. Hush and C. Scovel. Polynomial-time decomposition algorithms for support vector machines. *Machine Learning*, 51:51–71, 2003.

T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, chapter 11, pages 169–184. MIT Press, 1999.

S. S. Keerthi and E. G. Gilbert. Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, 46:351–360, 2002.

K. Lange. *Optimization*. Springer Texts in Statistics. Springer-Verlag, 2004.

C.-J. Lin. On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks*, 12:1288–1298, 2001.

N. List. *Convergence Rates for SVM-Decomposition-Algorithms*. Doctoral thesis, Department of Mathematics, Ruhr-Universität Bochum, 2008.

N. List and H. U. Simon. SVM-optimization and steepest-descent line search. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT 2009)*, 2009. Submitted.

T. M. Mitchell. The discipline of machine learning. Machine Learning Department technical report CMU-ML-06-108, Carnegie Mellon University, 2006.

E. Osuna, R. Freund, and F. Girosi. Improved training algorithm for support vector machines. In J. Principe, L. Giles, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII*, pages 276–285. IEEE Press, 1997.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 12, pages 185–208. MIT Press, 1999.

S. Romdhani, P. Torr, B. Schölkopf, and A. Blake. Efficient face detection by a cascaded support-vector machine expansion. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 460(2051):3283–3297, 2004.

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2002a.

I. Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18(3):768–791, 2002b.

I. Steinwart. Sparseness of support vector machines. *Journal of Machine Learning Research*, 4:2003, 2003.

I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer-Verlag, 2008.

V. Vapnik. *Statistical Learning Theory*. Springer-Verlag, New York, USA, 1998.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 2(16): 264–280, 1971.

D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.