Steyer, Lisa:

Adapting Support Vector Machines to Generalised Interval Data: Implementation of a Suitable Kernel for Convex Sets and its Comparison to a Minimax Approach

Masterarbeit, Sommersemester 2017

Fakultät für Mathematik, Informatik und Statistik

Ludwig-Maximilians-Universität München

https://doi.org/10.5282/ubm/epub.41017

MASTER'S THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Statistics at the Department of Statistics of the Ludwig-Maximilians-Universität München

---

Adapting Support Vector Machines to Generalised Interval Data:

Implementation of a Suitable Kernel for Convex Sets and its Comparison to a Minimax Approach

Author: Lisa Maike Steyer

Supervisor: Prof. Dr. Thomas Augustin
Georg Schollmeyer

Submission date: 7th August 2017

Declaration of Authorship

I hereby certify that this thesis is my own work, unless stated otherwise. All sources of information have been specifically acknowledged. Neither this nor a similar work was previously presented to another examination board or has been published.

München, August 4, 2017 .....

Abstract

Support Vector Machines (SVMs) are a supervised machine learning algorithm, which is widely used for classification of real-valued input data. The aim of this thesis is to adapt SVMs for classification to convex data, which is seen as a generalisation of interval data.

This work first presents the classical theory for empirical SVMs, which is slightly enhanced by considering a general input space. In the third chapter, convex sets as an input space are considered and a suitable kernel function for those sets is proposed. It is shown that the computation of evaluations can be simplified for interval data. Furthermore, the corresponding Gaussian kernel is shown to be universal, hence adapts well for arbitrary data structure.

A decision theoretical approach is discussed in the fourth chapter. In particular, several counterexamples are developed to reveal the inherent difficulties of this approach. Lastly, a classifier based on the minimax rule is compared to the kernel based approach present in the third section.

The R-package 'convexdatasvm', included in the electronic appendix, is developed to illustrate the computation of solutions for both approaches.

Dedicated to the memories of my loving Grandmother Rosa Weiß
(1930-2017)

Acknowledgements

I would like to express my thanks to all those who supported me throughout the work on this thesis. Firstly, I would like to acknowledge Prof. Dr. Thomas Augustin, for providing me the opportunity to write this thesis under his supervision. He and all members of his working group on 'methodological foundations of statistics and their applications' contributed mainly in form of monthly meetings, where they not only gave their valuable advise but also encourage other final year students to give their opinion. I am thankful for the many stimulating discussions arising from those meetings.

Out of this group of supportive people, I would particularly like to single out my supervisor Georg Schollmeyer. His critical remarks have always been spot on. I am also gratefully for him gently steering me in the right the direction whenever I needed it but still giving me the freedom to set my own priorities.

Furthermore I would like to acknowledge the contribution of Florian Kurz and Zoë Vallinga, who discussed patiently every English language issue I came up with.

Contents

1. Attempting to Capture Reality: Convex Data as Precise or as Imprecise Data

> *[W]e have an intuitive knowledge of our own existence, and a demonstrative knowledge of the existence of God; of the existence of anything else, we have no other but a sensitive knowledge; which extends not beyond the objects present to our senses.*
>
> John Locke [8, page 173]

Like many philosophers, Locke argues that to a certain extent, reality does not exist independently of its observer. On the one hand it depends on the observer's concept of the world, on the other hand it is influenced by his perception. To understand what this point of view means for common statistical practice, we firstly notice that most statistical models assume that properties of objects can be measured in real numbers; for example things have a precise height, age and weight.

In practice this underlying assumption, of real values as observable variables, does not guarantee that the perception of these quantities is precise. Consider, for instance a metalworker who needs to measure the length of an iron bar. Using a ruler, he might obtain the length as an interval, with a length of one millimetre. When he uses a calliper instead, he may be able to get a more precise result. Nevertheless, he still observes the length as an interval. Hence, even if the length of an iron bar is a precise real value, we will not be able to detect it.

In common statistical language these observed intervals are classified as imprecise data. This type of data is assumed to consist of subsets of $\mathbb{R}^d$ which include the "true" values (which are themselves real vectors). Imprecise data is seen as an alternative to precise data, which itself consists of real vectors. In the majority of cases a metalworker would not note the length of an iron bar as an interval. He would compare the length of the iron bar to the tick marks on his calliper and write down the closest one. Standard statistical models would then assume the measurement error, the difference between the observed value and the "true" value, to be randomly

distributed according to some distribution. This kind of uncertainty should not be confused with the notion of imprecision.

Below we will see that the concept of real values as properties of objects is itself questionable. Coming back to the example of the iron bar again, we notice that physicists generally do not assume objects such as iron bars to have a precise length. In thermodynamics it is postulated that the length of objects changes with temperature. Similarly, the theory of Brownian motion expects length to change with time on small time scales, even when the temperature is constant. If we consider general relativity the length even depends on the motion of the observer compared to the iron bar. These physical theories provide sufficient reasons to assume that the length of an iron bar is rather an interval than a precise real number.

However, if we believe the length of an object to be an interval and we measure it as an interval, it might be misleading to call this observed interval "imprecise". In particular when both intervals, the assumed "true" one and the observed one, are of about the same size. In this case it is natural to see the observed value as a precise measurement of the actual interval valued length. Analogously to standard procedures for real valued data, the observed interval could be modelled as a random variable. This point of view can also be useful when it is not clear that the assumed "true" value lies within the observed interval. For instance, if the length of the iron bar was close to a tick mark on the calliper, the iron worker might assign it to the wrong interval.

Based on these considerations, there are two meaningful ways to handle convex sets as input data. Either they are seen as imprecise data, hence we assume there is a "true" value within that set, or it is seen as precise data, measuring relevant properties directly up to some randomly distributed noise. The method one chooses mainly depends on the individual's opinion of the character of the observed variables.

Nevertheless, when considering Support Vector Machines (SVMs) as a classification algorithm, there is some reason to see convex data as precise data measured up to some noise. Since the SVM approach neither tries to model the conditional distribution of the label given the input data nor the joint distribution of the input data and the labels, it is often referred to as a "black box method". This means SVMs can be regarded as a flexible model of the relationship between input data and labels. Hence, regardless of the mechanism that led to convex sets as input

data, due to this "black box" property, one can hope SVMs with convex sets as input space perform sufficiently well.

Conversely, treating convex sets as imprecise data relies on the strong assumption that there is some "true" value within the convex set. Since this does not need to hold true for seeing convex sets as precise data, from a philosophical point of view, methods that can cope with this input space directly should be preferred. In Section 3 we will see how a specific kernel can be chosen to adapt Support Vector Machines to the input space

$$X_c = \{A \subset \mathbb{R}^d | A \text{ compact and convex}\}.$$

Here all elements of X_c are generally seen as unrelated input vectors. Therefore, the space of possible decision functions is very large and a priori it is not clear how the choice of a specific kernel restricts it. This is a disadvantage compared to the decision theoretical approach formulated in Section 4. In this section elements $A \in X_c$ are seen as imprecise values of some unknown variable $a \in A$. This automatically restricts the space of meaningful decision functions, since when a certain label is assigned to all points within a set $A \in X_c$, the same label should be assigned to A. This diverging behaviour of the two approaches discussed in this work should be kept in mind when deciding between them. Only when both approaches are meaningful for a given data set, their performance should be compared.

The next section gives some theoretical background of SVMs. Since the input space is not specified there, it is relevant for both approaches.

2. Support Vector Machines for Classification

In this section theoretical aspects of Support Vector Machines (SVMs) as an empirical classification algorithm are going to be discussed. In particular, we show that usually unique solutions exist. Moreover, reproducing kernel Hilbert spaces (RKHS) as special function spaces in which one can look for solutions are introduced. Moreover, we take a look at the issue of numerical optimisation for the hinge loss as an concrete example.

Even though we will not make any strong assumptions on the input space X, like for example $X = \mathbb{R}^d$, we will only discuss empirical SVMs in detail. That means we generally assume a given data set

$$\mathcal{D} = \{(x_i, y_i) | i = 1, \ldots, n\} \subseteq X \times \{-1, +1\}$$

where $(X, \mathcal{A})$ is a measurable space and all products are equipped with corresponding product σ-algebras. At the end of this section we will additionally cover some results on consistency. Here a sequence of data sets with increasing size is assumed.

Even thought Steinwart and Christmann in [16] focus on general (non-empirical) Support Vector Machines, we essentially follow their argumentation. See there for more details and for an additional discussion of Support Vector Regression.

2.1. Empirical Risk Minimisation

A key concept to describe whether a data point in $x \in X$ with label $y \in \{-1, +1\}$ agrees with a statistical model is that of a loss function.

Definition 2.1 (Loss function)
*A **loss function for classification** or simply loss is a measurable function*

$$\mathcal{L} : \mathbb{R} \times \{-1, 1\} \to \mathbb{R}_0^+ \cup \infty.$$

For $(x, y) \in \mathcal{D}$ and a given measurable function $f : X \to \mathbb{R}$ the value $\mathcal{L}(f(x), y)$ is called the loss of predicting y by $f(x)$.

Most loss functions for classification, which are used in practice, penalise data points (x, y) for which the sign of $f(x)$ does not agree with the label y associated with x. Hence $\text{sign}(f(x))$ is used for predictions. We look at loss functions of this kind in the following definition.

Definition 2.2 (Commonly used loss functions)
For $t \in \mathbb{R}$ and $y \in \{-1, +1\}$ define the following loss functions.

- **$0-1$ *loss:***

$$\mathcal{L}_{0-1}(t, y) = \mathbb{1}_{]-\infty,0]}(y \,\text{sign}(t)) = \begin{cases} 0 & \textit{for} \quad \text{sign}(t) = y \\ 1 & \textit{else} \end{cases}$$

- ***Hard margin loss:***

$$\mathcal{L}_{hm}(t, y) = \begin{cases} 0 & \textit{for} \quad yt \geq 1 \\ \infty & \textit{else} \end{cases}$$

- ***Hinge loss:***

$$\mathcal{L}_{hinge}(t, y) = \max\{0, 1 - yt\}$$

- ***Logistic loss:***

$$\mathcal{L}_{logist}(t, y) = \log(1 + \exp(-yt))$$

Definition 2.3 (Convex and monotonic loss)
A loss function $\mathcal{L} : \mathbb{R} \times \{-1, 1\} \to \mathbb{R}_0^+ \cup \infty$ is called

- ***convex** when $\mathcal{L}_y : \mathbb{R} \to \mathbb{R}_0^+ \cup \infty$, $t \mapsto \mathcal{L}(t, y)$ is convex for both $y = +1$ and $y = -1$,*
- ***monotonic** when $\mathcal{L}_y : \mathbb{R} \to \mathbb{R}_0^+ \cup \infty$, $t \mapsto \mathcal{L}(t, y)$ is monotonically increasing for $y = -1$ and monotonically decreasing for $y = +1$.*

Note that all loss functions given in Definition 2.2 are convex and monotonic, except the $0-1$ loss which is monotonic but not convex. Having a convex loss is vital for the uniqueness of the minimiser of the following functional, and to invent numerically stable procedures to find this optimiser.

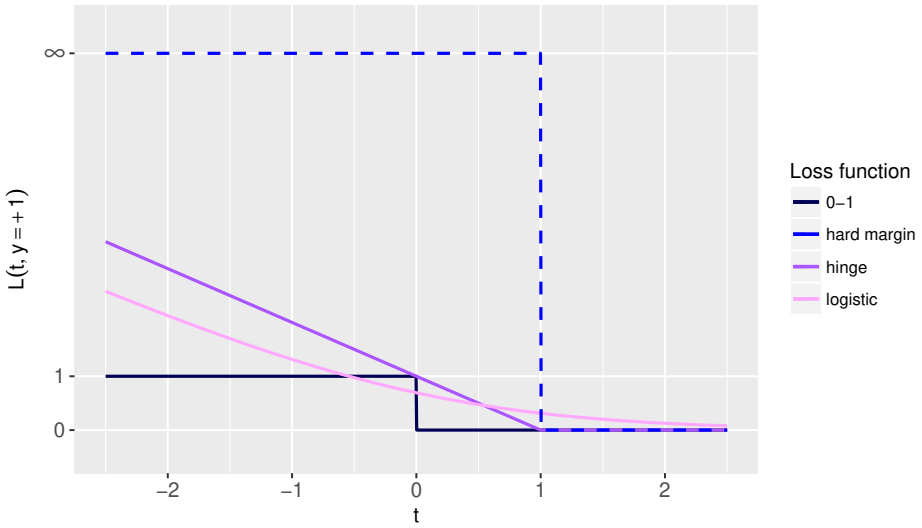


Figure 1: Common loss functions for classification.

Definition 2.4 (Risk)
Let $f : X \to \mathbb{R}$ be measurable, P be a measure on $(X \times \{-1, +1\},\ \mathcal{A} \otimes \mathcal{P}(\{-1, +1\}))$ and $\mathcal{L}$ be a loss function.

1. *The expected loss with respect to P is defined by*

$$\mathcal{R}_P(f) = \mathbb{E}_P[\mathcal{L}(f(x), y)] = \int_{X \times \{-1,+1\}} \mathcal{L}(f(x), y) dP \tag{1}$$

   *and is called the **risk of** f given P.*

2. *For a set of observations $\mathcal{D} = \{(x_i, y_i) | i = 1, \ldots, n\}$ with $(x_i, y_i) \overset{i.i.d.}{\sim} P$ is*

$$\mathcal{R}_{emp}(f) = \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} \mathcal{L}(f(x), y) \tag{2}$$

   *the **empirical risk of** f given $\mathcal{D}$. This is an approximation of the actual (but unknown) risk for a given loss.*

Instead of minimising the empirical risk, an extra term is commonly added to avoid overfitting. This procedure is known as structured risk minimisation. More precisely we try to minimise the following regularised empirical risk functional on some Hilbert space $\mathcal{H}$. Here the regularisation term is chosen to be $\|f\|_{\mathcal{H}}^2$ for $f \in \mathcal{H}$. It could also be replaced by a different continuous and increasing function of $\|f\|_{\mathcal{H}}^2$.

Definition 2.5 (Regularised empirical risk functional)
Let $\mathcal{H} \subseteq \mathbb{R}^X$ be some function space. Let $\lambda > 0$ and for $f \in \mathcal{H}$ let $\mathcal{R}_{emp}(f)$ be the empirical risk of f given $\mathcal{D}$. Then the functional

$$\begin{aligned} \mathcal{R} :&\mathcal{H} \to \mathbb{R} \cup \infty \\ &f \mapsto \lambda \|f\|_{\mathcal{H}}^2 + \mathcal{R}_{emp}(f) \end{aligned} \tag{3}$$

*is called regularised empirical risk functional. Alternatively an **objective function with offset** can be used:*

$$\begin{aligned} \mathcal{R}_{off} :&\mathcal{H} \times \mathbb{R} \to \mathbb{R} \cup \infty \\ &(f, b) \mapsto \lambda \|f\|_{\mathcal{H}}^2 + \mathcal{R}_{emp}(f + b). \end{aligned} \tag{4}$$

Subsequently we will focus on the risk without offset but corresponding results for the risk with offset are usually mentioned. One reason for preferring $\mathcal{R}$ over $\mathcal{R}_{off}$ is the next theorem, which states that unique minimiser for $\mathcal{R}$ exists under weak assumptions.

Theorem 2.6 (Existence of unique minimisers)
Let $\mathcal{L}$ be a finite and convex loss. Let $\mathcal{H} \subseteq \mathbb{R}^X$ be a Hilbert space such that the linear maps $\delta_x : \mathcal{H} \to \mathbb{R}, f \mapsto f(x)$ are continuous for all $x \in X$.
Then $\mathcal{R}$ (defined in Equation 3) has a unique minimiser.

Proof. Since $\mathbb{R} \to \mathbb{R}_0^+, t \mapsto \mathcal{L}(t, y_i)$ is convex for all $i = 1, \dots n$ these are continuous as a real functions (see Lemma A.7). Therefore $\mathcal{H} \to \mathbb{R}_0^+, f \mapsto \mathcal{L}(f(x_i), y_i)$ is continuous as a composition of continuous functions and consequently one concludes that $\mathcal{R}$ is continuous as a linear combination of continuous functions.

Moreover, we have $\mathcal{H} \to \mathbb{R}_0^+, f \mapsto \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x_i), y_i)$ being convex as a sum and positive multiple of convex functions. Hence $\mathcal{R}$ is strictly convex as a sum of a convex and a strictly convex function (the squared norm of f).

Furthermore, for every sequence $(f_n)_{n\in\mathbb{N}} \subset \mathcal{H}$ with $\|f\|_{\mathcal{H}} \to \infty$ we have

$$\mathcal{R}(f) \geq \|f\|_{\mathcal{H}}^2 \to \infty$$

which implies that $\mathcal{R}$ is coercive. Therefore we conclude, by applying Theorem A.10, that there exists a minimal solution f^* of $\mathcal{R}$ in the Hilbert space $\mathcal{H}$, which is in particular a reflexive Banach space. The uniqueness of f^* is a result of $\mathcal{R}$ being strictly convex. □

Remark 2.7
Analogously to Theorem 2.6 it can be shown that $\mathcal{R}_{off}$ (defined in Equation 4) has a minimiser. However one needs further assumptions to ensure uniqueness in that case.

Hilbert spaces that fulfil the second requirement of Theorem 2.6 will play an important role in subsection 2.3. The are called reproducing kernel Hilbert spaces. The naming can be understood when considering the proof of Theorem 2.19 in Subsection 2.3 and the subsequent comments.

Definition 2.8 (Reproducing kernel Hilbert space)
*Let $\mathcal{H} \subseteq \mathbb{R}^X$ be a Hilbert space. $\mathcal{H}$ is called a **reproducing kernel Hilbert space** (RKHS) on X if*

$$\begin{aligned} \delta_x : \mathcal{H} &\to \mathbb{R} \\ f &\mapsto f(x) \end{aligned}$$

is continuous for all $x \in X$.

Corollary 2.9
Let X be a Hilbert space, for example $X = \mathbb{R}^n$, and $\mathcal{H} = X'$ is its dual space, that is the vector space of all continuous linear functionals on X. Then we have

$$|f(x)| \leq \|f\|_{\mathcal{H}} \|x\|_X \quad \forall x \in X,\ f \in \mathcal{H}$$

by applying the Cauchy-Schwarz Inequality (Lemma A.2). Thus $\mathcal{H} \to \mathbb{R}, f \mapsto f(x)$ is continuous for all $x \in X$.

Hence the second condition of Theorem 2.6 is fulfilled when considering linear functionals on a Hilbert space. Therefore, its dual is a reproducing kernel Hilbert space and we can conclude existence and uniqueness of solutions.

2.2. Geometrical Interpretation

Consider the special case where $X = \mathbb{R}^d$ and $\mathcal{D}$ is linearly separable. That is, there exists a linear hyperplane in $\mathbb{R}^d$ such that all $x \in X$ with label $y = +1$ lie on one side of the plane and those with label $y = -1$ on the other. Furthermore, let $\mathcal{H} = (\mathbb{R}^d)'$, the dual of $\mathbb{R}^d$. Hence all $f \in \mathcal{H}$ can be written as $f_w = \langle w, \cdot \rangle$ for some $w \in \mathbb{R}^d$ (Riesz Representation Theorem [14, page 118]). Additionally assume a hard margin loss.

Example 2.10 (Linear separation without offset)
Considering a hard margin loss,

$$\mathcal{L}(y, \langle w, x \rangle) = \begin{cases} 0, & \text{for } y\langle w, x \rangle \geq 1 \\ \infty, & \text{else} \end{cases}$$

for all $(x, y) \in \mathcal{D}$, implies that for all $w \in \mathbb{R}^d$ such that $\mathcal{R}(f_w)$ is finite, all points with different labels are separated by the hyperplane $\{x \in \mathbb{R}^d | \langle w, x \rangle = 0\}$. Furthermore there are no points in between the two hyperplanes $\{x \in \mathbb{R}^d | \langle w, x \rangle = -1\}$ and $\{x \in \mathbb{R}^d | \langle w, x \rangle = +1\}$. The distance between those hyperplanes equals $\frac{2}{\|w\|_2}$. Hence minimising the risk $\mathcal{R}(f_w) = \lambda \|w\|_2^2$ is equivalent to maximising the distance between the separating hyperplanes. This distance is called the margin.

Example 2.11 (Linear separation with offset)
Similarly the loss function

$$\mathcal{L}(y, \langle w, x \rangle + b) = \begin{cases} 0, & \text{for } y(\langle w, x \rangle + b) \geq 1 \\ \infty, & \text{else} \end{cases} \qquad \forall (x, y) \in \mathcal{D}.$$

ensures that points with different labels are separated by all hyperplanes described by $\{x \in \mathbb{R}^d | \langle w, x \rangle + b = c\}$ $\forall c \in [-1, 1]$. Since the distance between the two hyperplanes

$\{x \in \mathbb{R}^d | \langle w, x \rangle + b = -1\}$ *and* $\{x \in \mathbb{R}^d | \langle w, x \rangle + b = +1\}$ *does not depend on the offset* b*, it can likewise be computed as* $\frac{2}{\|w\|_2}$.

Minimising the risk $\mathcal{R}_{off}(f, b) = \lambda \|w\|_2^2$ *is therefore still equivalent to maximising the distance between the separating hyperplanes. In this particular example the variable* b *can actually be interpreted as an geometrical offset of the hyperplane* $\{x \in \mathbb{R}^d | \langle w, x \rangle + b = 0\}$*, which is the distance to the origin.*

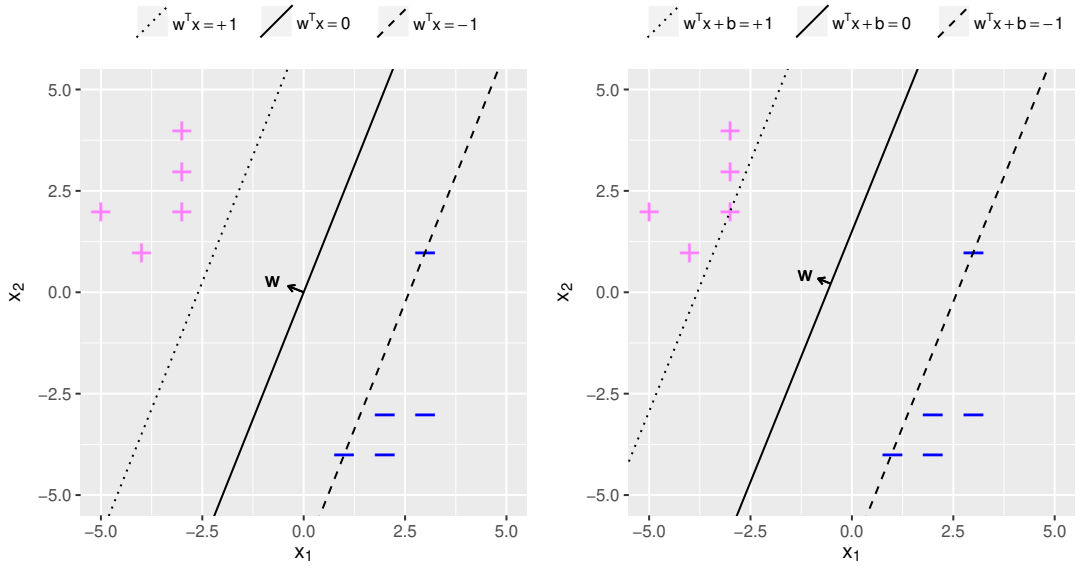


Figure 2: Linear separation in $\mathbb{R}^2$ without offset (left) and with offset (right).

Example 2.10 and Example 2.11 not only demonstrate the use of an offset, when the special case of linear separation is considered, they also illustrate where Support Vector Machines got their name from. The data points that lie on either

$$\{x \in \mathbb{R}^n | \langle w, x \rangle + b = -1\} \text{ or } \{x \in \mathbb{R}^n | \langle w, x \rangle + b = +1\}$$

are called Support Vectors. Only these determine the position of the separating hyperplane.

2.3. The Kernel Trick

Unlike in Subsection 2.2, linear separation might not be sufficient for separation of arbitrary datasets. On the other hand it is not clear in which Hilbert space $\mathcal{H}$ one should look for separating functions instead. Even if we had a suitable space of functions, optimisation of the regularised empirical risk functional might be difficult.

A common workaround for this problem is to define a mapping $\phi : X \to \mathcal{F}$ where $\mathcal{F}$ is some Hilbert space and find a separating hyperplane there. Hence the task becomes to find an element of $\mathcal{F}'$ that minimises the regularised empirical risk for data

$$\phi(\mathcal{D}) := \{(\phi(x_i), y_i) | i = 1, \dots, n\}.$$

This procedure effectively leads to a non-linear separating functional on X.

Definition 2.12 (Feature map)
Let $\phi : X \to \mathcal{F}$, with $\mathcal{F}$ being a Hilbert space.
Then ϕ is called a ***feature map*** *for the* ***feature space*** *$\mathcal{F}$.*

Remark 2.13
Corollary 2.9 shows that there exists an unique linear functional $f^* \in \mathcal{F}'$ that minimises the risk for data $\phi(\mathcal{D}) \subset \mathcal{F} \times \{-1, +1\}$. Moreover, the next lemma shows that this minimiser admits a certain representation.

Lemma 2.14
Let $\mathcal{D} = \{(x_i, y_i) | i = 1, \dots, n\} \subset X \times \{-1, +1\}$ and $\phi : X \to \mathcal{F}$ be a feature map with corresponding feature space $\mathcal{F}$. Let $\mathcal{F}'$ be the dual of the Hilbert space $\mathcal{F}$ and $f^* \in \mathcal{F}'$ be a minimiser of

$$\begin{aligned} \mathcal{R} : \mathcal{F}' &\to \mathbb{R} \\ f &\mapsto \lambda \|f\|_{\mathcal{F}'}^2 + \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(\phi(x_i)), y_i). \end{aligned} \tag{5}$$

Then f^* admits a representation

$$f^* = \langle \sum_{i=1}^{n} \alpha_i \phi(x_i), \cdot \rangle_{\mathcal{F}} \text{ with } \alpha_i \in \mathbb{R} \ \forall i = 1, \dots, n.$$

Proof. Let f^* be any minimiser of Equation 5. Let $\mathcal{H} = \text{span}(\{\phi(x_1), \dots \phi(x_n)\})$ and decompose $\mathcal{F}$ as a direct sum of $\mathcal{H}$ (which is closed, since it is the span of a finite set) and its orthogonal complement (see [14, page 100]):

$$\mathcal{F} = \mathcal{H} \oplus \mathcal{H}^{\perp}.$$

Then f^* can be written as

$$f^* = \langle h_1 + h_2, \cdot \rangle \quad \text{with } h_1 \in \mathcal{H} \text{ and } h_2 \in \mathcal{H}^{\perp}$$

using the Riesz Representation Theorem A.3. Let $(x, y) \in \mathcal{D}$ be an arbitrary point of the training data set. Applying f^* to $\phi(x)$ gives

$$f^*(\phi(x)) = \langle h_1, \phi(x) \rangle + \langle h_2, \phi(x) \rangle = \langle h_1, \phi(x) \rangle \quad \forall x \in X$$

and therefore

$$\mathcal{R}_{emp}(f^*) = \mathcal{R}_{emp}(\langle h_1, \cdot \rangle) \quad \text{on data } \phi(\mathcal{D}).$$

Applying the Pythagorean Theorem yields

$$\|f^*\|_{\mathcal{F}'}^2 = \|h_1 + h_2\|_{\mathcal{F}}^2 = \|h_1\|_{\mathcal{F}}^2 + \|h_2\|_{\mathcal{F}}^2 \geq \|h_1\|_{\mathcal{F}}^2 = \|\langle h_1, \cdot \rangle\|_{\mathcal{F}'}^2.$$

Since f^* is a minimiser of Equation 5 we get

$$\mathcal{R}(f^*) \leq \mathcal{R}(\langle h_1, \cdot \rangle) = \lambda \|\langle h_1, \cdot \rangle\|_{\mathcal{F}'}^2 + \mathcal{R}_{emp}(f^*) \leq \lambda \|f^*\|_{\mathcal{F}'}^2 + \mathcal{R}_{emp}(f^*) = \mathcal{R}(f^*).$$

Hence we have $\|f^*\|_{\mathcal{F}'} = \|\langle h_1, \cdot \rangle\|_{\mathcal{F}'}$. Applying the Pythagorean Theorem once more one obtains

$$\|\langle h_2, \cdot \rangle\|_{\mathcal{F}'}^2 = \|h_2\|_{\mathcal{F}}^2 = \|h_2\|_{\mathcal{F}}^2 + \|h_1\|_{\mathcal{F}}^2 - \|h_1\|_{\mathcal{F}}^2 = \|f^*\|_{\mathcal{F}'}^2 - \|\langle h_1, \cdot \rangle\|_{\mathcal{F}'}^2 = 0$$

which implies $h_2 = 0$ and consequently $f^* = \langle h_1, \cdot \rangle$. Since $h_1 \in \text{span}(\{\phi(x_1), \dots \phi(x_n)\})$, the desired representation follows immediately. □

Remark 2.15
Lemma 2.14 still holds true when a risk function with offset is used.

When we substitute the representation of f^* obtained in Lemma 2.14 back into Equation 5 we see that the problem is actually equivalent to minimising the objective function

$$\begin{aligned}&\lambda\|\sum_{i=1}^{n}\alpha_i\phi(x_i)\|_{\mathcal{F}}^2+\frac{1}{n}\sum_{j=1}^{n}\mathcal{L}\left(\langle\sum_{i=1}^{n}\alpha_i\phi(x_i),\phi(x_j)\rangle_{\mathcal{F}},y_j\right)\\=&\lambda\sum_{j=1}^{n}\sum_{i=1}^{n}\alpha_j\alpha_i\langle\phi(x_j),\phi(x_i)\rangle_{\mathcal{F}}+\frac{1}{n}\sum_{j=1}^{n}\mathcal{L}\left(\sum_{i=1}^{n}\alpha_i\langle\phi(x_i),\phi(x_j)\rangle_{\mathcal{F}},y_j\right)\end{aligned}$$

with respect to $\alpha=(\alpha_1,\ldots,\alpha_n)\in\mathbb{R}^n$.
One observes that the minimisation problem depends on $\{x_i,\mid i=1,\ldots,n\}$ and ϕ only via $\langle\phi(x_j),\phi(x_i)\rangle$. Hence, it might be sufficient to define a mapping (named a kernel) $k:X\times X\to\mathbb{R}$ instead of a feature map ϕ and a feature space $\mathcal{F}$.

Definition 2.16 (Kernel)
A function $k:X\times X\to\mathbb{R}$ is called a kernel on X if there exists a Hilbertspace $\mathcal{F}$ and a map $\phi:X\to\mathcal{F}$ such that

$$k(x_1,x_2)=\langle\phi(x_1),\phi(x_2)\rangle_{\mathcal{F}}\quad\forall x_1,x_2\in X. \tag{6}$$

This definition directly yields the following implications.

Lemma 2.17 (Basic properties of kernels)
Let $k:X\times X\to\mathbb{R}$ be a kernel. Then

1. *the kernel k is **symmetric**, that is $k(x_1,x_2)=k(x_2,x_1)\quad\forall x_1,x_2\in X$,*

2. *the kernel k is **positive semi-definite**, that is for all $n\in\mathbb{N}$, for all $\alpha_1,\ldots,\alpha_n\in\mathbb{R}$ and for all $x_1,\ldots,x_n\in X$ holds*

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(x_i,x_j)\geq 0. \tag{7}$$

3. We have the ***Cauchy-Schwarz Inequality****, that is*

$$k(x_1, x_2)^2 \leq k(x_1, x_1)k(x_2, x_2) \quad \forall x_1, x_2 \in X \tag{8}$$

Proof. The kernel k is clearly symmetric, since the scalar product on $\mathcal{F}$ is symmetric. Furthermore we compute

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) &= \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{F}} \\
&= \langle \sum_{i=1}^{n} \alpha_i \phi(x_i), \sum_{j=1}^{n} \alpha_j \phi(x_j) \rangle_{\mathcal{F}} \\
&= \| \sum_{i=1}^{n} \alpha_i \phi(x_i) \|_{\mathcal{F}}^2 \\
&\geq 0.
\end{aligned}$$

Hence k is positive semi-definite. This also implies the Cauchy-Schwarz Inequality (Lemma A.2) □

Remark 2.18

Note that some authors (for example Steinwart and Christmann [16]) call a mapping ***positive definite*** *when Inequality 7 holds true. We call this property* ***positive semi-definite****, since this definition coincides with the usual definition for matrices. More precisely a mapping $k : X \times X$ is positive semi-definite (according to the definition given here) if and only if all its Gram matrices*

$$K := (k(x_i, x_j))_{\substack{i=1,\dots,n \\ j=1,\dots,n}} \qquad \text{with } n \in \mathbb{N},\ x_1, \dots x_n \in X \tag{9}$$

are positive semi-definite.

Having defined kernels indirectly through a feature space and a feature map it seems natural to ask which mappings $k : X \times X \to \mathbb{R}$ actually define kernels on X. The next theorem can answer the question if a given mapping $k : X \times X \to \mathbb{R}$ is a kernel. That is whether there exists a feature map $\phi : X \to \mathcal{F}$ and Hilbert space $\mathcal{F}$ such that $k = \langle \phi, \phi \rangle_{\mathcal{F}}$.

Theorem 2.19 (Characterisation of kernel functions)
Let $k : X \times X \to \mathbb{R}$ be a symmetric function. Then k is a kernel if and only if it is positive semi-definite.

Proof. This proof is essentially the same as in [13, page 418]. However it introduces the concept of reproducing kernel Hilbert spaces, therefore we repeat it here. By Lemma 2.17 we already now that every kernel is positive semi-definite. To show the other direction let

$$\mathbb{V} := \operatorname{span}\left(\{k(x, \cdot) | x \in X\}\right).$$

and

$$\langle f, g\rangle_{\mathbb{V}} := \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \beta_j k(x_{1i}, x_{2j})$$

for $f = \sum_{i=1}^{n} \alpha_i k(x_{1i}, \cdot)$ and $g = \sum_{i=1}^{m} \beta_i k(x_{2i}, \cdot)$. Then $(\mathbb{V}, \langle\cdot,\cdot\rangle_{\mathbb{V}})$ defines an inner product space, since

- the mapping $\langle\cdot,\cdot\rangle_{\mathbb{V}}$ is **well defined**, as

  $$\langle f, g\rangle_{\mathbb{V}} = \sum_{j=1}^{m} \beta_j f(x_{2j}) = \sum_{i=1}^{n} \alpha_i g(x_{1i})$$

  is independent of concrete representations of f and g.

- It is clearly **linear** and **symmetric** by definition and

- **positive definite** since

  $$\langle f, f\rangle_{\mathbb{V}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(x_{1i}, x_{1j}) \geq 0 \quad \forall f \in \mathbb{V} \tag{10}$$

  because k is positive semi-definite. To show $\langle f, f\rangle_{\mathbb{V}} = 0$ implies $f = 0$ we first notice that $\langle\cdot,\cdot\rangle_{\mathbb{V}}$ is itself a positive semi-definite bilinear form on $\mathbb{V}$.

To see this let $f_1, \ldots, f_m \in \mathbb{V}$ and $\lambda_1, \ldots \lambda_m \in \mathbb{R}$. Then we have

$$\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j\langle f_i, f_j\rangle_{\mathbb{V}} = \langle\sum_{i=1}^{m}\lambda_i f_i, \sum_{j=1}^{m}\lambda_j f_j\rangle_{\mathbb{V}} \geq 0,$$

since $\sum_{i=1}^{m}\lambda_i f_i \in \mathbb{V}$ and Equation 10. Hence the Cauchy-Schwarz Inequality holds for $\langle\cdot,\cdot\rangle_{\mathbb{V}}$ (see Lemma A.2), hence we have

$$\langle f_1, f_2\rangle_{\mathbb{V}}^2 \leq \langle f_1, f_1\rangle_{\mathbb{V}}\langle f_2, f_2\rangle_{\mathbb{V}} \quad \forall f_1, f_2 \in \mathbb{V}.$$

Now let $f = \sum_{i=1}^{n}\alpha_i k(x_i, \cdot) \in \mathbb{V}$ be such that $\langle f, f\rangle_{\mathbb{V}} = 0$. Then we have for all $x \in X$:

$$f(x)^2 = \left(\sum_{i=1}^{n}\alpha_i k(x_i, x)\right)^2 = \langle f, k(x,\cdot)\rangle_{\mathbb{V}}^2 \leq \langle f, f\rangle_{\mathbb{V}}\langle k(x,\cdot), k(x,\cdot)\rangle_{\mathbb{V}} = 0. \quad (11)$$

This gives $f(x) = 0$ for all $x = X$ which is the same as $f = 0$.

Define $\mathcal{F}$ to be the completion of $\mathbb{V}$ with respect to the norm induced by $\langle\cdot,\cdot\rangle_{\mathbb{V}}$ (see [17, page 64]). Then $\mathcal{F}$ is a Hilbert space with inner product $\langle\cdot,\cdot,\rangle_{\mathcal{F}}$ such that

$$\langle f, g\rangle_{\mathcal{F}} = \langle f, g\rangle_{\mathbb{V}} \quad \forall f, g \in \mathbb{V}.$$

This shows that k is actually a kernel on $X \times X$ since we have for $x_1, x_2 \in X$

$$k(x_1, x_2) = \langle k(x_1,\cdot), k(x_2,\cdot)\rangle_{\mathcal{F}}.$$

Hence $\phi : X \to \mathcal{F}$, $x \to k(x,\cdot)$ defines a feature map on X with feature space $\mathcal{F}$. $\square$

The feature space $\mathcal{F}$ constructed in the proof will play an important role when considering Support Vector Machines based on a specific kernel function. Note that, Equation 11 implies that

$$|f(x)| \leq \|f\|_{\mathcal{F}}\sqrt{k(x,x)} \quad \text{for all } x \in X, f \in \mathcal{F}.$$

Hence

$$\begin{aligned} \delta_x : \mathcal{H} &\to \mathbb{R} \\ f &\mapsto f(x) \end{aligned}$$

is continuous for all $x \in X$, which means $\mathcal{F}$ is a reproducing kernel Hilbert space. Hence for any given kernel k we can construct a feature space that is a reproducing kernel Hilbert space.

Definition 2.20
Let $k : X \times X \to \mathbb{R}$ be a kernel. Then we denote by $\mathcal{H}_k$ the feature space constructed in the proof of Theorem 2.19. $\mathcal{H}_k$ is called the reproducing kernel Hilbert space associated with k.

Corollary 2.21 (Reproducing kernel)
Let k be a kernel on X and $\mathcal{H}_k$ be the reproducing kernel Hilbert space associated with it. Then we have

$$f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}_k} \quad \forall x \in X, \ f \in \mathcal{H}_k.$$

*Conversely if this property holds for any kernel function $k : X \times X \to \mathbb{R}$ with $k(x, \cdot) \in \mathcal{H}$ for all $x \in X$ and some Hilbert space $\mathcal{H}$ we call k a **reproducing kernel** of $\mathcal{H}$.*

Remark 2.22
There is a one-to-one correspondence between reproducing kernel Hilbert spaces and reproducing kernels. Many properties of the functions in the RKHS can equivalently been seen as properties of the kernel. For more details see Chapter 4 in [16], in particular Theorem 4.20 and Theorem 4.21.

Lemma 2.23
For any kernel $k : X \times X \to \mathbb{R}$ is $\mathcal{H}_k$ a reproducing kernel Hilbert space. Moreover if X is a metric space and k is continuous on $X \times X$ we have

$$\mathcal{H}_k \subset C(X),$$

where $C(X)$ denotes the space of all continuous functions on X. That is all functions in $\mathcal{H}_k$ are continuous.

Proof. We have implicitly shown

$$|\delta_x(f)| = |f(x)| \leq \|f\|_{\mathcal{H}_k}\|k(x,\cdot)\|_{\mathcal{H}_k} \quad \forall x \in X,\ f \in \mathcal{H}_k.$$

That is δ_x being continuous for all $x \in X$. Hence $\mathcal{H}_k$ is a reproducing kernel Hilbert space. Moreover for all $x_1, x_2 \in X$ and all $f \in \mathcal{H}_k$ we obtain

$$\begin{aligned}
|f(x_1) - f(x_2)| &= |\langle f, k(x_1,\cdot)\rangle_{\mathcal{H}_k} - \langle f, k(x_2,\cdot)\rangle_{\mathcal{H}_k}| \\
&= |\langle f, k(x_1,\cdot) - k(x_2,\cdot)\rangle_{\mathcal{H}_k}| \\
&\leq \|f\|_{\mathcal{H}_k}\|k(x_1,\cdot) - k(x_2,\cdot)\|_{\mathcal{H}_k} \\
&= \|f\|_{\mathcal{H}_k}\sqrt{k(x_1,x_1) + k(x_2,x_2) - 2k(x_1,x_2)}
\end{aligned}$$

Hence if k is continuous, so is f. □

Since $\mathcal{H}_k$ is a RKHS we can deduce that, given a finite and convex loss function $\mathcal{L}$, a unique solution to the optimisation problem given any valid kernel function can be found. More precisely we have

Corollary 2.24 (Representer Theorem)
Let $\mathcal{L}$ be a finite and convex loss and k be a kernel on X. Furthermore let $\mathcal{H}_k$ be the reproducing kernel Hilbert space associated with k. Then there exists a unique minimiser $f^* \in \mathcal{H}_k$ of

$$\begin{aligned}
\mathcal{R} : \mathcal{H}_k &\to \mathbb{R} \\
f &\mapsto \lambda\|f\|^2_{\mathcal{H}_k} + \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f(x_i), y_i).
\end{aligned} \tag{12}$$

Moreover, f^* admits a representation of the form

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

for some $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$.

This and similar statements are know as the "Representer Theorem" for example in [16, page 168] and in [13]. They are an immediate consequence of Lemma 2.14 and the reproducing property of kernels. Since there is a one-to-one correspondence between reproducing kernels and reproducing kernel Hilbert spaces one can equivalently replace the assumption of a kernel by requiring a RKHS in Corollary 2.24.

As a consequence it is sufficient to consider a kernel function instead of an explicit function space $\mathcal{H}$. The Hilbert space $\mathcal{H}$ is then implicitly assumed to be the reproducing kernel Hilbert space associated with k. Hence it is vital to have suitable kernel functions on hand. The following lemmata will show how to construct new kernel functions and give examples of commonly used ones.

Lemma 2.25 (Constructing kernel functions)
Let $k_i : X \times X \to \mathbb{R}$, $i \in \mathbb{N}$ be kernel functions and $\lambda \geq 0$.

1. *If $f : X \to \mathbb{R}$ is some function on X then*

$$\begin{aligned} X \times X &\to \mathbb{R} \\ (x_1, x_2)^T &\mapsto f(x_1) k_1(x_1, x_2) f(x_2) \end{aligned}$$

   is a kernel on $X \times X$.

2. *The mappings $k_1 + \lambda k_2$, and $k_1 \cdot k_2$ are kernels on $X \times X$.*

3. *If $\lim_{i \to \infty} k_i(x_1, x_2) =: k(x_1, x_2)$ exists for all $x_1, x_2 \in X$ then the limit*

$$\begin{aligned} k : X \times X &\to \mathbb{R} \\ (x_1, x_2)^T &\mapsto k(x_1, x_2) \end{aligned}$$

   is a kernel on $X \times X$.

Proof.

1. Since k_1 is a kernel on $X \times X$ there exists a feature space $\mathcal{F}$ and a feature map $\phi : X \to \mathcal{F}$ such that

$$k_1(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{F}} \quad \forall x_1, x_2 \in X.$$

   Hence

$$f(x_1)k_1(x_1, x_2)f(x_2) = f(x_1)\langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{F}} f(x_2) = \langle f(x_1)\phi(x_1), f(x_2)\phi(x_2) \rangle_{\mathcal{F}}$$

   for all $x_1, x_2 \in X$ which shows that $f \cdot \phi$ is a feature map for feature space $\mathcal{F}$.

2. The mapping $k_1 + \lambda k_2$ is clearly symmetric. Moreover, for all $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and for all $x_1, \ldots, x_n \in X$ holds

$$\begin{aligned} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j (k_1 + \lambda k_2)(x_i, x_j) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k_1(x_i, x_j) + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k_2(x_i, x_j) \\ &\geq 0. \end{aligned}$$

   This means $k_1 + \lambda k_2$ is positive semi-definite and by Theorem 2.19 a kernel. To show that the product of two kernels is a kernel let $\mathcal{F}_1$, $\mathcal{F}_2$ be feature spaces with feature maps ϕ_1, ϕ_2 for kernels k_1 and k_2 respectively. Let $\mathcal{F} = \mathcal{F}_1 \otimes \mathcal{F}_2$ be the tensor product Hilbert space of $\mathcal{F}_1$ and $\mathcal{F}_2$ with corresponding scalar product

$$\langle v_1 \otimes w_1, v_2 \otimes w_2 \rangle_{\mathcal{F}} = \langle v_1, v_2 \rangle_{\mathcal{F}_1} \langle w_1, w_2 \rangle_{\mathcal{F}_2} \quad \forall v_1, v_2 \in \mathcal{F}_1, w_1, w_2 \in \mathcal{F}_2.$$

   Hence $\phi : X \to \mathbb{R}$, $x \mapsto \phi_1(x) \otimes \phi_2(x)$ defines a feature map for feature space $\mathcal{F}$ since we have

$$\begin{aligned} \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{F}} &= \langle \phi_1(x_1) \otimes \phi_2(x_1), \phi_1(x_2) \otimes \phi_2(x_2) \rangle_{\mathcal{F}} \\ &= \langle \phi_1(x_1), \phi_1(x_2) \rangle_{\mathcal{F}_1} \langle \phi_2(x_1), \phi_2(x_2) \rangle_{\mathcal{F}_2} \\ &= k_1(\phi_1(x_1), \phi_1(x_2)) k_2(\phi_2(x_1), \phi_2(x_2)) \end{aligned}$$

   for all $x_1, x_2 \in X$. This shows that $k_1 \cdot k_2$ is a kernel on $X \times X$.

3. k is clearly symmetric and for all $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and for all $x_1, \ldots, x_n \in X$ holds

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) = \lim_{i\to\infty} \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k_i(x_i, x_j) \geq 0.$$

Hence k is positive semi-definite as well and therefore a kernel on $X \times X$.

□

Lemma 2.26 (Common kernel functions on $\mathbb{R}^d$)
Let $X = \mathbb{R}^d$ and $\langle \cdot, \cdot \rangle$ the usual inner product (dot product) on $\mathbb{R}^d$. Then the following functions are valid kernels on $X \times X$:

1. ***Linear kernel:*** $k(x_1, x_2) = \langle x_1, x_2 \rangle \quad \forall x_1, x_2 \in \mathbb{R}^d$
2. ***Polynomial kernel:*** $k(x_1, x_2) = (\langle x_1, x_2 \rangle + c)^m \quad \forall x_1, x_2 \in \mathbb{R}^d,\ c > 0,\ m \in \mathbb{N}$
3. ***Gaussian kernel:*** $k(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|_2^2) \quad \forall x_1, x_2 \in \mathbb{R}^d,\ \gamma > 0$
   Here $\|\cdot\|_2$ denotes the norm induced by the dot product.

Proof.

1. The usual scalar product on $\mathbb{R}^d$ is a kernel by construction.
2. Notice that every non-negative constant $c \in \mathbb{R}$ is a kernel since $\phi : X \to \mathbb{R}$, $x \mapsto \sqrt{c}$ is a feature map for feature space $\mathbb{R}$. The desired result can therefore be obtained by applying the second part of Lemma 2.25 and using the first part of this lemma.
3. Let $k^*(x_1, x_2) = \exp(2\gamma \langle x_1, x_2 \rangle) = \sum_{i=0}^{\infty} \frac{(2\gamma \langle x_1, x_2 \rangle)^i}{i!}$ for all $x_1, x_2 \in X$. Then k^* defines a kernel on $X \times X$ (called the exponential kernel) since it is the limit of positive linear combination of kernels (Lemma 2.25). Moreover we can write k as

$$k(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|_2^2) = \exp(-\gamma \|x_1\|^2) k^*(x_1, x_2) \exp(-\gamma \|x_2\|^2)$$

for all $x_1, x_2 \in X$, which shows that k defines a kernel on $X \times X$ by the first part of of the foregoing lemma.

□

Note that if k is a linear kernel, its reproducing kernel Hilbert space is precisely $\mathbb{R}^d$. Hence the risk for $f_w \in (\mathbb{R}^d)'$ with $f = \langle w, \cdot \rangle$ for some $w \in \mathbb{R}^d$ is the same as if we define $\mathcal{H}$ in Equation 3 to be $(\mathbb{R}^d)'$.

2.4. Optimisation Procedure for the Hinge Loss

We have seen that for a given kernel the optimisation problem becomes a convex problem in $\mathbb{R}^n$. Nevertheless, the objective function is not necessarily differentiable. This is even the case for one of the most commonly used loss function, the hinge loss. We will now discuss how the optimisation problem for the hinge loss without offset can be reformulated in a way that allows efficient numerical optimisation. This procedure can be adapted for other risks, for example for the hard margin loss or an additional offset.

In this subsection let $\mathcal{L}$ be the hinge loss (see Definition 2.2) and $k : X \times X \to \mathbb{R}$ be an arbitrary kernel on X. Denote by

$$K = (k(x_i, x_j))_{\substack{i=1,\dots,n \\ j=1,\dots,n}} = \begin{pmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{pmatrix}$$

the symmetric **Gram matrix** for data $\mathcal{D} = \{(x_i, y_i) | i = 1, \dots, n\}$. Thus the risk function without offset can be written as

$$\begin{aligned} &\mathcal{R} : \mathbb{R}^n \to \mathbb{R} \\ &\qquad \alpha \mapsto \lambda \alpha^T K \alpha + \frac{1}{n} \sum_{i=1}^{n} \max\{0, 1 - y_i \sum_{j=1}^{n} \alpha_j K_{ij}\}. \end{aligned} \tag{13}$$

Hence minimizing $\mathcal{R}$ with respect to $\alpha = (\alpha_1, \dots, \alpha_n)$ is equivalent to the following optimisation problem.

Definition 2.27 (Primal Optimisation Problem)
This convex optimisation problem is called the ***primal problem for the hinge loss****.*

$$\begin{aligned} &\textit{minimize} \quad \lambda\alpha^T K\alpha + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ &\textit{subject to} \quad \left.\begin{array}{rl} 1 - y_i\alpha^T K_i - \xi_i & \leq 0 \\ \xi_i & \leq 0 \end{array}\right\} \forall i = 1,\ldots,n \\ &\textit{with respect to} \quad \alpha, \xi \in \mathbb{R}^n, \end{aligned}$$

where $K_i^T = (k(x_i, x_j))_{j=1,\ldots,n}$ *for all* $i = 1,\ldots,n$.

Here so called slack variables $\xi_i \in \mathbb{R}_0^+$, $i = 1,\ldots,n$ were introduced. Solving the primal problem directly is still is numerically expensive thought. Thus a dual formulation is used instead. More precisely we have

Lemma 2.28
Let $\mu^* \in \mathbb{R}^n$ *be a solution to the following optimisation problem.*

$$\begin{aligned} &\textit{maximize} \quad \sum_{i=1}^{n}\mu - \frac{1}{4\lambda}\sum_{i=1}^{n}\sum_{j=1}^{n}\mu_i\mu_j y_i y_j K_{ij} \\ &\textit{subject to} \quad 0 \leq \mu_i \leq \frac{1}{n} \quad \forall i = 1,\ldots,n \\ &\textit{with respect to} \quad \mu \in \mathbb{R}^n. \end{aligned}$$

Then $\alpha^* = (\alpha_i^*)_{i=1,\ldots,n}$ *with* $\alpha_i^* = \frac{y_i\mu_i^*}{2\lambda}$ *for all* $i = 1,\ldots,n$ *minimises the risk given in Equation 13.*

Proof. Since both, the objective function and the constraints, are convex the duality gap vanishes. Hence the primal problem (Definition 2.27) is equivalent to the Lagrangian dual optimisation problem (see [2, page 267]) which can be formulated as

$$\begin{aligned} &\text{maximize} \quad \Phi(\mu, \nu) \\ &\text{subject to} \quad \mu_i, \nu_i \geq 0 \quad \forall i = 1,\ldots,n \\ &\text{with respect to} \quad \mu, \nu \in \mathbb{R}^n. \end{aligned}$$

with corresponding objective function

$$\Phi(\mu,\nu) = \inf_{\alpha,\xi\in\mathbb{R}^n} \Lambda(\alpha,\xi,\mu,\nu).$$

and Lagrangian

$$\Lambda(\alpha,\xi,\mu,\nu) = \lambda\alpha^T K\alpha + \frac{1}{n}\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\mu_i(1-\xi_i - y_i\alpha^T K_i) - \sum_{i=1}^{n}\nu_i\xi_i.$$

Necessary requirements to a solution of both the primal and the dual problem are given by the Karush-Kuhn-Tucker conditions (Theorem A.11). Applied to the primal problem (Definition 2.27) they become

$$\begin{aligned} I.1: \quad & \nabla_\alpha(\lambda\alpha^T K\alpha + \textstyle\sum_{i=1}^n \xi_i) + \sum_{i=1}^n \mu_i \nabla_\alpha(1 - y_i\alpha^T K_i - \xi_i) &&= 0 \\ I.2: \quad & \nabla_\xi(\textstyle\sum_{i=1}^n \xi_i) + \sum_{i=1}^n \mu_i \nabla_\xi(1 - y_i\alpha^T K_i - \xi_i) - \sum_{i=1}^n \nu_i \nabla_\xi \xi_i &&= 0 \end{aligned}$$

and for all $i = 1,\ldots,n$:

$$\begin{aligned} II.1: \quad & \mu_i(1 - y_i\alpha^T K_i - \xi_i) &&= 0 \\ II.2: \quad & \nu_i\xi_i &&= 0 \\ III: \quad & \mu_i,\nu_i &&\geq 0. \end{aligned}$$

Which means we need to have for all $i = 1,\ldots,n$:

$$\begin{aligned} I.1: \quad & 2\lambda\alpha^T K_i &&= \textstyle\sum_{j=1}^n \mu_i y_i K_{ij} \\ I.2: \quad & \mu_i + \nu_i &&= \tfrac{1}{n} \\ II.1: \quad & \mu_i y_i \alpha^T K_i &&= \mu_i(1-\xi_i) \\ II.2: \quad & \nu_i\xi_i &&= 0 \\ III: \quad & \mu_i,\nu_i &&\geq 0. \end{aligned}$$

Using Equations I.2 and II.2 one obtains

$$\frac{1}{n}\xi_i = \mu_i\xi_i \overset{I.1}{=} \mu_i - \mu_i y_i \alpha^T K_i, \qquad (*)$$

and from I.1 one concludes

$$\begin{aligned}\lambda\alpha^T K\alpha &= \frac{1}{2}\left(\sum_{i=1}^{n}\mu_i\mu_j K_{ij}\right)_{j=1,\dots,n}\alpha \\ &= \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\mu_i\mu_j K_{ij}\alpha_j \\ &= \frac{1}{2}\sum_{i=1}^{n}\mu_i\mu_j\alpha^T K_i. \qquad (**)\end{aligned}$$

Hence the Lagrangian can be reformulated as

$$\begin{aligned}\Lambda(\alpha,\xi,\mu,\nu) = \lambda\alpha^T K\alpha + \frac{1}{n}\sum_{i=1}^{n}\xi_i &\overset{*}{=} \lambda\alpha^T K\alpha + \sum_{i=1}^{n}\mu_i - \sum_{i=1}^{n}\mu_i y_i\alpha^T K_i \\ &\overset{**}{=} \sum_{i=1}^{n}\mu_i - \frac{1}{2}\sum_{i=1}^{n}\mu_i y_i\alpha^T K_i \\ &\overset{I.1}{=} \sum_{i=1}^{n}\mu_i - \frac{1}{2}\sum_{i=1}^{n}\mu_i y_i\frac{1}{2\lambda}\sum_{j=1}^{n}\mu_j y_j K_{ij} \\ &= \sum_{i=1}^{n}\mu_i - \frac{1}{4\lambda}\sum_{i=1}^{n}\sum_{j=1}^{n}\mu_i\mu_j y_i y_j K_{ij}.\end{aligned}$$

where the condition $\mu_i, \nu_i \geq 0$ for all $i = 1, \dots, n$ can be fulfilled by demanding μ to be within $\left[0, \frac{1}{n}\right]$. Hence maximising $\Phi(\mu, \nu)$ with respect to μ and ν is equivalent to the optimisation problem stated in this lemma. To construct a solution to the primal problem let μ^* be a solution to the optimisation problem given in Lemma 2.28. We have just shown that this means (μ^*, ν^*) with $\nu^* = \frac{1}{n} - \mu^*$ is a solution to the Lagrangian dual problem. For all $i = 1, \dots, n$ let

$$\begin{aligned}\alpha_i^* &= \frac{y_i\mu_i^*}{2\lambda} \\ \xi_i^* &= \begin{cases} 0 & \text{if } \mu_i^* = 0 \\ 1 - y_i\alpha^{*T}K_i & \text{else,}\end{cases}\end{aligned}$$

which implies $\mu_i^*\xi_i^* = \mu_i^*(1 - y_i\alpha^{*T}K_i)$ for all $i = 1, \dots, n$ and therefore

$$\sum_{i=1}^{n}\mu_i^*(1 - \xi_i^* - y_i\alpha^{*T}K_i) = 0.$$

Moreover we have

$$\sum_{i=1}^{n} \nu_i^* \xi_i^* = \sum_{i=1}^{n} (\frac{1}{n} - \mu_i^*)\xi_i^* = \frac{1}{n}\sum_{i=1}^{n} \xi_i^* - \sum_{i=1}^{n} \mu_i^* \xi_i^*$$

and as a consequence the Lagrangian can be calculated as

$$\Lambda(\alpha^*, \xi^*, \mu^*, \nu^*) = \frac{1}{4\lambda}\sum_{i=1}^{n}\sum_{j=1}^{n} \mu_i^*\mu_j^* y_i^* y_j^* K_{ij} + \sum_{i=1}^{n} \mu_i^* - \frac{1}{2\lambda}\sum_{i=1}^{n}\sum_{j=1}^{n} \mu_i^*\mu_j^* y_i^* y_j^* K_{ij}$$

Which simplifies to

$$\sum_{i=1}^{n} \mu_i^* - \frac{1}{4\lambda}\sum_{i=1}^{n}\sum_{j=1}^{n} \mu_i^*\mu_j^* y_i^* y_j^* K_{ij} = \Phi(\mu^*, \nu^*).$$

Therefore, we know, by duality, that (α^*, ξ^*) is a solution to the primal problem stated in Definition 2.27. Moreover, this means α^* is a minimiser of the risk given in Equation 13. □

2.5. Selection of Tuning Parameters

Up to this point we have seen the tuning parameter $\lambda > 0$ as a fixed value that is priorly chosen to avoid to close adaptation to the data $\mathcal{D}$. By the law of large numbers, we know that for $n = |\mathcal{D}|$ tending to infinity, the empirical risk tends to the actual risk for all decision functions f. Therefore, intuitively, for small training data we need a large λ to control the generalization error.

Cross validation In practice this behaviour gives no advise for selecting a specific parameter, thought. Hence usually one considers a finite set of tuning parameters $\lambda_1, \ldots, \lambda_m$. To choose the best one among them we try to minimise the expected risk for decision functions $f_{\lambda_1}, \ldots, f_{\lambda_m}$. Here f_{λ_i} is the minimiser of the regularized risk (Definition 2.5) with tuning parameter λ_i for all $i = 1, \ldots, m$, respectively. The risk (Equation 1) is then approximated by applying a **k-fold cross validation** algorithm. For $k = n$ this coincides with the leave-one-out algorithm. Here the risk

is approximated by

$$\mathcal{R}_P(f_{\lambda_j}) \approx \frac{1}{n}\sum_{j=1}^{n} \mathcal{L}(f^*_{\lambda_j}(x_j), y_j) \quad \text{for all } j = 1, \ldots, m \tag{14}$$

where $f^*_{\lambda_j}$ denotes the minimiser of the regularised risk with tuning parameter λ_j and data $\mathcal{D}_j = \{(x_i, y_i) | i = 1, \ldots, n, i \neq j\}$. Since $\mathcal{D} \approx \mathcal{D}_j$ we expect $f_{\lambda_j} \approx f^*_{\lambda_j}$ and therefore, an nearly unbiased estimate of the expected loss with respect to P.

When a classification task is performed one might actually be interested in the $0-1$ loss but be using the hinge loss for optimisation (as this loss is convex, see Theorem 2.6). For the ultimate selection of a tuning parameter λ and consequently a decision function f_λ one can replace the loss $\mathcal{L}$ in Equation 14 by any desired loss function. Moreover, similarly to the selection of a tuning parameter, one can use k-fold cross-validation to select a kernel among a finite set of possible kernels. For example to decide for a certain parameter $\gamma \in \mathbb{R}$ when considering Gaussian kernels (see Lemma 2.26).

Data independent selection for universal kernels Additionally to this data dependent way of choosing the tuning parameter λ, for a certain class of kernels there exists a sequence $(\lambda_n)_{n\in\mathbb{N}}$ that ensures consistency of the classification algorithm. Hence this second approach provides a parameter λ_n for a given data set of size $|\mathcal{D}| = n$, previously to minimising the regularised risk.

Before giving the main result of this subsection (Theorem 2.33), we will formally define what it means for a classifier to be consistent.

Definition 2.29 (Classifier)
For a measurable input space X equipped with σ-Algebra $\mathcal{A}$ let

$$(X \times \{-1, +1\})_\infty = \{\mathcal{D} \subseteq X \times \{-1, +1\} |\ |\mathcal{D}| < \infty\}$$

be the set of all finite training data sets. A classifier $\mathcal{C}$ is map that assigns every $\mathcal{D} \in (X \times \{-1, +1\})_\infty$ a decision function $f_\mathcal{D}$. That is

$$\begin{aligned} \mathcal{C} : (X \times \{-1, +1\})_\infty &\to \{f \in X^\mathbb{R} |\ f \text{ Borel measurable}\} \\ \mathcal{D} &\mapsto f_\mathcal{D}. \end{aligned}$$

In the context of Support Vector Machines the map that assigns every data set $\mathcal{D}$ the minimiser of the regularised risk (Equation 3) is a classifier for fixed $\lambda \in \mathbb{R}$. The general aim for a classifier is for $\mathcal{D}$ being drawn independently from $X \times \{-1, 1\}$ with respect to some distribution P, to find a decision function that approximately minimises the risk of misclassification. This smallest achievable risk of misclassification is called Bayes risk.

Definition 2.30 (Bayes risk)
For any probability distribution P on $X \times \{-1, +1\}$ is

$$\mathcal{R}_P = \inf\{\mathcal{R}_P(f) |\ f \in X^{\mathbb{R}},\ f \text{ Borel measurable}\}$$

the Bayes risk of P. Here the risk of a measurable $f : X \to \mathbb{R}$ is defined to be

$$\mathcal{R}_P(f) = P(\text{sign}(f(x)) \neq y).$$

It is clear that not every classification method used in practice can be expected to achieve the Bayes risk. If this strong property holds for a classifier it is called universally consistent. Formally we have

Definition 2.31 (Universal consistency)
Let $\mathcal{D}$ being drawn identically and independently as elements of $X \times \{-1, 1\}$ with respect to some distribution P. A classifier is said to be universally consistent if

$$\mathcal{R}_P(f_{\mathcal{D}}) \to \mathcal{R}_P \text{ for } |\mathcal{D}| \to \infty$$

in probability with respect to P.

In general we do not expect the SVM classifier to be universally consistent. If for instance a linear kernel is used, only linear separating functions can be obtained. This means every distribution P that causes non-linear structure has a lower Bayes risk than the risk achievable by linear separation. Hence we only expected the Bayes risk to be achieved by classifiers based on Support Vector Machines with kernel k, if the corresponding reproducing kernel Hilbert space is large enough. Precisely we want a universal kernel.

Definition 2.32 (Universal kernel)
Let X be a compact metric space. Let $k : X \times X \to \mathbb{R}$ be a continuous kernel. k is called universal, if for the corresponding reproducing kernel Hilbert space $\mathcal{H}_k$ holds

$$\overline{\mathcal{H}_k}^{\|\cdot\|_\infty} = C(X).$$

That is $\mathcal{H}_k$ being dense in $C(X)$. $C(X)$ denotes the space of real valued, continuous functions on X equipped with the uniform norm $\|\cdot\|_\infty$.

It can be shown (see for example [16, page 155]) that the Gaussian kernel defined in Lemma 2.26 is universal on every compact $X \subset \mathbb{R}^d$. The next Theorem shows that there exist suitable null-sequences $(\lambda_n)_{n\in\mathbb{N}}$ such that a SVM classifier based on a universal kernel is universally consistent.

Theorem 2.33 (SVMs with universal kernels are universally consistent)
Let $k : X \times X \to \mathbb{R}$ be a universal kernel on a compact metric space X and let L be the hinge loss. Moreover, let $(\lambda_n)_{n\in\mathbb{N}}$ with $\lambda_n > 0\ \forall n \in \mathbb{N}$ be such that

$$\lambda_n \to 0 \quad \text{for } n \to \infty$$
$$n\lambda_n^2 \to \infty \quad \text{for } n \to \infty.$$

Then the classifier based on a Support Vector Machine with kernel k and the risk without offset (Equation 3) is universally consistent.

Proof. See Theorem 3.20 in [15, page 136]. □

Remark 2.34
*Steinwart also gives a stronger assumption on the sequence $(\lambda_n)_{n\in\mathbb{N}}$ to ensure universal consistency of the SVM classifier **with offset**. Here the sequence of tuning parameter $(\lambda_n)_{n\in\mathbb{N}}$ should achieve $\frac{n}{\log(n)}\lambda_n^2 \to \infty$ (see Example 1.1 in [15]).*

Similar results exists for other loss functions than the hinge loss. Furthermore, the condition on $(\lambda_n)_{n\in\mathbb{N}}$ can be weakened for certain kernels or under restriction of the probability distribution P.

Nevertheless, Theorem 2.33 only requires a certain asymptotic behaviour of the sequence of parameters $(\lambda_n)_{n\in\mathbb{N}}$. Hence it does not tell us how to choose one parameter for fixed sample size, as the limit is for example not influenced by multiplication with a constant value. Thus, rather than giving concrete advice on how to select one parameter λ, the results of this subsection can partially explain why Support Vector Machines based on universal kernels adapt well.

3. Convex Sets as Data Points

In this section we look at an input space X_c that consists of all compact and convex subsets of $\mathbb{R}^d$. Hence we define

$$X_c = \{A \subset \mathbb{R}^d | A \text{ compact and convex}\}.$$

Since in Section 2 the input space was not further defined, all theoretical results apply to the input space X_c without any limitations. In all of the following examples the hinge loss is used and the parameter λ is set to 1, as long as not stated otherwise.

3.1. Towards a Kernel for Convex Sets

Some known kernel functions on $\mathbb{R}^d$, for example the Gaussian kernel, are based on the Euclidean distance as a dissimilarity measure. To adapt those kernels to convex sets as data points, one has to find a suitable distance function on X_c. Do and Poulet [4] suggest to replace the Euclidean distance in the formula for the Gaussian kernel by the Hausdorff distance d_H. However, the resulting kernel is not necessarily positive definite.

Definition 3.1 (Hausdorff distance)
For $A, B \in X_c$ let

$$d_H(A, B) = \max\{\sup_{a \in A} \inf_{b \in B} \|a - b\|_2, \sup_{b \in B} \inf_{a \in A} \|a - b\|_2\}.$$

Example 3.2
Let $d = 1$ and $I_1 = [0.5, 1.4]$, $I_2 = [1, 1.1]$, $I_3 = [0.5, 0.6]$, $I_4 = [0, 0.9]$. The Gram matrix for the Hausdorff distance can be shown to be

$$\begin{pmatrix} 0.00 & 0.50 & 0.80 & 0.50 \\ 0.50 & 0.00 & 0.50 & 1.00 \\ 0.80 & 0.50 & 0.00 & 0.50 \\ 0.50 & 1.00 & 0.50 & 0.00 \end{pmatrix}.$$

Hence the Gram matrix for the kernel

$$k(A, B) = \exp(-d_H(A, B)^2) \quad \forall A, B \in X_c$$

can be computed as

$$K \approx \begin{pmatrix} 1.00 & 0.78 & 0.53 & 0.78 \\ 0.78 & 1.00 & 0.78 & 0.37 \\ 0.53 & 0.78 & 1.00 & 0.78 \\ 0.78 & 0.37 & 0.78 & 1.00 \end{pmatrix}.$$

For the determinant of K holds $\det(K) \approx -0.1$, hence K is not positive semi-definite which means the kernel k is not positive semi-definite.

It is problematic for a kernel to be indefinite as the construction of the reproducing kernel Hilbert space (see Theorem 2.19) strongly depends on the kernel being positive semi-definite. Moreover, the corresponding optimisation problem does not need to yield a unique minimiser. Even though Ong et al. [10] provide some results on adapting kernel methods for indefinite kernels, these are not covered by the classical theory for Support Vector Machines given in Section 2.

That is the reason why we try a different approach to obtain a positive semi-definite kernel. Instead of defining a kernel function directly, we assume a suitable feature map for convex subsets of $\mathbb{R}^d$.

3.2. Support Functions as a Feature Space

Definition 3.3 (Support function)
Let $X_c = \{A \subset \mathbb{R}^d | A$ compact and convex$\}$. For $A \in X_c$ define it's support function as

$$\begin{aligned} h_A : \mathbb{R}^d &\to \mathbb{R} \\ v &\mapsto \max_{a \in A} \langle a, v \rangle, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is the usual scalar product on $\mathbb{R}^d$.

Note that h_A is well defined for all $A \in X_c$, since $\langle \cdot, v \rangle$ is continuous for all $v \in \mathbb{R}^d$. Thus, it attains its maximum on all compact $A \subset \mathbb{R}^d$. Intuitively its value can be understood as the greatest signed distance from A to the origin in direction $v \in \mathbb{R}^d$ (see Figure 3). To see this let for $a \in A$

$$a_v = \frac{\langle a, v \rangle}{\|v\|_2^2} v$$

be the projection of a on span(v). Then we have $\|a_v\|_2 = \frac{|\langle a,v \rangle|}{\|v\|_2}$ and consequently maximising $\langle a, v \rangle$ is equivalent to maximising $\text{sign}(\langle a, v \rangle)\|a_v\|_2$. Moreover, we get

$$h_A(v) = \begin{cases} \max\limits_{a \in A} \langle a, v \rangle = \max\limits_{a \in A} \|a_v\|_2 & \text{if } \exists a \in A: \ \langle a, v \rangle \geq 0 \\ \max\limits_{a \in A} \langle a, v \rangle = -\min\limits_{a \in A} \|a_v\|_2 & \text{else.} \end{cases}$$

for all $v \in \mathbb{R}^d$ with $\|v\|_2 = 1$.

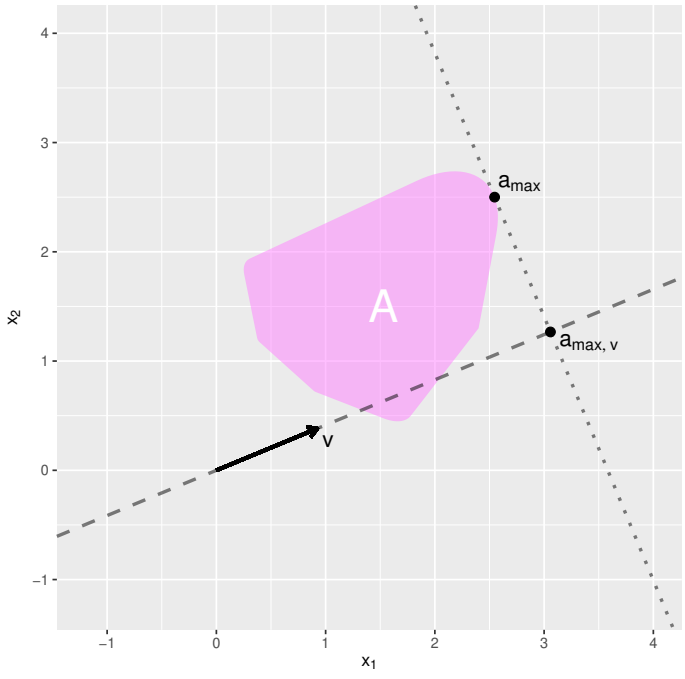


Figure 3: Geometrical interpretation of support functions. The length of $a_{max,v}$ corresponds to the value of the support function $h_A(v) = \langle a_{max}, v \rangle$ for $\|v\|_2 = 1$.

Lemma 3.4
The support function h_A is Lipschitz continuous for all $A \in X_c$.

Proof. Let $A \in X_c$ and $v_1, v_2 \in \mathbb{R}^d$. Then we have

$$\begin{aligned}\max_{a\in A}\langle a, v_1\rangle &= \max_{a\in A}\left(\langle a, v_1 - v_2\rangle + \langle a, v_2\rangle\right)\\ &\leq \max_{a\in A}\langle a, v_1 - v_2\rangle + \max_{a\in A}\langle a, v_2\rangle.\end{aligned}$$

Hence we get

$$\max_{a\in A}\langle a, v_1\rangle - \max_{a\in A}\langle a, v_2\rangle \leq \max_{a\in A}\langle a, v_1 - v_2\rangle \leq \max_{a\in A}|\langle a, v_1 - v_2\rangle| \leq \max_{a\in A}\|a\|_2\|v_1 - v_2\|_2,$$

where we used the Cauchy-Schwarz Inequality (Lemma A.2). Since A is bounded as a compact set, there is a $M \in R$ such that $\|a\|_2 \leq M$ for all $a \in A$. Therefore, by exchanging v_1 and v_2 we conclude

$$|h_A(v_1) - h_A(v_2)| = |\max_{a\in A}\langle a, v_1\rangle - \max_{a\in A}\langle a, v_2\rangle| \leq M\|v_1 - v_2\|_2,$$

which shows that h_A is Lipschitz continuous. □

Remark 3.5
The support function h_A is positive homogeneous, that is for all $\lambda > 0$ and all $v \in \mathbb{R}^d$ holds

$$h_A(\lambda v) = \max_{a\in A}\lambda v = \lambda \max_{a\in A} v = \lambda h_A(v).$$

Hence h_A is fully determined by its values on the unit sphere S_{d-1}.

Therefore, we will look at restrictions of support functions on the unit sphere S_{d-1}. Firstly notice that those restrictions belong to a Hilbert space.

Lemma 3.6
The support function h_A is square-integrable on the unit sphere. More precisely we have $h_A|_{S_{d-1}} \in L_2(S_{d-1})$ for all $A \in X_c$.

Proof. Let $A \in X_c$. Since we have shown in Lemma 3.4 that h_A is continuous on $\mathbb{R}^d$ it is borel measurable. Furthermore, $|h_A|$ attains its maximum $M \in \mathbb{R}$ on the

compact set S_{d-1}. Thus, we have

$$\int_{S_{d-1}} |h_A(v)|^2 dv \leq \int_{S_{d-1}} M^2 dv = M^2 |S_{d-1}| < \infty,$$

where $|S_{d-1}|$ denotes the surface area of the unit sphere as a $d-1$ dimensional manifold embedded in $\mathbb{R}^d$. □

Definition 3.7 (Kernel for convex sets)
Let $X_c = \{A \subset \mathbb{R}^d | A$ compact and convex$\}$. Define the feature map (for a feature space $L_2(S^{d-1})$) by

$$\begin{aligned} \phi : X_c &\to L_2(S_{d-1}) \\ A &\mapsto \sqrt{\frac{d}{|S_{d-1}|}} h_A. \end{aligned}$$

Denote by k_c the corresponding kernel on X_c.

Hence a feature map on X_c is given by mapping any compact and convex subset of $\mathbb{R}^d$ to a multiple of its support function. The Lebesgue space $L_2(S_{d-1})$ acts here as a feature space. This implies that k_c can be written as

$$k_c(A_1, A_2) = \langle \phi(A_1), \phi(A_2) \rangle_{L_2(S^{d-1})} = \frac{d}{|S_{d-1}|} \int_{S_{d-1}} \max_{a \in A_1} \langle a, v \rangle \max_{a \in A_2} \langle a, v \rangle \, dv \qquad (15)$$

for all $A_1,\ A_2 \in X$. The constant $\dfrac{d}{|S_{d-1}|}$ is introduced to ensure that this kernel actually generalises linear Support Vector Machines on $\mathbb{R}^d$.

Lemma 3.8
For point sets $A = \{a\}$ and $B = \{b\}$ we have

$$k_c(A, B) = a^T b.$$

Hence k_c acts on point sets like the usual inner product on $\mathbb{R}^d$.

Proof. Let $A = \{a\}$ and $B = \{b\}$. Then we have

$$\begin{aligned} k_c(A, B) &= \frac{d}{S_{d-1}} \int_{S_{d-1}} \max_{a \in A} \langle a, v \rangle \max_{b \in B} \langle b, v \rangle \, dv \\ &= \frac{d}{S_{d-1}} \int_{S_{d-1}} a^T v b^T v \, dv \\ &= \frac{d}{S_{d-1}} \int_{S_{d-1}} \sum_{i=1}^{d} \sum_{j=1}^{d} a_i v_i b_j v_j \, dv \\ &= \frac{d}{S_{d-1}} \sum_{i=1}^{d} \sum_{j=1}^{d} a_i b_j \int_{S_{d-1}} v_i v_j \, dv \\ &= \sum_{i=1}^{d} a_i b_i \frac{d}{S_{d-1}} \int_{S_{d-1}} v_i^2 \, dv \\ &= a^T b \end{aligned}$$

where we used

$$\int_{S_{d-1}} v_i v_j \, dv = 0 \quad \forall i \neq j,$$

with $i, j \in 1, \ldots, d$, which is due to symmetry and

$$d \int_{S_{d-1}} v_i^2 \, dv = \sum_{i=1}^{d} \int_{S_{d-1}} v_i^2 \, dv = \int_{S_{d-1}} \|v\|^2 \, dv = \int_{S_{d-1}} 1 \, dv = |S_{d-1}|$$

for all $i = 1, \ldots, d$. □

The next lemma shows that every minimiser of the regularised empirical risk actually acts like an affine linear classifier on sets of equal shape.

Lemma 3.9
Let $\mathcal{D} = \{(A_i, y_i) | A_i \in X_c,\ y_i \in \{-1, +1\},\ i = 1, \ldots, n\}$ and let $\mathcal{H}_c$ be the reproducing kernel Hilbert space associated with k_c. Moreover, let $f \in \mathcal{H}_c$ be a minimiser of

$$\begin{aligned} \mathcal{R} : \mathcal{H}_c &\to \mathbb{R} \\ f &\mapsto \lambda \|f\|_{\mathcal{H}_c}^2 + \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(A_i), y_i). \end{aligned}$$

Let $B \in X_c$, then

$$\begin{aligned} f_B : \mathbb{R}^d &\to \mathbb{R} \\ x &\mapsto f(x+B) \end{aligned}$$

is affine linear.

Proof. Let $x \in \mathbb{R}^d$. We have

$$\max_{a \in (x+B)} \langle a, v \rangle = \max_{b \in B} \langle x + b, v \rangle = \max_{b \in B} \left(\langle x, v \rangle + \langle b, v \rangle \right) = \langle x, v \rangle + \max_{b \in B} \langle b, v \rangle$$

and therefore,

$$\begin{aligned} k_c(A_i, x+B) &= \frac{d}{|S_{d-1}|} \int_{S_{d-1}} \max_{a \in A_i} \langle a, v \rangle (\langle x, v \rangle + \max_{b \in B} \langle b, v \rangle) \, dv \\ &= \frac{d}{|S_{d-1}|} \int_{S_{d-1}} \max_{a \in A_i} \langle a, v \rangle \sum_{j=1}^{d} x_j v_j \, dv + k_c(A_i, B) \\ &= \sum_{j=1}^{d} x_j \frac{d}{|S_{d-1}|} \int_{S_{d-1}} \max_{a \in A_i} \langle a, v \rangle \langle e_j, v \rangle \, dv + k_c(A_i, B) \\ &= \langle x, k_c(A_i, \{e_j\})_{j=1,\dots,d} \rangle + k_c(A_i, B) \end{aligned}$$

for all $i = 1, \dots, n$. Here does e_j denote the j-th unit vector in $\mathbb{R}^d$. Since we have by Corollary 2.24 (Representer Theorem)

$$f(x+B) = \sum_{i=1}^{n} \alpha_i k_c(A_i, x+B)$$

for all $x \in \mathbb{R}^d$, we conclude that f is affine linear in $x \in \mathbb{R}^d$. □

Remark 3.10

1. *The same result holds true when a risk with offset is used.*
2. *If we choose $B = \{0\}$ we have $x + B = x \in \mathbb{R}^d$ and $k_c(A_i, B) = 0$ for all $i = 1, \dots, n$. Hence this lemma shows that the restriction of f on points sets is linear.*

3. *The proof shows that the parameter of the decision function*

$$f(x) = \langle w, x \rangle + b \quad \forall x \in \mathbb{R}^d$$

can be computed as

$$w = \sum_{i=1}^{n} \alpha_i k_c(A_i, \{e_j\})_{j=1,\dots,d}$$
$$b = \sum_{i=1}^{n} \alpha_i k_c(A_i, B).$$

Lemma 3.9 gives an intuitive understanding of how Support Vector Machines with kernel k_c act with respect to the position of sets. As an example, Figure 4 shows how a minimiser separates points. That is setting $B = \{0\}$ in Lemma 3.9.

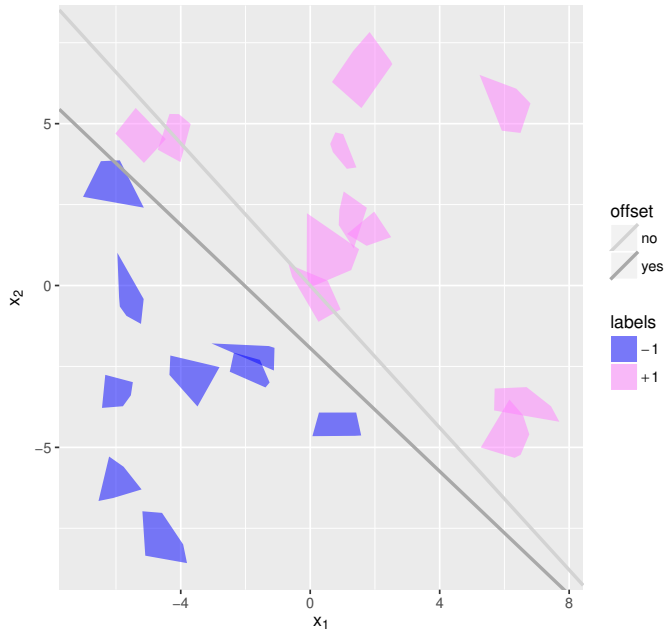


Figure 4: Restricting the minimising functional to point sets for the risk without and with offset. The grey lines indicate the separating hyperplanes respectively.

However, it is not straightforward to see how the shape of sets $A_i \in X_c$, $i = 1, \ldots, n$ influences the resulting decision function and how this function then acts on sets of different shape. Since calculating the kernel k_c for sets of arbitrary shape is numerically expensive we will restrict X_c to the set of d-dimensional intervals.

3.3. Restriction on Interval Data

Let $X_I \subset X_c$ be the set of all d-dimensional intervals. That is

$$X_I = \{A \in X_c |\ A = \bigtimes_{i=1}^{d} [a_{-1i}, a_{+1i}] \text{ with } a_{-1i} \leq a_{+1i}\ \forall i = 1, \ldots, d\}.$$

The next theorem will show that evaluating k_c on d-dimensional intervals can be simplified. Afterwards this representation of the reduced kernel is used to understand how Support Vector Machines using k_c act on interval-valued data.

Theorem 3.11
Let $A, B \in X_I$ with $A = \bigtimes_{i=1}^{d} [a_{-1i}, a_{+1i}]$ and $B = \bigtimes_{i=1}^{d} [b_{-1i}, b_{+1i}]$. Then the kernel for convex sets evaluated at sets A and B can be computed as

$$k_c(A, B) = \frac{1}{2} a^T M b,$$

with $a = (a_{-11}, a_{+11}, \ldots, a_{-1d}, a_{+1d})^T \in \mathbb{R}^{2d}$, $b = (b_{-11}, b_{+11}, \ldots, b_{-1d}, b_{+1d})^T \in \mathbb{R}^{2d}$ and a matrix $M \in \mathbb{R}^{2d \times 2d}$ defined by

$$M = \begin{pmatrix} \mathbb{1}_2 & A & \ldots & \ldots & A \\ A & \mathbb{1}_2 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \mathbb{1}_2 & A \\ A & \ldots & \ldots & A & \mathbb{1}_2 \end{pmatrix} \quad \text{with } \mathbb{1}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ A = \begin{pmatrix} \frac{1}{\pi} & -\frac{1}{\pi} \\ -\frac{1}{\pi} & \frac{1}{\pi} \end{pmatrix}.$$

Proof. For $s \in \{-1,1\}^d$ let $S_{d-1}(s) = \{v \in S_{d-1} | v_i s_i > 0,\ \forall i = 1,\dots,d\}$. Then we have for all $v \in S_{d-1}(s)$

$$h_A(v) = \max_{a\in A}\langle a, v\rangle = \max_{a\in A}\sum_{i=1}^{d} a_i v_i = \sum_{i=1}^{d}\max_{a_{-1i}\le a_i\le a_{+1i}} a_i v_i = \sum_{i=1}^{d} a_{s_i i} v_i = a_s^T v$$

and analogously

$$h_B(v) = b_s^T v$$

where $a_s = (a_{s_i i})_{i=1,\dots,d}^T$ and $b_s = (b_{s_i i})_{i=1,\dots,d}^T$. Hence the integral associated with the kernel can be written as

$$\begin{aligned}\int_{S_{d-1}} h_A(v)h_B(v)\,dv &= \sum_s \int_{S_{d-1}(s)} a_s^T v b_s^T v\,dv \\ &= \sum_s \int_{S_{d-1}(s)} \sum_{i,j=1}^{d} (a_s b_s^T)_{i,j} v_i v_j\,dv \\ &= \sum_s \sum_{i,j=1}^{d} a_{s_i i} b_{s_j j} \int_{S_{d-1}(s)} v_i v_j\,dv.\end{aligned}$$

Due to symmetry and Lemma 3.12 we have

$$\int_{S_{d-1}(s)} v_i v_j\,dv = s_i s_j \int_{S_{d-1}^+} v_1 v_2\,dv = \frac{s_i s_j\,|S_{d-1}|}{2^{d-1}d\pi} \quad \forall i \neq j$$

where $S_{d-1}^+ = S_{d-1}(s)$, $s = (1,\dots,1)^T$. We get for $i = j$:

$$\int_{S_{d-1}(s)} v_i v_j\,dv = \int_{S_{d-1}^+} v_i^2\,dv = \frac{|S_{d-1}|}{2^d d}$$

since

$$|S_{d-1}| = \int_{S_{d-1}} 1\,dv = \sum_{i=1}^{d}\int_{S_{d-1}} v_i^2\,dv = \sum_{i=1}^{d}\sum_s \int_{S_{d-1}(s)} v_i^2\,dv = d2^d \int_{S_{d-1}^+} v_i^2\,dv$$

for all $i = 1,\dots,n$.

Hence we conclude:

$$\begin{aligned}
\int_{S_{d-1}} h_A(v)h_B(v)\ dv &= \sum_{i,j=1}^{d} \sum_{s} a_{s_i i} b_{s_j j} \int_{S_{d-1}(s)} v_i v_j \, dv \\
&= \sum_{i,j=1}^{d} 2^{d-2} \sum_{s_i,s_j\in\{-1,1\}} a_{s_i i} b_{s_j j} \int_{S_{d-1}(s)} v_i v_j \, dv \\
&= 2^{d-2} \sum_{i,j=1}^{d} \sum_{s_i,s_j\in\{-1,1\}} a_{s_i i} b_{s_j j} s_i s_j \int_{S^+_{d-1}} v_i v_j \, dv \\
&= \frac{2^{d-2}\,|S_{d-1}|}{2^{d-1}d} \left(\sum_{i=1}^{d} \sum_{s_i\in\{-1,1\}} a_{s_i i} b_{s_i i} + \sum_{\substack{i,j=1 \\ i\neq j}}^{d} \sum_{s_i,s_j\in\{-1,1\}} \frac{s_i s_j}{\pi} a_{s_i i} b_{s_j j} \right) \\
&= \frac{|S_{d-1}|}{2d} \left(\sum_{i=1}^{2d} a_i b_i + \sum_{\substack{i,j=1 \\ |i-j|\geq 2}}^{2d} \frac{(-1)^{i+j}}{\pi} a_i b_j \right).
\end{aligned}$$

Therefore dividing the last equation by $\frac{d}{|S_{d-1}|}$ yields the desired result. □

Lemma 3.12
For all $d \in \mathbb{N}, d \geq 2$ holds

$$\int_{S^+_{d-1}} v_1 v_2 \ dv = \frac{|S_{d-1}|}{2^{d-1}d\pi}.$$

Here S_{d-1} denotes the unit sphere, that is the surface area of the d-dimensional unit ball and $S^+_{d-1} = \{v \in S_{d-1} |\ v_i \geq 0\ \forall i = 1, \ldots, d\}$.

Proof. By the Fundamental Theorem of Calculus we have

$$\int_{S^+_{d-1}} v_1 v_2 \ dv = \frac{d}{dR} \int_{r=0}^{R} \int_{S^+_{d-1}(r)} v_1 v_2 \ dv \ dr \Big|_{R=1} = \frac{d}{dR} \int_{B^+_d(R)} v_1 v_2 \ dv \Big|_{R=1},$$

where the last equality is due to Theorem A.4 (Integration in spherical coordinates). Here does $S_{d-1}(r)$ denote the surface area of $B_d(r)$, the d-dimensional ball with center at the origin and radius r. Using a transformation to polar coordinates from

two dimensional Cartesian coordinates and Fubini's Theorem one computes

$$\begin{aligned}
\int\limits_{B_d^+(R)} v_1 v_2 \; dv &= \int\limits_{B_{d-2}^+(R)} \int\limits_{\substack{B_2^+(\sqrt{R^2-x}) \\ x=\sum_{i=3}^d v_i^2}} v_1 v_2 \; d(v_1, v_2) \; d(v_3, \ldots, v_d) \\
&= \int\limits_{B_{d-2}^+(R)} \int\limits_{r=0}^{\sqrt{R^2-\sum_{i=3}^d v_i^2}} \int\limits_{t=0}^{\frac{\pi}{2}} r^3 \cos(t) \sin(t) \; dt \; dr \; d(v_3, \ldots, v_d) \\
&= \int\limits_{B_{d-2}^+(R)} \left[\frac{r^4}{4}\right]_{r=0}^{\sqrt{R^2-\sum_{i=3}^d v_i^2}} \left[\frac{1}{2}\sin^2(t)\right]_{t=0}^{\frac{\pi}{2}} \; d(v_3, \ldots, v_d) \\
&= \frac{1}{8} \int\limits_{B_{d-2}^+(R)} (R^2 - \|v\|_2^2)^2 \; dv.
\end{aligned}$$

This can further be simplified by using spherical coordinates (Theorem A.4) once more;

$$\begin{aligned}
\int\limits_{B_d^+(R)} v_1 v_2 \; dv &= \frac{1}{8} \int\limits_{r=0}^{R} (R^2 - r^2)^2 |S_{d-3}^+(r)| \; dr \\
&= \frac{|S_{d-3}|}{8 \cdot 2^{d-2}} \int\limits_{r=0}^{R} (R^4 - 2R^2 r^2 + r^4)^2 r^{d-3} \; dr \\
&= \frac{|S_{d-1}|(d-2)}{2^{d+2}\pi} \left[\frac{R^4 r^{d-2}}{d-2} - \frac{2R^2 r^d}{d} + \frac{r^{d+2}}{d+2}\right]_{r=0}^{R} \\
&= \frac{|S_{d-1}| R^{d+2}}{2^{d+2}\pi} \left(1 - \frac{2(d-2)}{d} + \frac{(d-2)}{d+2}\right).
\end{aligned}$$

Here we used a recursive formula for the area of a unit sphere (Lemma A.5) to derive the third equality. The desired result is therefore obtained by

$$\frac{d}{dR} \int_{B_d^+(R)} v_1 v_2 \; dv \Big|_{R=1} = \frac{|S_{d-1}| R^{d+1}}{2^{d+2}\pi} \left(2d - \frac{2(d+2)(d-2)}{d}\right) \Big|_{R=1} = \frac{|S_{d-1}|}{2^{d-1} d\pi}.$$

□

The next lemma shows how solutions of Support Vector Machines with kernel k_c and input space X_I can be interpreted. It turns out that every minimiser can be decomposed as a sum of two linear functionals. One maps the midpoint vector of the d-dimensional intervals and the other the vector of interval lengths in every coordinate direction.

Lemma 3.13
Let $\mathcal{D} = \{(A_i, y_i) | A_i \in X_I, y_i \in \{-1, +1\}, i = 1, \ldots, n\}$ and let $\mathcal{H}_c$ be the reproducing kernel Hilbert space associated with k_c. Moreover, let $f \in \mathcal{H}_c$ be a minimiser of

$$\begin{aligned} \mathcal{R} : \mathcal{H}_c &\to \mathbb{R} \\ f &\mapsto \lambda \|f\|^2_{\mathcal{H}_c} + \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(A_i), y_i). \end{aligned}$$

Then there exist $w_1, w_2 \in \mathbb{R}^d$ such that

$$f(B) = w_1^T m(B) + w_2^T l(B)$$

for all $B \in X_I$.

The functions m and l assign the midpoint and the length in each direction to an d-dimensional interval, respectively. That is for $B = \bigtimes_{i=1}^{d} [b_{-1i}, b_{+1i}] \in X_I$

$$\begin{aligned} m(B) &= \frac{1}{2}(b_{-1i} + b_{+1i})_{i=1,\ldots,d} \\ l(B) &= (b_{+1i} - b_{-1i})_{i=1,\ldots,d}. \end{aligned}$$

Proof. Let $A, B \in X_I$ with $m(A) = x$, $l(A) = a$, $m(B) = y$, $l(B) = b$. Denote by

$$\begin{aligned} \tilde{x} &= (x_1, x_1, \ldots, x_d, x_d)^T \\ \tilde{a} &= \frac{1}{2}(-a_1, a_1, \ldots, -a_d, a_d)^T \\ \tilde{y} &= (y_1, y_1, \ldots, y_d, y_d)^T \\ \tilde{b} &= \frac{1}{2}(-b_1, b_1, \ldots, -b_d, b_d)^T \end{aligned}$$

where $\tilde{x}, \tilde{a}, \tilde{y}, \tilde{b} \in \mathbb{R}^{2d}$.

Defining $M \in \mathbb{R}^{2d \times 2d}$ like in Theorem 3.11 one computes

$$\begin{aligned}
\tilde{x}^T M \tilde{y} &= \tilde{x}^T \tilde{y} = 2x^T y \\
\tilde{x}^T M \tilde{b} &= \tilde{x}^T \tilde{b} = 0 \\
\tilde{a}^T M \tilde{y} &= \tilde{a}^T \tilde{y} = 0 \\
\tilde{a}^T M \tilde{b} &= \tilde{a}^T \tilde{b} + \frac{1}{\pi} \sum_{\substack{i,j=1 \\ i \neq j}}^{d} a_i b_j \\
&= \frac{1}{2} a^T b - \frac{1}{\pi} \sum_{i=1}^{d} a_i b_i + \frac{1}{\pi} \sum_{i,j=1}^{d} a_i b_j \\
&= \left((\frac{1}{2} - \frac{1}{\pi}) a^T + \frac{1}{\pi} \sum_{i=1}^{d} a_i (1, \ldots, 1) \right) b.
\end{aligned}$$

Hence we deduce (using Theorem 3.11) that the kernel k_c evaluated at sets $A, B \in X_I$ can be written as

$$\begin{aligned}
k_c(A, B) &= \frac{1}{2} (\tilde{x} + \tilde{a})^T M (\tilde{y} + \tilde{b}) \\
&= x^T y + \left((1 - \frac{2}{\pi}) a^T + \frac{2}{\pi} \sum_{i=1}^{d} a_i (1, \ldots, 1) \right) b.
\end{aligned}$$

Reconsidering the representation of the minimiser given in the Representer Theorem (Corollary 2.24) we obtain that w_1, w_2 are linear combinations of the prefactors of y and b in the equation above, respectively. □

The next two examples shall demonstrate what this decomposition means in practice for classifying interval data. In both examples the tuning parameter λ is set to one and the kernel for convex sets k_c is used. Evaluations are computed using the results of Theorem 3.11.

Example 3.14

Let (A, Y) *be a random variable with* $P(Y = -1) = P(Y = 1) = \frac{1}{2}$, $A \in X_I$ *with midpoint vector* $M \sim U([-100, 100]^2)$ *and length vector* L. *For the conditional distribution of the length* L *given the label* Y *shall hold*

$$L|Y \sim 4 \begin{pmatrix} 2 - Y \\ 2 + Y \end{pmatrix} + Exp(1)$$

where Exp(1) denotes a exponential distribution with rate parameter set to 1. Figure 5 shows 100 identically and independently drawn samples $(A_i, y_i) \sim (A, Y)$.

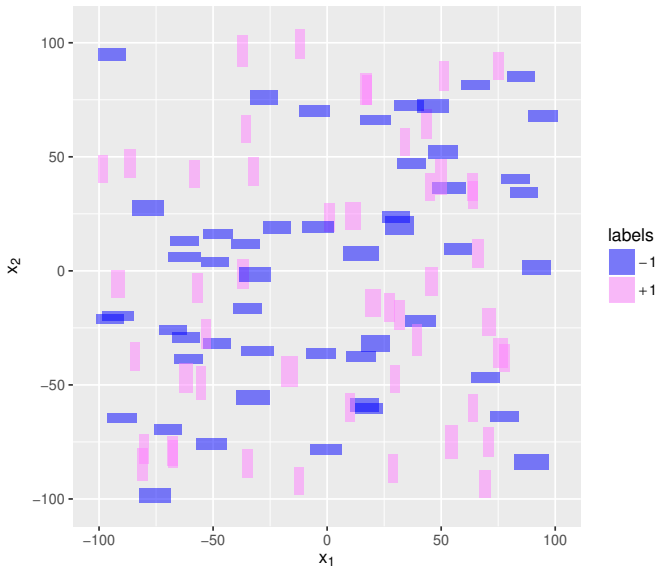


Figure 5: Interval data with non-predictive position but predictive shape.

This data is now used to train a Support Vector Machine with kernel k_c. *According to Theorem 3.11 and the Representer Theorem (Corollary 2.24) we obtain* $w \in \mathbb{R}^4$ *such that for the risk minimiser* f *and all* $B = [b_{-11}, b_{+11}] \times [b_{-12}, b_{+12}] \in X_I$ *holds*

$$f(B) = (b_{-11}, b_{+11}, b_{-12}, b_{+12})w.$$

Define $(m_1, m_2)^T = m(B)$ *to be the vector of midpoints and* $(l_1, l_2)^T = l(B)$ *to be the vector of lengths. Hence we can rewrite the decision functions* f *as*

$$\begin{aligned} f(B) = w^T \begin{pmatrix} m_1 - 0.5l_1 \\ m_1 + 0.5l_1 \\ m_2 - 0.5l_2 \\ m_2 + 0.5l_2 \end{pmatrix} &= w^T \begin{pmatrix} m_1 \\ m_1 \\ m_2 \\ m_2 \end{pmatrix} + 0.5w^T \begin{pmatrix} -l_1 \\ +l_1 \\ -l_2 \\ +l_2 \end{pmatrix} \\ &= w^T \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} + 0.5w^T \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \end{aligned}$$

For a simulated example data set $\mathcal{D} = \{(A_i, y_i) \overset{i.i.d}{\sim} (A, Y)\}$ *(see Figure 5) the risk is optimised by*

$$w = \begin{pmatrix} 0.1207 \\ -0.1215 \\ -0.1191 \\ 0.1186 \end{pmatrix}.$$

Thus $w_1, w_2 \in \mathbb{R}^2$ *defined like in Lemma 3.13 can be computed as*

$$\begin{aligned} w_1 &= \begin{pmatrix} 1, 1, 0, 0 \\ 0, 0, 1, 1 \end{pmatrix} w \approx \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ w_2 &= 0.5 \begin{pmatrix} -1, 1, 0, 0 \\ 0, 0, -1, 1 \end{pmatrix} w \approx \begin{pmatrix} -0.12 \\ 0.12 \end{pmatrix}. \end{aligned}$$

This shows that the position of the midpoints $m(B)$ *of the* 2*-dimensional interval* B *has no influence on the predicted label. Intervals which spread more in the first coordinate direction than in the second are predicted to have label* $y = -1$*, whereas intervals that spread more in the second coordinate direction than in the first one are assumed to have label* $y = +1$*. Hence the interval length* $l(B)$ *is predictive.*

Due to the decomposition given in Lemma 3.13 it is also clear that no Support Vector Machine with kernel k_c and no offset can distinguish between d-dimensional

intervals of different size. When we consider sets of form

$$B = [-b, b]^d \text{ for } b \in \mathbb{R}_0^+,$$

we get $f(B) = (-1, 1, \ldots, -1, 1)w_2 b$. This implies

$$\operatorname{sign}(f(B)) = \operatorname{sign}((-1, 1, \ldots, -1, 1)w_2),$$

hence all sets are assumed to have the same label. The next example shows how an offset can be used to avoid this behaviour.

Example 3.15
Let (A, Y) be a random variable with $P(Y = -1) = P(Y = 1) = \frac{1}{2}$ and $A \in X_I$ with midpoint vector $M \sim U([-50, 50]^2)$, and length vector L. The conditional distribution of the length L given the label Y is given by

$$L|Y \sim 2|\mathcal{N}(3 - Y, 0.8)|.$$

Here $\mathcal{N}(\mu, \sigma)$ denotes a normal distribution with mean μ and standard derivation σ. Figure 6 shows a sample of 100 independently and identically drawn sets according to the distribution of (A, Y).

Firstly, a Support Vector Machine without offset is trained on suchlike dataset of sample size 1000. Similarly to the example before, we obtain a risk minimiser represented by

$$w = \begin{pmatrix} 0.0583 \\ -0.0584 \\ 0.0587 \\ -0.0579 \end{pmatrix}.$$

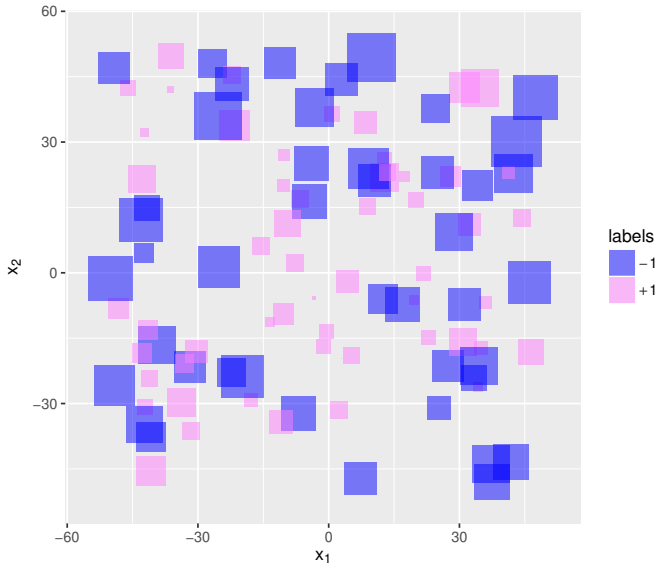


Figure 6: Interval data with non-predictive position but predictive size.

As expected, this corresponds to

$$w_1 \approx \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \text{ and } w_2 \approx \begin{pmatrix} -0.06 \\ -0.06 \end{pmatrix},$$

which means the classifier can not distinguish between interval sets of different size, since $f(B)$ is strictly negative for all two dimensional intervals B.

Secondly, an additional offset $d \in \mathbb{R}$ is considered. In this case we obtain the minimising values

$$w = \begin{pmatrix} 0.1747 \\ -0.1748 \\ 0.1752 \\ -0.1743 \end{pmatrix} \text{ and } d = 2.09.$$

This implies

$$w_1 \approx \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \text{ and } w_2 \approx \begin{pmatrix} -0.17 \\ -0.17 \end{pmatrix},$$

hence for $B \in X_I$ *with each side of same length* $l(B)$ *we get*

$$\begin{aligned} f(B) > 0 &\Leftrightarrow 2.09 - 2 \cdot 0.17 l(B) > 0 \\ &\Leftrightarrow l(B) < 6.1. \end{aligned}$$

When we compare the histogram of the lengths $l(B)$ *in the example data set differentiated by label (Figure 7), the calculated decision value seems to be reasonable.*

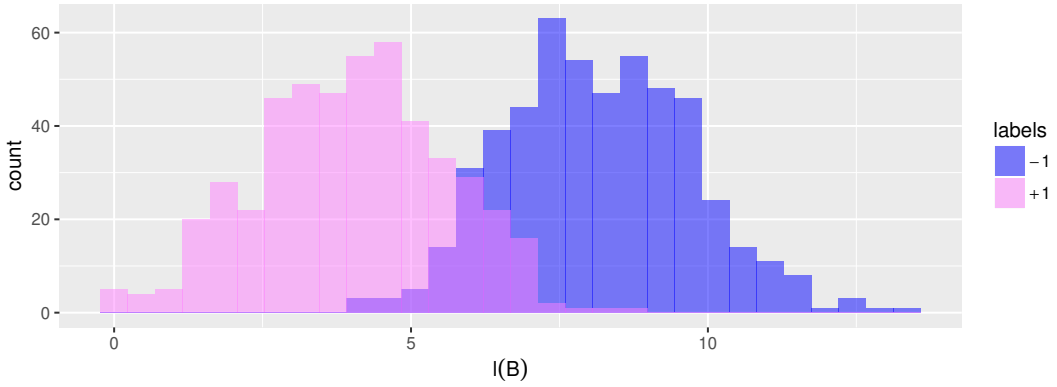


Figure 7: Size of the two dimensional intervals differentiated by label.

3.4. The Gaussian Kernel for Convex Sets

As seen in Lemma 3.13, every decision function obtained by a SVM with kernel k_c on interval data, is additive linear as a function of the position and the length of the interval. This is a very limiting behaviour, as not even linear interactions between position and shape can be detected using this kernel. Moreover, Example 3.15 shows that SVMs without offset and kernel k_c can not even distinguish between interval sets of different size. Hence it is questionable whether those Support Vector Machines can adapt well to sets of different shape.

To overcome these restrictions on the decision function, the "kernel trick" described in Subsection 2.3 can be used to perform more flexible classification of convex sets. Similarly as in Lemma 2.26 for kernels on $\mathbb{R}^d$, valid kernel functions on X_c can be constructed as transformations of k_c. Some examples are shown in the next lemma.

Lemma 3.16 (More kernel functions for convex sets)
Let $X_c = \{A \subset \mathbb{R}^d | \ A$ compact and convex$\}$ and k_c be the kernel on $X_c \times X_c$ defined in Definition 3.7. Then the following functions are valid kernels on $X_c \times X_c$:

1. ***Polynomial kernel:***

$$k(A_1, A_2) = (k_c(A_1, A_2) + c)^m$$

for all $A_1, A_2 \in X_c,\ c > 0,\ m \in \mathbb{N}$.

2. ***Exponential kernel:***

$$k(A_1, A_2) = \exp\left[\gamma k_c(\phi(A_1), \phi(A_2))\right]$$

for all $A_1, A_2 \in X_c,\ \gamma > 0$.

3. ***Gaussian kernel:***

$$k(A_1, A_2) = \exp\left(-\gamma \|\phi(A_1) - \phi(A_2)\|^2_{L_2(S^{d-1})}\right)$$

for all $A_1, A_2 \in X_c,\ \gamma > 0$. Here ϕ and $L_2(S^{d-1})$ denote the feature map and the feature space defined in Definition 3.7.

Proof. Analogously to the proof of Lemma 2.26 we use the results of Lemma 2.25, where the scalar product on $\mathbb{R}^d$ is replaced by k_c. The reasoning there only uses that the scalar product (or k_c respectively) is positive semi-definite. □

Remark 3.17
The Gaussian kernel for convex sets (Lemma 3.16) is similar to the indefinite kernel defined in the approach of Do and Poulet [4]. The kernel there can be written in terms of support functions as

$$k(A_1, A_2) = \exp(-\gamma d_H(A_1, A_2)^2) = \exp(-\gamma \|\phi(A_1), \phi(A_2)\|_\infty^2)$$

for all interval sets $A_1, A_2 \in X_I$ and $\gamma > 0$ (see Equation 16). The only difference is, that the uniform norm $\|\cdot\|_\infty$ on the space of support functions is replaced by the L^2-norm. This modification ensures positive semi-definiteness of the kernel.

To ensure asymptotic approximation of the best achievable decision function one might desire to have an universal kernel on X_c (see Theorem 2.33). This request is meaningful since X_c equipped with the Hausdorff distance (Definition 3.1) is a metric space. Moreover, the next lemma shows that little effort is needed to construct compact subsets of X_c.

Lemma 3.18
Let $K \subset \mathbb{R}^d$ be compact. Then $X_{c,0} = \{A \subseteq K |\ A \text{ compact}\}$ equipped with the Hausdorff distance d_H (Definition 3.1) is a compact metric space.

Proof. Henrikson [7] showed that $(X_{c,0}, d_H)$ is indeed a metric space (Proposition 2-2) which is totally bounded (Theorem 3-1) and complete (Theorem 3-3). Since every complete and totally bounded metric space is also compact (see [1, page 81]) we obtain that $(X_{c,0}, d_H)$ is compact. □

The following theorem is due to Christmann and Steinwart [3] and shows how to construct universal kernels on other input spaces than subsets of $\mathbb{R}^d$. This general result is then used to show that the Gaussian kernel defined in Lemma 3.16 is universal on a suitable input space.

Theorem 3.19 (Universal kernels on non-standard input spaces)
Let X be a compact metric space and $\mathcal{H}$ be a separable Hilbert space. Let $\phi : X \to \mathcal{H}$ be continuous and injective.

1. *Denote by $M = \sup_{x_1,x_2 \in X} \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{H}}$. Let $(a_i)_{i \in \mathbb{N}} \subset \mathbb{R}_0^+$ such that*

$$\sum_{i=0}^{\infty} a_i M^i < \infty.$$

   Then $k : X \times X \to \mathbb{R}$ defined by

$$k(x_1, x_2) = \sum_{i=0}^{\infty} a_i \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{H}}^i \quad \forall x_1, x_2 \in X,$$

   is a universal kernel.

2. *The Gaussian kernel given by*

$$k(x_1, x_2) = \exp(-\gamma \|\phi(x_1) - \phi(x_2)\|_{\mathcal{H}}^2) \quad \forall x_1, x_2 \in X,$$

   is universal for all constant $\gamma > 0$.

Proof. See Theorem 2.2 in [3, page 4]. □

We will see that all requirements of Theorem 3.19 are full filled for the reproducing kernel Hilbert space $\mathcal{H}_c$ belonging to k_c and every compact subset of X_c. In detail we can state:

Lemma 3.20
Let $X_{c,0} \subset X_c$ be compact and $k : X_{c,0} \times X_{c,0}$ be the exponential kernel or the Gaussian kernel defined in Lemma 3.16. Then k is universal.

Proof. Here we have $\mathcal{H} = L_2(S_{d-1})$ which is a separable Hilbert space. The exponential kernel can be written as

$$\begin{aligned} k(A_1, A_2) &= \exp\left[\gamma k_c(\phi(A_1), \phi(A_2))\right] \\ &= \sum_{i=0}^{\infty} \frac{(\gamma k_c(\phi(A_1), \phi(A_2)))^i}{i!} \\ &= \sum_{i=0}^{\infty} \frac{\gamma^i}{i!} \langle \phi(A_1), \phi(A_2) \rangle_{\mathcal{H}}^i \end{aligned}$$

for all $A_1, A_2 \in X_{c,0}$. The power series involved, $\sum_{i=0}^{\infty} \frac{(\gamma M)^i}{i!}$, converges for every $M \in \mathbb{R}^+$. Hence it is left to show that the feature map defined in 3.7 is continuous and injective in order to prove that both the exponential and the Gaussian kernel are universal. To do so we write the Hausdorff distance in terms of support functions. That is

$$d_H(A_1, A_2) = \|h_{A_1} - h_{A_2}\|_\infty \quad \forall A_1, A_2 \in X_c, \tag{16}$$

where $\|f\|_\infty = \sup_{v \in S_{d-1}} |f(v)|$ denotes the uniform norm on the unit sphere. A proof for this statement can be found in [12, page 66]. This immediately implies that ϕ is injective, as we have

$$d_H(A_1, A_2) = \sqrt{\frac{d}{S_{d-1}}} \|\phi(A_1) - \phi(A_2)\|_\infty \quad \forall A_1, A_2 \in X_c$$

and therefore, $A_1 = A_2$ whenever $\phi(A_1) = \phi(A_2)$.
Lastly, we conclude that ϕ is continuous since

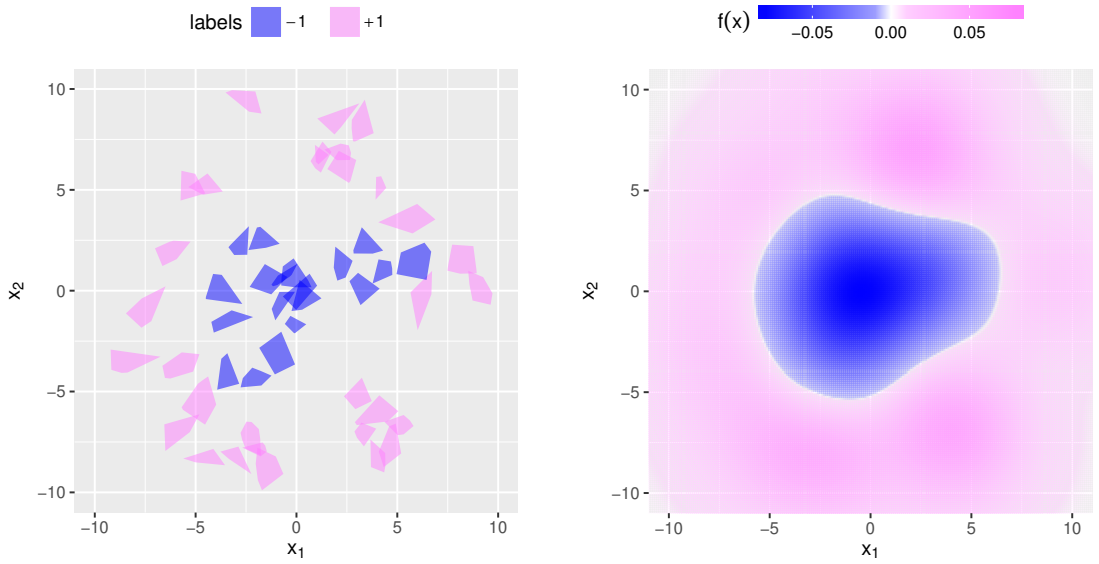
$$\begin{aligned} \|\phi(A_1) - \phi(A_2)\|_{L_2(S_{d-1})}^2 &= \|\sqrt{\frac{d}{S_{d-1}}} \left(h_{A_1} - h_{A_2}\right) \|_{L_2(S_{d-1})}^2 \\ &= \frac{d}{S_{d-1}} \int_{S_{d-1}} |h_{A_1} - h_{A_2}|^2 dv \\ &\leq \frac{d}{S_{d-1}} \int_{S_{d-1}} \|h_{A_1} - h_{A_2}\|_\infty^2 dv = d \cdot d_H(A_1, A_2). \end{aligned}$$

□

Remark 3.21

The SVM classifier based on the Gaussian kernel for convex sets is universally consistent; assuming a suitable loss function and an appropriate null sequence $(\lambda_n)_{n\in\mathbb{N}}$. *For further details see for example the assumptions given in Theorem 2.33 and the subsequent remark.*

Hence Support Vector Machines based on the Gaussian kernel for convex sets are expected to learn well for arbitrary distributions on the set of convex sets. To understand how decision functions obtained by SVMs with Gaussian kernel behave, we will look at the restriction to point sets and to interval sets again. Figure 8a shows convex data which is clearly not linearly separable.



(a) Example of convex data which is not linearly separable

(b) The decision function obtained for point sets $\{x\}$, $x \in \mathbb{R}^d$

Figure 8: Classification based on the Gaussian kernel for convex sets. Parameters λ and σ are set to 1 and 0.1 respectively.

The decision function in Figure 8b seems to describe the position of the convex sets differentiated by label well. It was computed applying the results of the next lemma (Lemma 3.22).

Lemma 3.22
Let k be the Gaussian kernel as defined in Lemma 3.16. Let $A \in X_c$ and $x \in \mathbb{R}^d$. Moreover denote by

$$\xi_A = k_c(A, \{e_j\})_{j=1,\dots,d} \in \mathbb{R}^d$$

a vector with kernel evaluations at set A and all unit vectors e_j for $j = 1, \dots, d$.

1. *The kernel evaluated at the point set $\{x\}$, $x \in \mathbb{R}$ can then be computed as*

$$k(A, \{x\}) = \exp\left[-\gamma(k_c(A, A) - \|\xi_A\|_2^2)\right] \exp\left[-\gamma\|x - \xi_A\|_2^2\right].$$

2. *Let $A, B \in X_I$ with $B = \overset{d}{\underset{i=1}{\times}}[-b_i, b_i]$. Then*

$$k(A, x + B) = \exp\left[-\gamma(\|(a-b)^T M(a-b)\|_2^2 - \|\xi_A\|_2^2)\right] \exp\left[-\gamma\|x - \xi_A\|_2^2\right],$$

where the vectors $a, b \in \mathbb{R}^{2d}$ and the matrix $M \in \mathbb{R}^{2d} \times \mathbb{R}^{2d}$ are defined as in Theorem 3.11.

Proof. Let $A, B \in X_c$, $x \in \mathbb{R}^d$ and k be the Gaussian kernel for convex sets. We compute

$$\begin{aligned} k(A, x + B) &= \exp\left[-\gamma\|\phi(A) - \phi(x + B)\|_{L_2(S_{d-1})}^2\right] \\ &= \exp\left[-\gamma\langle\phi(A) - \phi(x + B), \phi(A) - \phi(x + B)\rangle_{L_2(S_{d-1})}\right] \\ &= \exp\left[-\gamma\left(k_c(A, A) - 2k_c(A, x + B) + k_c(x + B, x + B)\right)\right] \\ &= \frac{\exp\left(-\gamma k_c(A, A)\right)}{\exp\left(-2\gamma k_c(A, B)\right)} \frac{\exp\left(-\gamma k_c(x + B, x + B)\right)}{\exp\left(-2\gamma x^T \xi_A\right)} \end{aligned}$$

where $\xi_A = k_c(A, \{e_j\})_{j=1,\dots,d} \in \mathbb{R}^d$. To obtain the last equality we used the computations made in the proof of Lemma 3.9.

1. Now let $B = \{0\}$. Then the equation above simplifies to

$$\begin{aligned} k(A, \{x\}) &= \exp\left(-\gamma k_c(A, A)\right) \exp\left[-\gamma\left(\|x\|_2^2 - 2x^T\xi_A\right)\right] \\ &= \exp\left[-\gamma\left(k_c(A, A) - \|\xi_A\|_2^2\right)\right] \exp\left(-\gamma\|x - \xi_A\|_2^2\right). \end{aligned}$$

2. Similarly, for $A, B \in X_I$, $A = \overset{d}{\underset{i=1}{\times}}[a_{-1i}, a_{+1i}]$ and $B = \overset{d}{\underset{i=1}{\times}}[-b_i, b_i]$ we get

$$\begin{aligned} k(A, x+B) &= \frac{\exp\left(-\gamma a^T M a\right)}{\exp\left(-2\gamma a^T M b\right)} \exp\left[-\gamma\left(\|x\|_2^2 + b^T M b - 2x^T \xi_A\right)\right] \\ &= \exp\left[-\gamma(\|(a-b)^T M (a-b)\|_2^2 - \|\xi_A\|_2^2)\right] \exp\left[-\gamma\|x - \xi_A\|_2^2\right], \end{aligned}$$

featuring the results of Theorem 3.11 and Lemma 3.13.

□

In the next example, the second part of this lemma is used to calculate the decision function for an example data set with predictive interaction of position and shape. This data set contains, like the data set in Example 3.14, two dimensional intervals of two different shapes (up to some random noise). One shape is an interval that spreads mainly along the x_1-axis, the other along the x_2-axis. Contrarily to the data set in Example 3.14, the label does not only depend on the orientation of the interval, but on the interaction of orientation and the position relative to the x_1-axis.

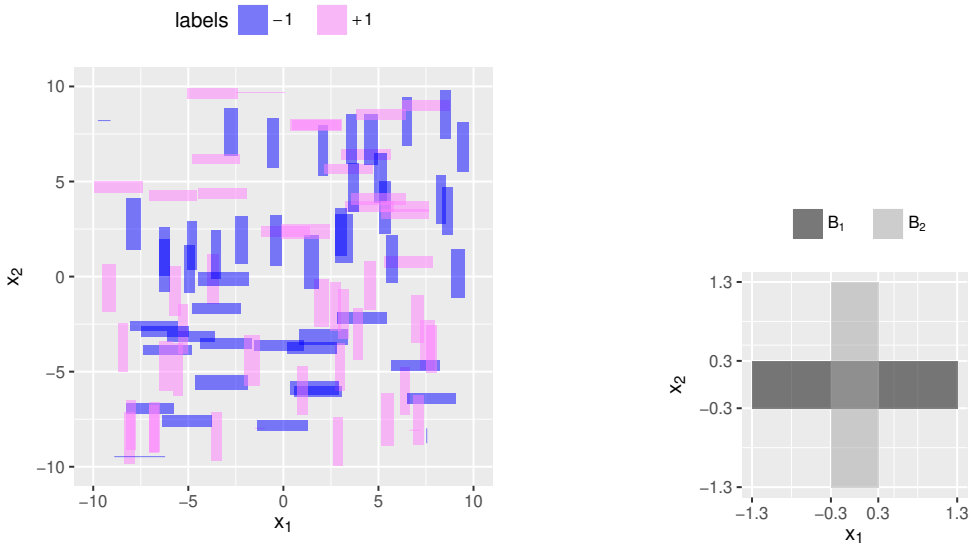
Example 3.23
Let (A, Y) be a random variable with $P(Y = -1) = P(Y = 1) = \frac{1}{2}$ and $A \in X_I$ with midpoint vector $M = \binom{M_1}{M_2} \sim U([-10, 10]^2)$, and length vector L. For the conditional distribution of the length L, given the label Y and the position of the midpoint M shall hold

$$L|Y, M \sim \begin{pmatrix} 1.5 - Y \\ 1.5 + Y \end{pmatrix} \mathbb{1}_{M_2 \geq 0} + \begin{pmatrix} 1.5 + Y \\ 1.5 - Y \end{pmatrix} \mathbb{1}_{M_2 < 0} + Exp(10)$$

where Exp(10) denotes a exponential distribution with rate parameter set to 10. Figure 9a shows 100 identically and independently drawn samples $(A_i, y_i) \sim (A, Y)$. For the conditional expectation of the interval lengths holds therefore

$$\begin{aligned} l_1 &= \mathbb{E}[L|Y = -1, M_2 \geq 0] = \mathbb{E}[L|Y = 1, M_2 < 0] = \begin{pmatrix} 2.5 \\ 0.5 \end{pmatrix} + \mathbb{E}[E] = \begin{pmatrix} 2.6 \\ 0.6 \end{pmatrix}, \\ l_2 &= \mathbb{E}[L|Y = 1, M_2 \geq 0] = \mathbb{E}[L|Y = -1, M_2 < 0] = \begin{pmatrix} 0.5 \\ 2.5 \end{pmatrix} + \mathbb{E}[E] = \begin{pmatrix} 0.6 \\ 2.6 \end{pmatrix}. \end{aligned}$$

For $j = 1, 2$ *let* $B_j = \frac{1}{2} \overset{2}{\underset{i=1}{\times}} [-l_{ji}, l_{ji}]$*, thus* B_1 *and* B_2 *are the expected interval sets centred at the origin.*



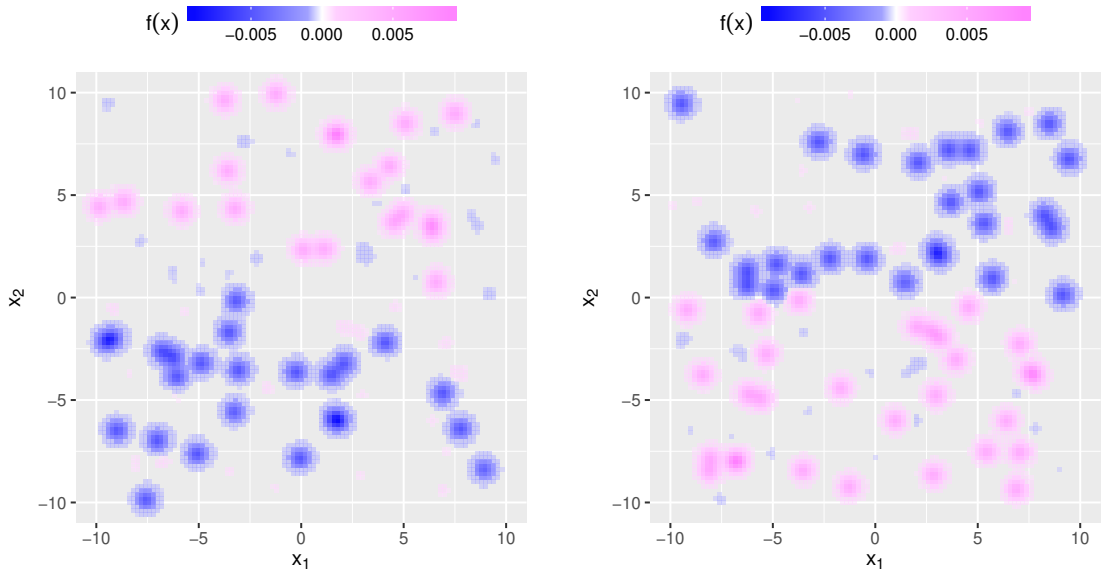
(a) 100 identically and independently drawn realisation of (A, Y).

(b) Expected shapes:
$B_1 = [-1.3, 1.3] \times [-0.3, 0.3]$
$B_2 = [-0.3, 0.3] \times [-1.3, 1.3]$.

Figure 9: Data set with predictive interaction of position and shape. Sets of shape B_1 have negative label in the half plane described $\{x \in \mathbb{R}^2 |\ x_2 < 0\}$, sets of shape B_2 have negative label in the half plane $\{x \in \mathbb{R}^2 |\ x_2 > 0\}$.

To understand how a decision function obtained by a SVM algorithm featuring the Gaussian kernel for convex sets behaves, we will look at restrictions to set of the form $x + B_1$ *and* $x + B_2$*. Figure 10 shows* $f|_{x+B_1}$ *and* $f|_{x+B_2}$ *for* f *being the minimiser of the risk without offset, the Gaussian kernel defined in Lemma 3.16 and parameters* λ *and* γ *set to* 1 *and* 5*, respectively.*

It seems that the classifier can distinguish between sets of differently orientated intervals. Nevertheless, at least for the present choice of parameters it does not generalise the position of the interval sets well. Hence the risk for the decision function is still greater than the Bayes risk. This minimal risk would be obtained by a function that separates the two half spaces $\{x \in \mathbb{R}^2 |\ x_2 \geq 0\}$ *and* $\{x \in \mathbb{R}^2 |\ x_2 < 0\}$.



(a) Decision function for sets $x + B_1$.

(b) Decision function for sets $x + B_2$.

Figure 10: Decision function obtained for sets of form $x + B$ given the position $x \in \mathbb{R}^d$ for two predictive shapes.

The problem arising in the last example seems to be due to the Gaussian kernel being only dependent on one shape parameter γ. This means a Support Vector Machine based on the Gaussian kernel for convex sets only generalises well for either the shape or the position of the intervals. To get good adaptation to both, one needed to modify the kernel further, for example by including a second shape parameter.

4. Decision Theoretical Approach to Classifying Convex Data

Whereas the construction of a kernel for convex sets seems to be unique to this work, decision theoretical approaches to generalise Support Vector Machines to interval data have already been discussed. See for example papers by Utkin, Chekh and Zhuk [19] and Wiencierz and Cattaneo [20]. The later study a minimax approach for Support Vector Regression with interval data in detail. In this section we discuss some approaches of classifying convex data using SVMs and compare their results to those obtained using a Support Vector Machine featuring a kernel for convex sets, as given in the previous section. Consider, like before, a given data set of form

$$\mathcal{D} = \{(A_i, y_i) |\ A_i \subseteq \mathbb{R}^d \text{ compact and convex, } y_i \in \{-1, 1\},\ i = 1, \dots, n\}.$$

Equivalently one can ask for

$$\mathcal{D} \subseteq X_c \times \{-1, 1\} \text{ with } |\mathcal{D}| = n < \infty,$$

where $X_c = \{A \subseteq \mathbb{R}^d \text{ compact and convex}\}$. The general task is to decide for a statistical model, in our case for a classification function, given the assumption that there is some "true" value a_i within A_i for all $i = 1, \dots, n$. These decision theoretical strategies can be split into two main approaches. The first one is referred to as "decision under risk". Here the decision maker assumes probabilities for the possible outcomes. The second one is referred to as "decision under uncertainty", here the probabilities are either unknown or do not exist. For further investigation of these concepts see for example an introduction to decision theory by Peterson [11].

When considering decision under risk, instead of the risk given in Definition 2.5 one tries to minimise the regularised Bayesian risk

$$\mathcal{R}_B(f) = \|f\|_{\mathcal{H}}^2 + \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{P_i}[\mathcal{L}(f(a_i), y_i)]$$

for a measurable function $f : \mathbb{R}^d \to \mathbb{R}$ within some Hilbert space of functions $\mathcal{H}$. Here one assumes some probability distribution P_i and some random variable $a_i \sim P_i$

for all $i = 1, \ldots, n$. P_i should be related to the convex sets observed. For example P_i might have support on A_i for all $i = 1, \ldots, n$. This is the case in the following example.

Example 4.1

Like in Example 2.10, let $\mathcal{H} = (\mathbb{R}^d)'$ and $\mathcal{L}$ be the hard margin loss. Furthermore, let P_i be a continuous probability distribution on $\mathbb{R}^d$ with support equal to A_i for all $i = 1, \ldots, n$.

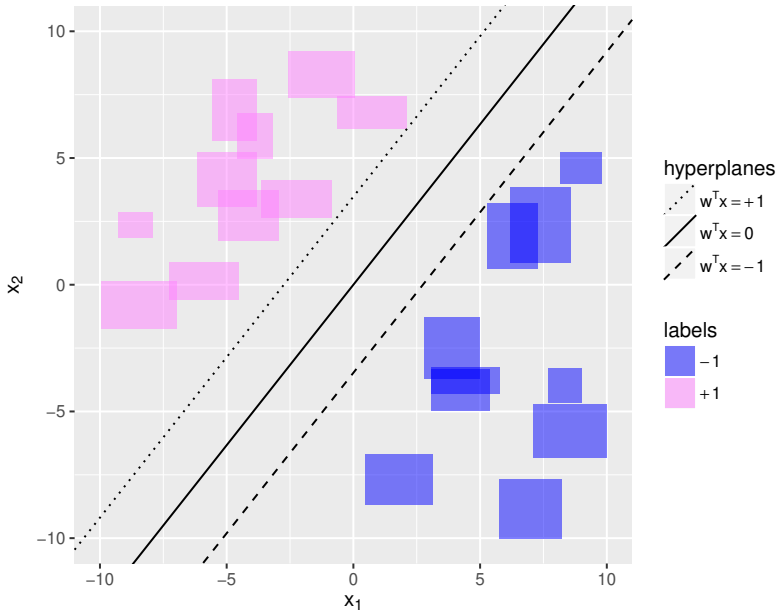


Figure 11: Decision function obtained under risk for the hard margin loss.

One computes for all $w \in \mathbb{R}^d$ and $i = 1, \ldots, n$

$$\mathbb{E}_{P_i}[\mathcal{L}(\langle w, a_i \rangle, y_i)] = \begin{cases} 0, & \text{if } y_i \langle w, a_i \rangle \geq 1 \text{ for almost all } a_i \in A_i \\ \infty, & \text{else.} \end{cases}$$

Hence $\mathcal{R}_B(f_w)$ is finite if and only if for all $i = 1, \ldots, n$ holds

$$P_i[y_i\langle w, a_i\rangle \geq 1] = 0 \;\Leftrightarrow\; \lambda_d(\{a_i \in A_i | y_i\langle w, a_i\rangle < 1\}) = 0.$$

Here does λ_d denote the d-dimensional Lebesgue measure on $\mathbb{R}^d$. The equivalence is due to P_i being continuous with support equal to A_i. Hence a minimiser of $\mathcal{R}_B$ exists if and only if there exists a $w \in \mathbb{R}^d$ such that all A_i's with label $y_i = +1$ lie within the half-plane $\{x \in \mathbb{R}^d | \langle w, x\rangle \geq 1\}$ and all A_i's with label $y_i = -1$ lie within the half-plane $\{x \in \mathbb{R}^d | \langle w, x\rangle \leq -1\}$. If this is the case for some $w \in \mathbb{R}^d$ the corresponding Bayesian risk becomes $\mathcal{R}_B(f) = \|f\|^2_{\mathcal{H}}$. This means minimising the risk is again equivalent to maximising the distance between the separating hyperplanes (compare to Example 2.10). Figure 11 shows the separating hyperplanes for an example data set consisting of interval data.

Optimal solutions for the regularised Bayesian risk generally exist, due to the expectation functional preserving convexity. This means $\mathcal{R}_B$ is still strictly convex and therefore yields an unique minimiser. Nevertheless, this minimising function might be hard to obtain as expectations needed to be calculated. Moreover, the outcome of this optimisation problem is, for other loss functions than the hard margin loss, strongly depended on the choice of the priory distributions $P_i, i = 1, \ldots, n$. One might not want to make this strong assumption. This leads to the approach based on decision under uncertainty, which we will discuss in detail.

4.1. Decision Under Uncertainty

In this setting we can not tell how likely a certain outcome $(a_1, \ldots, a_n) \in \bigtimes_{i=1}^{n} A_i$ is and do not want to make any assumptions. On way of dealing with this issue is to apply a decision rule to all possible outcomes and then look at the set of all actions obtained. In the context of classification, the decision rule corresponds to the classifier as discussed in Subsection 2.5. The set of possible actions is here the set of decision function obtained as the image of the classifier restricted to input sets of form

$$\{(a_i, y_i) |\ i = 1, \ldots, n\} \quad \text{with } (a_1, \ldots, a_n) \in \bigtimes_{i=1}^{n} A_i.$$

Then the input space $\mathbb{R}^d$ can be canonically split into three distinct parts. A set of input vectors for which all decision functions are strictly positive, a set where all decision functions are negative and the remaining vectors for which both, positive and negative values, exist. Figure 12 shows how these sets can look like in the case of linear separation without offset.

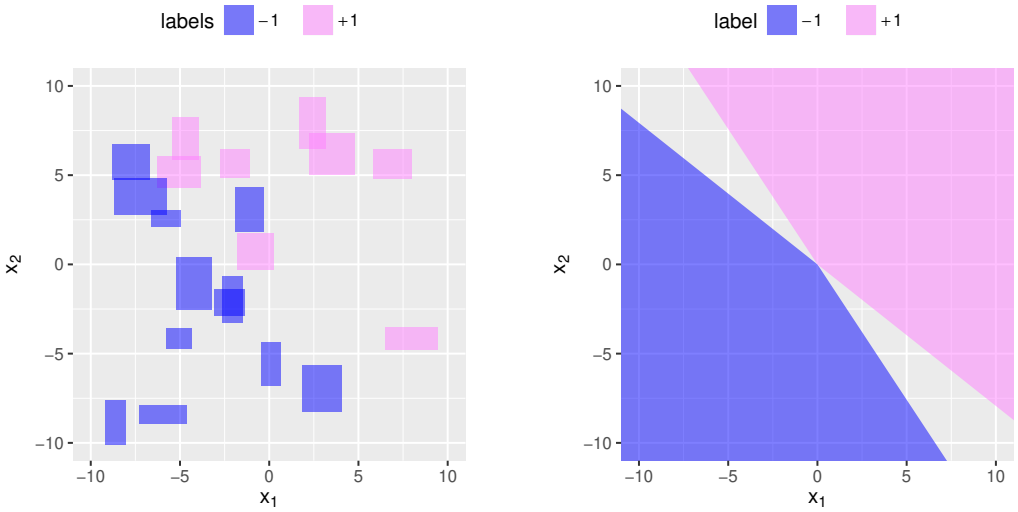


Figure 12: The figure on the right-hand side indicates areas where all decision functions (obtained using a SVM classifier on the interval data set shown in the left plot) have equal sign.

The area for which decision functions with both, positive and negative values, exist can in general be very large. Furthermore, those areas are hard to obtain, as every possible combination of $(a_1, \dots a_n)$ with $a_i \in A_i$ has to be taken into account. Even in the case of linear separation, an optimiser has to be found for every combination of extreme points. This is even for small data sets not feasible. To avoid those difficulties, one can only consider the minimal and the maximal risk for every decision function $f \in \mathcal{H}$ given data $\{(a_i, y_i) |\ i = 1, \dots, n\}$ with $a_i \in A_i$ for all $i = 1, \dots, n$, instead.

Definition 4.2 (Minimal and maximal risk)
Let $\mathcal{D} = \{(A_i, y_i)|\ i = 1, \dots, n\} \subseteq X_c \times \{-1, 1\}$ be an input data set consisting of convex data. The functional

$$\begin{aligned} \mathcal{R}_{min} :&\mathcal{H} \to \mathbb{R} \\ &f \mapsto \lambda \|f\|_{\mathcal{H}}^2 + \frac{1}{n} \sum_{i=1}^{n} \inf_{a_i \in A_i} \mathcal{L}(f(a_i), y_i) \end{aligned} \tag{17}$$

is called minimal SVM risk functional. Similarly one defines the maximal SVM risk functional as

$$\begin{aligned} \mathcal{R}_{max} :&\mathcal{H} \to \mathbb{R} \\ &f \mapsto \lambda \|f\|_{\mathcal{H}}^2 + \frac{1}{n} \sum_{i=1}^{n} \sup_{a_i \in A_i} \mathcal{L}(f(a_i), y_i). \end{aligned} \tag{18}$$

Remark 4.3
Alternatively, an additional offset can be added. In this case the subsequent results could be modified just like in Section 2. For the sake of a clear presentation, those modifications are not made here, though.

A common strategy of decision making under uncertainty is to minimise either the minimal or the maximal risk. The next theorem and the subsequent example demonstrate that the two functionals given in Definition 4.2 do not behave the same way. Whereas the maximal risk is still convex and therefore yields a minimiser, this property is not guaranteed for the minimal risk.

Theorem 4.4 (Unique minimiser exist for $\mathcal{R}_{max}$)
Let $\mathcal{L}$ be a finite and convex loss. Let $\mathcal{H} \subseteq \{f : \mathbb{R}^d \to \mathbb{R}\}$ be a Hilbert space such that the linear maps $\delta_x : \mathcal{H} \to \mathbb{R},\ f \mapsto f(x)$ are continuous for all $x \in \mathbb{R}^d$.
Then $\mathcal{R}_{max}$ (defined in Equation 18) has a unique minimiser.

Proof. Similar to the proof of 2.6 we will see that $\mathcal{R}_{max}$ is convex and coercive. Unlike the risk $\mathcal{R}$ given in Equation 3, $\mathcal{R}_{max}$ is not necessarily continuous. Nevertheless it is lower-semicontinuous, which is sufficient for concluding that a minimiser exists (see Theorem A.10).

We have seen in the proof of Theorem 2.6 that

$$\begin{aligned}\mathcal{H} &\to \mathbb{R}_0^+ \\ f &\mapsto \mathcal{L}(f(a), y)\end{aligned}$$

is convex and continuous for all $a \in \mathbb{R}^d$ and $y \in \{-1, +1\}$. Hence

$$\begin{aligned}\mathcal{H} &\to \mathbb{R}_0^+ \\ f &\mapsto \sup_{a \in A} \mathcal{L}(f(a), y)\end{aligned}$$

is convex and lower-semicontinuous (see Lemma A.8) for every set $A \subseteq \mathbb{R}^d$. This implies that

$$\begin{aligned}\mathcal{H} &\to \mathbb{R}_0^+ \\ f &\mapsto \frac{1}{n} \sum_{i=1}^{n} \sup_{a_i \in A_i} \mathcal{L}(f(a_i), y_i)\end{aligned}$$

is convex and lower-semicontinuous as a positive multiple and a sum of convex and lower-semicontinuous functions. Thus $\mathcal{R}_{max}$ is strictly convex and lower-semicontinuous as $f \to \|f\|_{\mathcal{H}}^2$ is strictly convex and continuous. Since $\mathcal{R}_{max}$ is clearly coercive as well, Mazur's Theorem A.10 states that it has a minimiser, which is unique due to $\mathcal{R}_{max}$ being strictly convex. □

Example 4.5 (Minimiser are not unique for $\mathcal{R}_{min}$)
Let $\lambda = 1$, $X = \mathbb{R}^2$, $\mathcal{H} = (\mathbb{R}^2)'$ and $\mathcal{L}$ be the hinge loss. Let

$$\mathcal{D} = \{(A, y)\} \quad \textit{with } A = [-1, 1] \times \{0\} \textit{ and } \ y = 1.$$

Then we have for all $f \in \mathcal{H} = (\mathbb{R}^2)'$ and equivalently (see Example 2.10) for all $w = \binom{w_1}{w_2} \in \mathbb{R}^2$ and $f_w = \langle w, \cdot \rangle$:

$$\begin{aligned}\mathcal{R}_{min}(f_w) &= \|f_w\|_{\mathcal{H}}^2 + \inf_{x \in A} \mathcal{L}(f_w(x), 1) \\ &= \|w\|_2^2 + \inf_{x \in A} \max\{0, 1 - \langle w, x \rangle\},\end{aligned}$$

and therefore

$$\mathcal{R}_{min}(f_w) = w_1^2 + w_2^2 + \inf_{x_1 \in [-1,1]} \max\{0, 1 - w_1 x_1\}.$$

- *For* $w_1 \in \mathbb{R}\backslash]-1,1[$ *and for* $x_1 = \frac{1}{w_1} \in [-1,1]$ *it holds* $\max\{0, 1 - w_1 x_1\} = 0$ *which implies*

  $$0 \leq \inf_{x_1 \in [-1,1]} \max\{0, 1 - w_1 x_1\} \leq 0,$$

  which implies $\mathcal{R}_{min}(f_w) = w_1^2 + w_2^2$ *for all* $w = \binom{w_1}{w_2} \in \mathbb{R}\backslash]-1,1[\times\mathbb{R}$.

- *For* $w_1 \in]-1,1[$ *we have* $1 - w_1 x_1 \geq 1 - |w_1| > 0$ *and for* $x_1 = \operatorname{sign}(w_1) \in [-1,1]$ *holds* $1 - w_1 x_1 = 1 - |w_1|$ *hence we conclude*

  $$\inf_{x_1 \in [-1,1]} \max\{0, 1 - w_1 x_1\} = 1 - |w - 1|.$$

  This implies $\mathcal{R}_{min}(f_w) = w_1^2 + w_2^2 + 1 - |w_1|$ *for all* $w = \binom{w_1}{w_2} \in]-1,1[\times\mathbb{R}$.

Putting these cases together one concludes

$$\mathcal{R}_{min}(f_w) = w_1^2 + w_2^2 + (1 - |w_1|)\mathbb{1}_{[-1,1]}(w_1) \quad \forall w = \binom{w_1}{w_2} \in \mathbb{R}.$$

Hence the minimisation with respect to w_2 *can be done independently of the minimisation with respect to* w_1*. Therefore, the minimisation problem simplifies to*

$$\begin{aligned} &\textit{minimize} \quad \mathcal{R}_1(w_1) := w_1^2 + (1 - |w_1|)\mathbb{1}_{[-1,1]}(w_1) \\ &\textit{with respect to} \quad w_1 \in \mathbb{R}. \end{aligned}$$

Since $\mathcal{R}_1$ *is differentiable on* $\mathbb{R} \setminus \{-1, 0, 1\}$ *one calculates*

$$\mathcal{R}_1'(w_1) = 2w_1 - \begin{cases} \operatorname{sign}(w_1) & \textit{if } w_1 \in]-1,1[\\ 0 & \textit{else} \end{cases}$$

for all $w_1 \notin \{-1, 0, 1\}$.

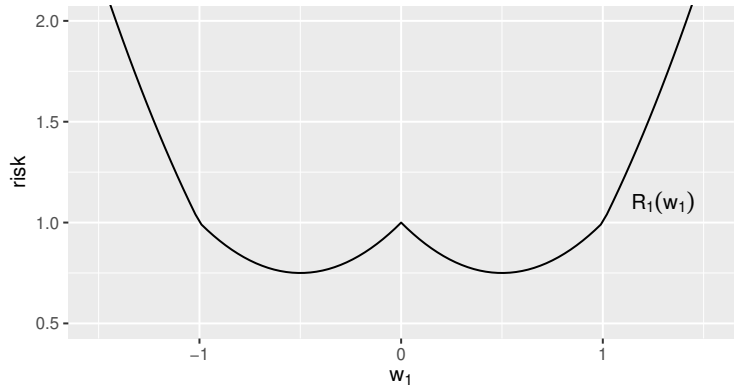


Figure 13: Objective function $\mathcal{R}_1$.

Hence we see for all $w_1 \notin \{-1, 0, 1\}$

$$\begin{aligned} \mathcal{R}_1'(w_1) = 0 &\Leftrightarrow 2w_1 - \text{sign}(w_1) = 0 \\ &\Leftrightarrow w_1 \in \left\{-\frac{1}{2}, \frac{1}{2}\right\} \end{aligned}$$

and $\mathcal{R}_1(-\frac{1}{2}) = \mathcal{R}_1(\frac{1}{2}) = \frac{1}{4} + \frac{1}{2} = \frac{3}{4}$.

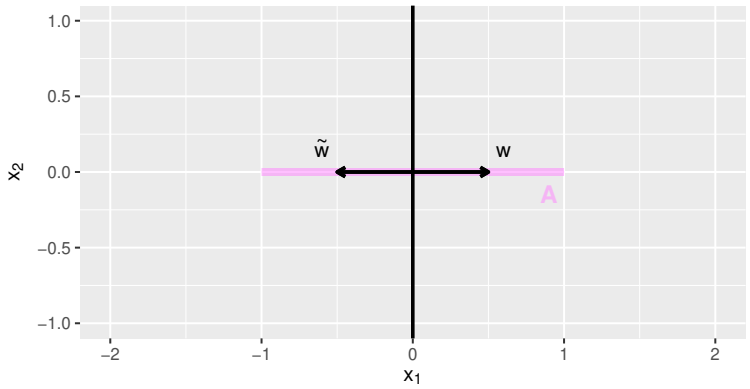


Figure 14: Minimiser $\tilde{w}$ and w of $\mathcal{R}_{min}$.

Since we have additionally $\mathcal{R}_1(-1) = \mathcal{R}_1(0) = \mathcal{R}_1(1) = 1 > 0.75$*, we conclude that both* $f_{\tilde{w}}$ *and* f_w *with* $\tilde{w} = \binom{-0.5}{0}$ *and* $w = \binom{0.5}{0}$ *are minimiser for* $\mathcal{R}_{min}$*, hence solutions are not unique.*

Minimising the maximal risk $\mathcal{R}_{max}$ can be interpreted as an insurance against the worst case. Having seen that $\mathcal{R}_{max}$ yields a unique minimiser we can define:

Definition 4.6 (Minimax Support Vector Machine)
The optimisation problem

$$\begin{aligned} &\textit{minimise} \quad \mathcal{R}_{max}(f) \\ &\textit{with respect to } f \in \mathcal{H} \end{aligned}$$

is called Minimax Support Vector Machine.

Even though solutions to the Minimax SVM exist in general, numerical optimisation of $\mathcal{R}_{max}$ can still be difficult. One might hope that the kernel trick developed in Subsection 2.3 could be adapted. This was possible when minimisation and maximisation could be exchange. The optimisation problem would then become

$$\begin{aligned} &\text{maximise} \quad \min_{f\in\mathcal{H}}\{\lambda\|f\|_{\mathcal{H}}^2 + \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f(a_i), y_i)\} \\ &\text{with respect to } a_i \in A_i \ \text{ for all } i = 1, \dots, n. \end{aligned}$$

Hence the kernel trick could be used to simplify the inner minimisation. Nevertheless, the following example shows that this optimisation problem is in general not equivalent to minimising the maximal risk.

Example 4.7
Consider an input data set

$$\mathcal{D} = \{(A_1, 1), (A_2, -1)\}$$

with $A_1 = \{\left(\begin{smallmatrix}0\\0\end{smallmatrix}\right)\}$ *and* $A_2 = [-1, 1] \times \{1\}$. *Moreover, let* $\mathcal{L}$ *be the hinge loss and* $\mathcal{H}$ *be the reproducing kernel Hilbert space defined via the feature map*

$$\begin{aligned} \phi : \mathbb{R}^2 &\to \mathbb{R}^3 \\ \left(\begin{smallmatrix}x_1\\x_2\end{smallmatrix}\right) &\mapsto \begin{pmatrix} x_2\cos(x_1\pi) \\ x_2\sin(x_1\pi) \\ x_2+1 \end{pmatrix}. \end{aligned}$$

The tuning parameter is set to $\lambda = \frac{1}{8}$. *Figure 15 shows the image of both input sets in the feature space* $\mathbb{R}^3$. *These two sets are to be separated by a hyperplane containing the origin.*

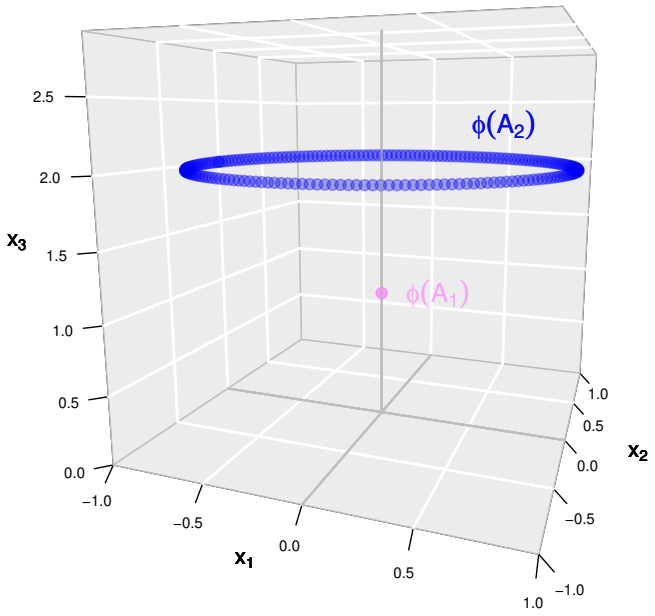


Figure 15: Input sets $\phi(A_1)$ and $\phi(A_2)$ with corresponding labels $y_1 = +1$ indicated in pink and $y_2 = -1$ in blue.

Minimisation of $\mathcal{R}_{max}$, the maximal risk *For* $w \in \mathbb{R}^3$ *denote by* $f_w = \langle w, \phi(\cdot)\rangle$ *the corresponding function in* $\mathcal{H}$. *Then we have for every* $f \in \mathcal{H}$ *such a representation; hence the optimisation problem can be equivalently formulated as*

$$\textit{minimise } \mathcal{R}_{max}(w) := \mathcal{R}_{max}(f_w) \quad \textit{with respect to } w \in \mathbb{R}^3.$$

Assume $w = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \in \mathbb{R}^3$ *is a minimiser of* $\mathcal{R}_{max}$ *and let* $\tilde{w} = \begin{pmatrix} -w_1 \\ -w_2 \\ w_3 \end{pmatrix}$. *Then one computes*

$$\begin{aligned}
\sup_{a_2 \in A_2} \mathcal{L}(f_{\tilde{w}}(a_2), -1) &= \sup_{a \in [-1,1]} \mathcal{L}(\langle \tilde{w}, \begin{pmatrix} \cos(a\pi) \\ \sin(a\pi) \\ 2 \end{pmatrix} \rangle, -1) \\
&= \sup_{a \in [-1,1]} \mathcal{L}(\langle w, \begin{pmatrix} -\cos(a\pi) \\ -\sin(a\pi) \\ 2 \end{pmatrix} \rangle, -1) \\
&= \sup_{a \in [-1,1]} \mathcal{L}(\langle w, \begin{pmatrix} \cos((a+1)\pi) \\ \sin((a+1)\pi) \\ 2 \end{pmatrix} \rangle, -1) \\
&= \sup_{a \in [0,2]} \mathcal{L}(\langle w, \begin{pmatrix} \cos(a\pi) \\ \sin(a\pi) \\ 2 \end{pmatrix} \rangle, -1) \\
&= \sup_{a \in [-1,1]} \mathcal{L}(\langle w, \begin{pmatrix} \cos(a\pi) \\ \sin(a\pi) \\ 2 \end{pmatrix} \rangle, -1),
\end{aligned}$$

since we have

$$\{ \begin{pmatrix} \cos(a\pi) \\ \sin(a\pi) \\ 2 \end{pmatrix} |\ a \in [0,2] \} = \{ \begin{pmatrix} \cos(a\pi) \\ \sin(a\pi) \\ 2 \end{pmatrix} |\ a \in [-1,1] \}.$$

Hence one concludes

$$\begin{aligned}
\mathcal{R}_{max}(\tilde{w}) &= \frac{1}{8}\|\tilde{w}\|_2^2 + \frac{1}{2}\left[\mathcal{L}(\langle \tilde{w}, \phi\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}\right) \rangle, 1) + \sup_{a \in [-1,1]} \mathcal{L}(\langle \tilde{w}, \begin{pmatrix} \cos(a\pi) \\ \sin(a\pi) \\ 2 \end{pmatrix} \rangle, -1) \right] \\
&= \frac{1}{8}\|w\|_2^2 + \frac{1}{2}\left[\mathcal{L}(w_3, 1) + \sup_{a \in [-1,1]} \mathcal{L}(\langle w, \begin{pmatrix} \cos(a\pi) \\ \sin(a\pi) \\ 2 \end{pmatrix} \rangle, -1) \right] \\
&= \mathcal{R}_{max}(w).
\end{aligned}$$

Theorem 4.4 shows that the minimiser of $\mathcal{R}_{max}$ *is unique. Hence* $\mathcal{R}_{max}(\tilde{w}) = \mathcal{R}_{max}(w)$ *for a minimiser* w *implies* $w = \tilde{w}$. *This means any minimiser* $w \in \mathbb{R}^3$ *must have form* $w = \begin{pmatrix} 0 \\ 0 \\ w_3 \end{pmatrix}$ *for some* $w_3 \in \mathbb{R}$. *It is therefore sufficient to consider the following reduced minimisation problem.*

$$\textit{Minimise } \mathcal{R}_3(w_3) := \mathcal{R}_{max}(\begin{pmatrix} 0 \\ 0 \\ w_3 \end{pmatrix}) \quad \textit{with respect to } w_3 \in \mathbb{R}.$$

This risk can be simplified to

$$\begin{aligned}\mathcal{R}_3(w_3) &= \frac{1}{8}w_3^2 + \frac{1}{2}\left[\mathcal{L}(w_3, 1) + \mathcal{L}(2w_3, -1))\right] \\ &= \frac{1}{8}w_3^2 + \frac{1}{2}\left[\max\{0, 1 - w_3\} + \max\{0, 1 + 2w_3\}\right]\end{aligned}$$

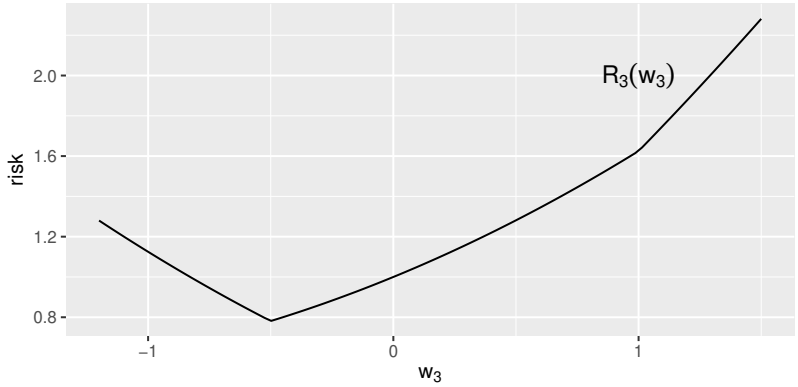


Figure 16: Objective function $\mathcal{R}_3$ yielding a unique minimiser.

$\mathcal{R}_3$ is differentiable for all $w_3 \notin \{-\frac{1}{2}, 1\}$ with

$$\mathcal{R}_3'(w_3) = \frac{1}{4}w_3 + \frac{1}{2}\begin{cases} -1 & \text{if } w_3 < -\frac{1}{2} \\ 1 & \text{if } -\frac{1}{2} < w_3 < 1 \\ 2 & \text{if } 1 < w_3. \end{cases}$$

Thus we observe

$$\begin{aligned} \mathcal{R}_3'(w_3) &= \frac{1}{4}w_3 - \frac{1}{2} < 0 && \text{for all } w_3 < -\frac{1}{2} \\ \mathcal{R}_3'(w_3) &\geq \frac{1}{4}w_3 + \frac{1}{2} > -\frac{1}{8} + \frac{1}{2} > 0 && \text{for all } w_3 > -\frac{1}{2} \end{aligned}$$

This means a minimiser can only be found at points w_3 where $\mathcal{R}_3$ is not differentiable at. Hence we conclude by computing $\mathcal{R}_3(-\frac{1}{2}) = \frac{25}{32}$ and $\mathcal{R}_3(1) = \frac{13}{8} > \mathcal{R}_3(-\frac{1}{2})$ that $w = \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{2} \end{pmatrix}$ is the unique minimiser of $\mathcal{R}_{max}$.

The SVM solution in the least favourable case *For all $a \in [-1,1]$ let*

$$\begin{aligned}\mathcal{R}_a : \mathbb{R}^3 &\to \mathbb{R} \\ w &\mapsto \frac{1}{8}\|w\|_2^2 + \frac{1}{2}\left[\mathcal{L}(\langle w, \begin{pmatrix}0\\0\\1\end{pmatrix}\rangle, 1) + \mathcal{L}(\langle w, \begin{pmatrix}\cos(a\pi)\\ \sin(a\pi)\\ 2\end{pmatrix}\rangle, -1)\right]\end{aligned}$$

be the corresponding risk. The unique minimiser $w = \begin{pmatrix}0\\0\\-\frac{1}{2}\end{pmatrix}$ of $\mathcal{R}_{max}$ is for no $a \in [-1,1]$ the minimiser of $\mathcal{R}_a$. To see this let $w_a = \begin{pmatrix}-\cos(a\pi)\\ -\sin(a\pi)\\ 0\end{pmatrix}$ for all $a \in [-1,1]$. We compute

$$\begin{aligned}\mathcal{R}_a(w_a) &= \frac{1}{8}\|w_a\|_2^2 + \frac{1}{2}\left[\mathcal{L}(\langle w_a, \begin{pmatrix}0\\0\\1\end{pmatrix}\rangle, 1) + \mathcal{L}(\langle w_a, \begin{pmatrix}\cos(a\pi)\\ \sin(a\pi)\\ 2\end{pmatrix}\rangle, -1)\right] \\ &= \frac{1}{8} + \frac{1}{2}\left[\mathcal{L}(0,1) + \mathcal{L}(-\cos^2(a\pi) - \sin^2(a\pi), -1)\right] \\ &= \frac{1}{8} + \frac{1}{2}(1+0) \\ &= \frac{5}{8}.\end{aligned}$$

This shows that $\mathcal{R}_a(w_a) = \frac{20}{32}$ is strictly smaller than

$$\begin{aligned}\mathcal{R}_a(w) &= \frac{1}{8}\|w\|_2^2 + \frac{1}{2}\left[\mathcal{L}(\langle w, \begin{pmatrix}0\\0\\1\end{pmatrix}\rangle, 1) + \mathcal{L}(\langle w, \begin{pmatrix}\cos(a\pi)\\ \sin(a\pi)\\ 2\end{pmatrix}\rangle, -1)\right] \\ &= \frac{1}{32} + \frac{1}{2}\left[\mathcal{L}(-\frac{1}{2}, 1) + \mathcal{L}(-1,-1)\right] \\ &= \frac{1}{32} + \frac{3}{4} \\ &= \frac{25}{32}.\end{aligned}$$

This means the unique minimiser of $\mathcal{R}_{max}$ can not be found by considering the optimisation problem

$$\textit{maximise} \min_{w \in \mathbb{R}^3} \mathcal{R}_a(w) \quad \textit{with respect to } a \in [-1,1].$$

Example 4.7 shows that one can not apply a procedure like the 'kernel trick' described in Subsection 2.3 in order to obtain a non-linear classifier. Nevertheless,

the optimisation problem

$$\begin{aligned} &\text{maximise} \quad \min_{f\in\mathcal{H}}\{\lambda\|f\|_{\mathcal{H}}^2 + \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f(a_i), y_i)\} \\ &\text{with respect to } a_i \in A_i \ \text{ for all } i = 1,\ldots,n. \end{aligned}$$

yields under certain circumstances, which needed to be studied in more detail, a maximiser $(a_1^*,\ldots,a_n^*) \in \times_{i=1}^n A_i$ and a corresponding minimiser $f^* \in \mathcal{H}$. This optimiser $f^* \in \mathcal{H}$ can be seen as an alternative classification function to the minimiser of the maximal risk $\mathcal{R}_{max}$. An advantage is that then $\mathcal{H}$ can be defined as a reproducing kernel Hilbert space associated with some kernel. Hence the optimisation problem can be simplified; especially when the maximum is attained at the boundary of the A_i's for all $i = 1,\ldots,n$. Examples of Support Vector Machines for interval valued training data, where the optimisation problem above simplifies further, are considered by Utkin, Chekh and Zhuk [19, page 295-303].

4.2. The Linear Minimax Support Vector Machine

We have seen in the previous subsection that the 'kernel trick' can not be used to construct function spaces for the Minimax Support Vector Machine. Moreover, numerical optimisation of $\mathcal{R}_{max}$ is in general expensive, as evaluating $\mathcal{R}_{max}(f)$ for a function $f \in \mathcal{H}$ involves itself n maximisation problems, where n denotes the number of input sets. These maximisation problems simplify when we consider linear separation. That is we choose $\mathcal{H} = (\mathbb{R}^d)'$ and might add an optional offset $b \in \mathbb{R}$. One can show that in this case the maxima on the A_i's can always be found at their extreme points. These are defined as follows.

Definition 4.8 (Extreme points)
For every subset $C \subseteq \mathbb{R}^d$ define its extreme points $\mathcal{E}(C)$ to be all points $x \in C$ that cannot be written as a convex combination of points in $C \setminus \{x\}$. That is

$$x = ty + (1-t)z$$

for some $t \in]0,1[$ implies $y = x$ and $z = x$.

Corollary 4.9
Let $A \subseteq \mathbb{R}^d$ be compact and convex and $y \in \{-1, 1\}$. Furthermore consider a convex loss function $\mathcal{L}$ and an optional offset $b \in \mathbb{R}$. Let

$$\begin{aligned} \phi : A &\to \mathbb{R} \\ a &\mapsto \mathcal{L}(\langle w, a\rangle + b, y). \end{aligned}$$

Then ϕ attains its maximum at an extreme point of A.

Proof. Bauers Maximum Principle (see [1, page 298]) states that every upper semi-continuous function on a compact and convex subset has a maximum that is an extreme point. Since ϕ is convex, as a composition of a linear and a convex function, and continuous (compare to the proof of Theorem 2.6), this corollary is an immediate consequence [2, page 136]. □

We observe that the extreme points of a convex polygon are precisely its corners. Hence the last corollary in particular shows that, in the case of polygons as input data, the maxima can be found in the corners of these sets. Since we consider a finite input set and every polygon has a finite number of extreme points, we are left with a simplified optimisation problem:

$$\begin{aligned} &\text{minimise} \quad \max_{a_i \in \mathcal{E}(A_i)} \lambda \|w\|_2^2 + \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\langle a_i, w\rangle, y_i) \\ &\text{with respect to } w \in \mathbb{R}^d. \end{aligned} \tag{19}$$

This is a so called 'convex finite min-max problem'; hence belongs to a class of optimisation problems for which numerical solving procedures are proposed. See for example the incremental method described by Gaudioso, Giallombardo and Miglionico [5] as well as the regularisation method in case of a differentiable loss presented by Gigola and Gomez [6]. However, problem 19 is not discussed in detail here as we will focus on finding a solution to the linear Minimax SVM when the input set consists of d-dimensional intervals and the loss function is convex and monotonic. In this case we do not only know that the maximum is attained in one of the corners, we also know in which one.

Corollary 4.10
Let $\mathcal{D} \subseteq X_I \times \{-1, 1\}$ with $n = |\mathcal{D}| < \infty$, hence we consider interval-valued input sets. For $A = \bigtimes_{i=1}^{d} [a_{-1i}, a_{+1i}]$, $s \in \{-1, +1\}^d$ and $y \in \{-1, +1\}$ define $a_{s,A,y} = (a_{ys_ii})_{i=1,\dots,d}^T$. Moreover, let $\mathcal{L}$ be a convex and monotonic loss function. Then $w \in \mathbb{R}^d$ is a minimiser of

$$\mathcal{R}_{max}(w) = \lambda \|w\|_2^2 + \frac{1}{n} \sum_{(A,y)\in\mathcal{D}} \sup_{a\in A} \mathcal{L}(\langle a, w\rangle, y)$$

if and only if it is a solution to the constrained problem

$$\begin{aligned} &\textit{minimise} \quad \lambda \|w\|_2^2 + \frac{1}{n} \sum_{(A,y)\in\mathcal{D}} \mathcal{L}(\langle a_{s,A,y}, w\rangle, y) \\ &\textit{with respect to } w \in \mathbb{R}_s^d \end{aligned}$$

where $\mathbb{R}_s^d = \{x \in \mathbb{R}^d | x_i s_i > 0,\ \forall i = 1, \dots, d\}$ and $s \in \{-1, +1\}^d$ being a minimiser of

$$\begin{aligned} \{-1, +1\}^d &\to \mathbb{R} \\ s &\mapsto \inf_{w\in\mathbb{R}_s^d} \lambda \|w\|_2^2 + \frac{1}{n} \sum_{(A,y)\in\mathcal{D}} \mathcal{L}(\langle a_{s,A,y}, w\rangle, y). \end{aligned}$$

Proof. For all $w \in R$ consider the linear functional

$$\begin{aligned} A &\to \mathbb{R} \\ a &\mapsto \langle a, w\rangle = \sum_{i=1}^{d} a_i w_i, \end{aligned}$$

which attains its maximum at $a_{s,A,+1}$ and its minimum at $a_{s,A,-1}$ where $s = (sign(w_i))_{i=1,\dots,d}$. Since $\mathcal{L}$ is monotonic this implies

$$\sup_{a\in A} \mathcal{L}(\langle a, w\rangle, y) = \mathcal{L}(\langle a_{s,A,y}, w\rangle y),$$

where $s = (sign(w_i))_{i=1,\dots,d}$ for all $w \in \mathbb{R}^d$.

Hence we conclude

$$\mathcal{R}_{max}(w) = \lambda\|w\|_2^2 + \frac{1}{n}\sum_{(A,y)\in\mathcal{D}} \mathcal{L}(\langle a_{s,A,y}, w\rangle y)$$

for all $w \in \mathbb{R}_s^d$ and $s \in \{-1,+1\}^d$. □

Corollary 4.10 shows that the minimisation problem can be split into 2^d constrained minimisation problems. Hence the number of numerical minimisations, which need to be performed is finite and only depends on the dimension of the input space d not on the number of input sets n. This result is used to calculate the linear separating hyperplanes plotted in Figure 17.

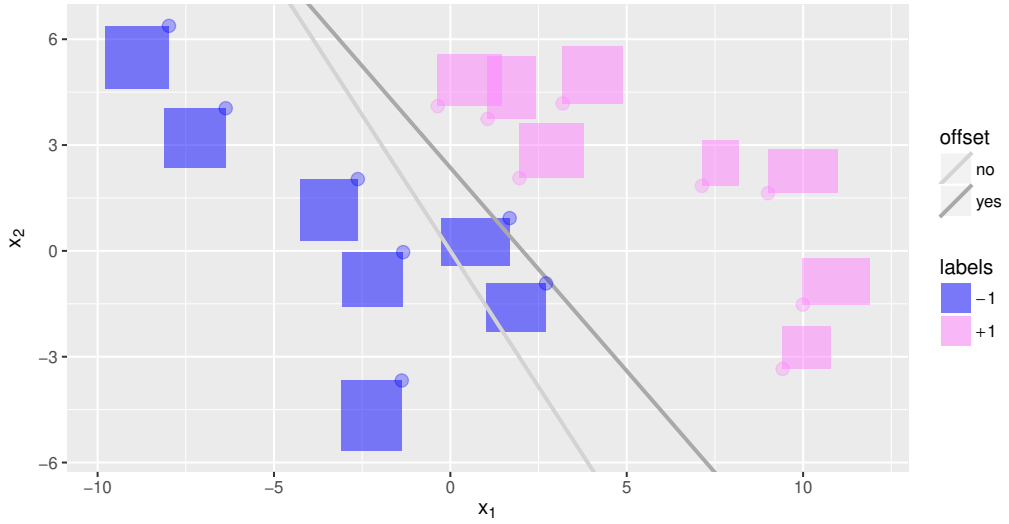


Figure 17: Linear separating functions obtained using the hinge loss and a minimax approach; dots in the corners of the intervals indicate critical points for the optimisation.

4.3. Comparision to the Kernel Based Approach

In this subsection we are comparing the results of the minimax approach discussed in this section and the kernel based approach given in Section 3. One clear advantage of the kernel based approach is that both linear and non-linear separation can easily be achieved. Contrarily, only linear separation seems to be numerically feasible

when the maximal risk is considered as an objective function. Hence a comparison of both approaches can only be done in the case of linear separation.

Figure 18 shows the separating hyperplane obtained by the two approaches discussed. The separating hyperplane for the kernel based approach is, as before in Section 3, the restriction to point sets. On this example data set the decision functions differ noticeable, hence predictions for may not agree.

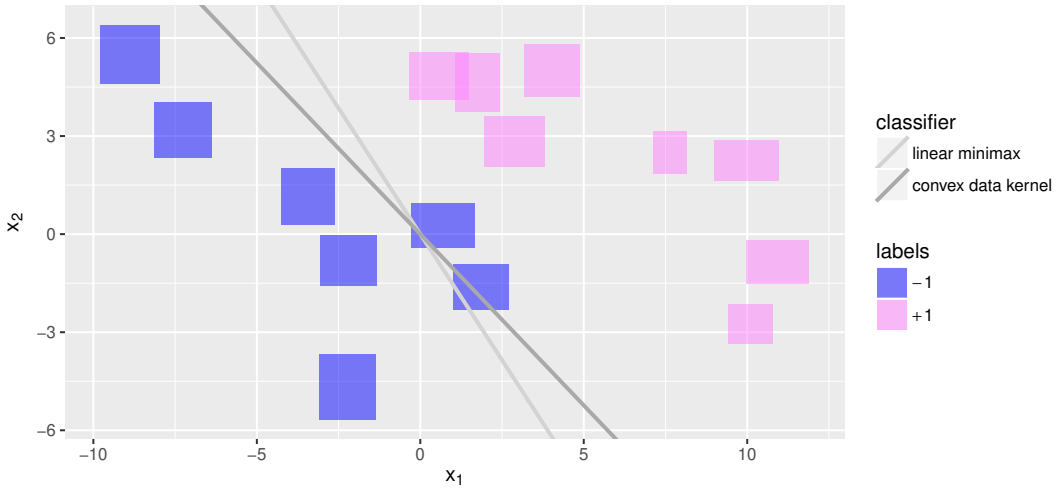


Figure 18: Separating hyperplane obtained by the minimax approach compared to the separating hyperplane for point sets obtained via the linear kernel for convex sets.

Whereas the separating hyperplane obtained by a minimax approach can be interpreted in a straightforward manner, which is close to the interpretation of decision functions for point sets, the decision function obtained by the kernel based approach is not so uncomplicated to interpret. Nevertheless, when classifying interval data by a linear decision function the effect on midpoints and lengths of the intervals can be analysed.

Hence to compare both approaches it might be convenient to analyse their performance. For Support Vector Machines the label of an input vector is usually predicted as the sign of the decision function evaluate at this input vector. Hence this procedure gives predictions for the kernel based approach as well. Predictions for the minimax classifier are not as canonical as the ones for the kernel based ap-

proach. In this case we are going to predict the label by minimising the maximal loss given a decision function f. For all $A \in X_c$ is that the minimiser of

$$\begin{aligned} \{-1, 1\} &\to \mathbb{R} \\ y &\mapsto \sup_{a \in A} \mathcal{L}(f(x), y). \end{aligned}$$

The next example shows an interval valued data set for which the SVM classifier based on the linear kernel for convex sets preforms clearly better.

Example 4.11
Let (A, Y) be a random variable with $P(Y = -1) = P(Y = 1) = \frac{1}{2}$ and

$$A = [M_1 - 0.5, M_1 + 0.5] \times [M_2 - E, M_2 + E] \in X_I,$$

with $M_1 \sim U[-30, 30]$ and $E \sim |\mathcal{N}(10, 1)|$. The conditional distribution of M_2, the midpoint in the second direction, given the label Y is defined as

$$M_2|Y \sim \mathcal{N}(-4Y, 1).$$

The tuning parameter λ is set to 1. Figure 19 shows a sample of 40 independently and identically drawn sets according to the distribution of (A, Y).

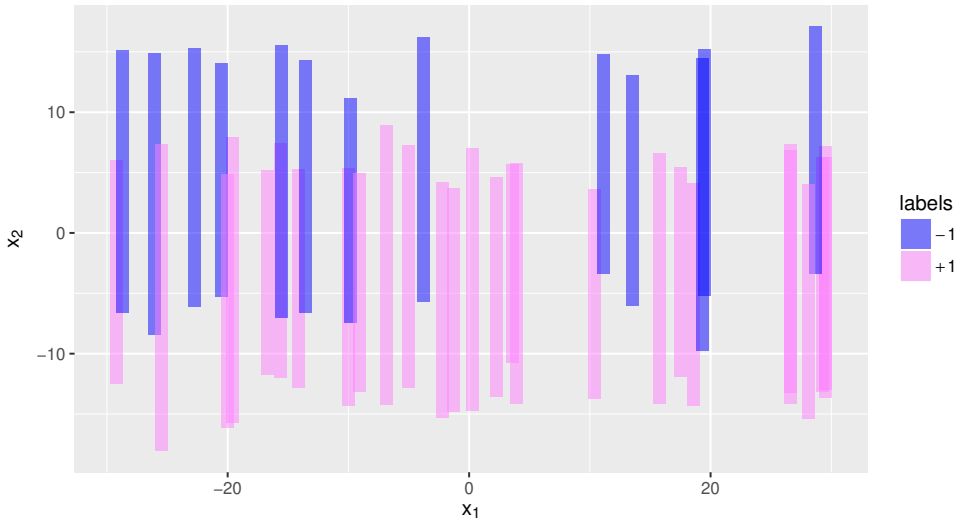


Figure 19: Example data set consisting of overlapping intervals.

Decision function obtained by the linear minimax classifier *Given this example data set the minimiser of $\mathcal{R}_{max}$ is obtained as*

$$w = \begin{pmatrix} 0.0354 \\ 0.0000 \end{pmatrix} \approx \begin{pmatrix} 0.04 \\ 0.00 \end{pmatrix},$$

using the results of Corollary 4.10. Hence the classifier can not distinguish between sets with midpoint above the x_1-axis which are more likely to have a negative label and sets with midpoint below the x_1-axis which are more likely to have a positive label. The probability of predicting a wrong label is calculated as

$$\begin{aligned} & P[M_1 > 0 \cap Y = -1] + P[M_1 \leq 0 \cap Y = 1] \\ & \overset{ind.}{=} P[M_1 > 0]P[Y = -1] + P[M_1 \leq 0]P[Y = 1] \\ & = (P[M_1 > 0] + P[M_1 \leq 0]\frac{1}{2} \\ & = \frac{1}{2}. \end{aligned}$$

Note that nearly half of the interval sets are expected to lie completely in the 'wrong' half space. Hence every sensible criterion for predictions produces the wrong label, not only the criterion used here.

SVM classifier featuring the linear kernel for convex sets *The optimiser of the regularised empirical risk for a Support Vector Machine with kernel k_c discussed in Section 3 is calculated as*

$$w = \begin{pmatrix} -0.0049 \\ 0.0049 \\ -0.1555 \\ -0.1425 \end{pmatrix}.$$

Thus $w_1, w_2 \in \mathbb{R}^2$ as defined in Lemma 3.13 can be computed as

$$w_1 = \begin{pmatrix} 1,1,0,0 \\ 0,0,1,1 \end{pmatrix} w \approx \begin{pmatrix} 0 \\ -0.30 \end{pmatrix}$$
$$w_2 = 0.5 \begin{pmatrix} -1,1,0,0 \\ 0,0,-1,1 \end{pmatrix} w \approx \begin{pmatrix} 0.00 \\ 0.01 \end{pmatrix}.$$

Hence, intervals with midpoints above the x_1-axis are expected to have negative label, which is mainly correct. More precisely, for the probability of predicting a wrong label holds approximately (since $w_2^T \approx (0,0)$)

$$\begin{aligned}
&P[(M_1, M_2)w_1 > 0 \cap Y = -1] + P[(M_1, M_2)w_1 \leq 0 \cap Y = 1] \\
&= P[-0.3M_2 > 0 \cap Y = -1] + P[-0.3M_2 \leq 0 \cap Y = 1] \\
&= P[M_2 < 0 | Y = -1]P[Y = -1] + P[M_2 \geq 0 | Y = 1]P[Y = 1] \\
&\overset{symm.}{=} \frac{1}{2} P[M_2 < 0 | Y = -1] \\
&= \frac{1}{2}\Phi(-4) \\
&< 0.0001.
\end{aligned}$$

5. Summary and Outlook

Thinking about measurements as interval or convex sets seems to be natural in many incidents. Whether one tries to take the observed variable's variation into account or assumes that the variable is in fact a convex set. Whenever its character as a set is related to some output, it is not satisfying to replace the convex input set by a real valued vector. Nevertheless, imputation methods seem to be common statistical practice, as methods to analyse interval or convex data are limited. The lack of methods with its corresponding need for simplification impedes the adequate study of such data. Further developing methods dealing with convex data one might hope to encourage scientists to collect interval or convex data without hesitation, whenever the data structure suggests it.

This work contributes to this development. We have seen that Support Vector Machines for classification can be adapted to convex input data. This was done by defining a kernel function acting on those sets. Furthermore this approach proved to be sufficiently flexible, as the Gaussian kernel based on this kernel was shown to be universal. Therefore, independently of the nature of the observed convex data, one can assume that the presented approach will successfully capture its structure. Applying the results of this work to concrete data sets to assess the capacity of the methods proposed could be the object of further study.

Depending on the properties of certain given data this kernel based approach might need further exploration. On the one hand it could be extended to regression problems by using appropriate loss functions. Despite classification, this is the second field of statistical problems where Support Vector Machines are mainly used. Their theory can be developed analogously to the theory of SVMs for classification, as described in the first section of this work. On the other hand, even when sticking to classification problems, one might need to establish better computational procedures for other convex sets than intervals. One aim would be to find effective procedures for convex polytops, at least in two dimensions. Evaluating the presented kernel for arbitrary convex sets is numerically expensive, as for every combination of input sets a multi-dimensional integral on the unit sphere has to be approximated. This is certainly not feasible for 'real life' applications. One way of dealing with this time-consuming computations might be to evaluate the kernel par-

allel, since evaluations for different sets are independent of each other. Moreover, to achieve better adaptation and generalisation for complex data structures, the Gaussian kernel could be extended by adding an additional second tuning parameter. The concrete realisation of this extension would need further examination, in particular one might be interested in whether the resulting kernel is still universal.

In addition to the kernel based approach a decision theoretical approach was discussed in the fourth section of this work. Whereas the theory for the first approach can fully be inherited from the classical results given in the first section, this second approach revealed some major difficulties. In particular uniqueness of solutions could only be obtained when the maximal risk is considered. However, even the solution to this so called minimax SVM is in general hard to obtain, as minimisation and maximisation cannot be exchanged. One way of dealing with those numerical difficulties could be to restrict oneself to certain combinations of loss and kernel functions, for which the corresponding optimisation problem simplifies. In particular, this is the case for linear minimax SVMs. Similar to the kernel based approach, the decision theoretical approach can be extended to regression problems. Their theory, especially problems where the dependent variable is interval valued but not the predictors, was covered by Wiencierz and Cattaneo [20].

Linear Support Vector machines based on a kernel for convex sets showed better performance on simulated data sets than corresponding minimax SVMs. However, the resulting decision function is more straightforward to interpret for the decision theoretical approach. Hence both approaches can be appropriate for analysing a given data set. Moreover, it needs to be explored to what extent SVMs for convex data improve the handling of interval and convex data compared to substituting these sets by precise values. This work hopes to encourage scientists to collect interval and convex data by contributing to the number of statistical methods that are tailored to generalised interval data.

List of Figures

References

[1] C.D. Aliprantis and K. Border. *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Studies in Economic Theory. Springer Berlin Heidelberg, 2013.

[2] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming - Theory and Algorithms*. John Wiley and Sons, New York, 3. edition, 2013.

[3] A. Christmann and I. Steinwart. Universal kernels on non-standard input spaces. In *Advances in neural information processing systems*, pages 406–414, 2010.

[4] T.-N. Do and F. Poulet. Kernel methods and visualization for interval data mining. In *Proceedings of the Conference on Applied Stochastic Models and Data Analysis, ASMDA*, pages 345–354, 2005.

[5] M. Gaudioso, G. Giallombardo, and G. Miglionico. An incremental method for solving convex finite min-max problems. *Mathematics of Operations Research*, 31(1):173–187, 2006.

[6] C. Gigola and S. Gomez. A regularization method for solving the finite convex min-max problem. *SIAM Journal on Numerical Analysis*, 27(6):1621–1634, 1990.

[7] J. Henrikson. Completeness and total boundedness of the hausdorff metric. *MIT Undergraduate Journal of Mathematics*, 1:69–80, 1999.

[8] J. Locke and S.H. EMMENS. *Selections from Locke's Essay on the Human Understanding; with introduction and notes by S. H. Emmens, etc.* Weale's series. Virtue Brothers and Company, 1866.

[9] R.D. Mauldin. *The Scottish Book: Mathematics from The Scottish Café, with Selected Problems from The New Scottish Book*. Springer International Publishing, 2015.

[10] C. S. Ong, X. Mary, S. Canu, and A. J. Smola. Learning with non-positive kernels. In *Proceedings of the twenty-first international conference on Machine learning*, page 81. ACM, 2004.

[11] M. Peterson. *An Introduction to Decision Theory*. Cambridge Introductions to Philosophy. Cambridge University Press, 2009.

[12] R. Schneider. *Convex Bodies: The Brunn–Minkowski Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2014.

[13] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, COLT '01/EuroCOLT '01, pages 416–426, London, UK, UK, 2001. Springer-Verlag.

[14] A.H. Siddiqi. *Applied Functional Analysis: Numerical Methods, Wavelet Methods, and Image Processing*. Chapman & Hall/CRC Pure and Applied Mathematics. CRC Press, 2003.

[15] I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, 2005.

[16] I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer New York, 2008.

[17] J. Szücs and J. Weidmann. *Linear Operators in Hilbert Spaces*. Graduate Texts in Mathematics. Springer New York, 2012.

[18] M.E. Taylor. *Measure Theory and Integration*. Graduate studies in mathematics. American Mathematical Society.

[19] L. V. Utkin, A. I. Chekh, and Y. A. Zhuk. Classification svm algorithms with interval-valued training data using triangular and epanechnikov kernels. 2015.

[20] A. Wiencierz and M. Cattaneo. On the validity of minimin and minimax methods for support vector regression with interval data. 2015.

A. Mathematical Preliminaries

This section deals with some essential mathematical background to this work. However, it should not be seen as a complete presentation, rather as a thematic classification of the underlying mathematical theory.

A.1. Topology and Integration

The theory of Hilbert spaces and multi dimensional integration theory is used throughout this work. Hence some results are presented here. Since properties of Hilbert spaces are vital for the corresponding minimiser of the regularised empirical risk (Definition 2.5), its definition and a characterisation of its topological dual is given here.

Definition A.1 (Real Hilbert space)
Let $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ be a real inner product space. Moreover, let $\mathcal{H}$ be complete with respect to the metric induced by the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Then $\mathcal{H}$ is called a real Hilbert space.

The following inequality is shown for every bilinear, positive semi-definite map. In particular it holds true for the inner product of a Hilbert space.

Lemma A.2 (Cauchy-Schwarz Inequality)
Let X be a set and $k : X \times X \to \mathbb{R}$ be bilinear and positive semi-definite. Then we have

$$k(x_1, x_2)^2 \leq k(x_1, x_1)k(x_2, x_2) \tag{20}$$

for all $x_1, x_2 \in X$. In particular we have for a Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$:

$$\langle x_1, x_2 \rangle_{\mathcal{H}}^2 \leq \|x_1\|_{\mathcal{H}}^2 \|x_2\|_{\mathcal{H}}^2$$

for all $x_1, x_2 \in \mathcal{H}$. Here $\| \cdot \|_{\mathcal{H}}$ denotes the norm induces by the inner product.

Proof. Let $\alpha_1 = k(x_2, x_2)$, $\alpha_2 = -k(x_1, x_2)$. Since k is positive semi-definite we have

$$\begin{aligned} 0 &\leq \sum_{i=1}^{2}\sum_{j=1}^{2} \alpha_i \alpha_j k(x_i, x_j) \\ &= k(x_2, x_2)^2 k(x_1, x_1) - 2k(x_2, x_2)k(x_1, x_2)^2 + k(x_1, x_2)^2 k(x_2, x_2) \\ &= k(x_2, x_2)(k(x_1, x_1)k(x_2, x_2) - k(x_1, x_2)^2). \end{aligned}$$

Assuming $k(x_2, x_2) = 0$ leads to $k(x_1, x_2) = 0$. Hence in this case the desired inequality is trivially fulfilled. One can therefore assume without loss of generality that $k(x_2, x_2) > 0$ which implies $k(x_1, x_2)^2 \leq k(x_1, x_1)k(x_2, x_2)$. □

Theorem A.3 (Riesz Representation Theorem)
Let $\mathcal{H}$ be a real Hilbert space. Then $\mathcal{H}$ is isometrically isomorphic to its dual via the embedding

$$\begin{aligned} \mathcal{H} &\to \mathcal{H}' \\ x &\mapsto \langle x, \cdot \rangle_{\mathcal{H}}. \end{aligned}$$

Hence for every continuous linear functional f on $\mathcal{H}$ exists an unique $x \in \mathcal{H}$ such that

$$f = \langle x, \cdot \rangle_{\mathcal{H}} \text{ and } \|f\|_{\mathcal{H}'} = \|x\|_{\mathcal{H}}.$$

Proof. See [14, page 104]. □

To evaluate the presented kernel at interval sets a multi-dimensional integral on the unit sphere needed to be solved. The proof of the corresponding theorem (Theorem 3.11) relied on some standard results in integration theory. One of them, relating the integral in d dimensions to those on $d-1$-dimensional spheres, is given here. The subsequent lemma derives a recursive formula for the surface area of a d dimensional unit sphere.

Theorem A.4 (Integration in spherical coordinates)
Let $f : \mathbb{R}^n \to \mathbb{R}$ be Lebesgue integrable. Then

$$\int_{\mathbb{R}^d} f(x)\ dx = \int_0^\infty \int_{S_{d-1}} f(rv) r^{d-1}\ dv\ dr,$$

where S_{d-1} denotes the sphere of d-dimensional ball with radius 1. That is

$$S_{d-1} = \{x \in \mathbb{R}^d |\ \sum_{i=1}^{d} x_i^2 = 1\}.$$

Proof. This theorem is a consequence of the so called co-area formula. A proof can be found in [18, page 89]. □

Lemma A.5
We have for all $d \geq 2$

$$|S_d| = \frac{2\pi}{d}|S_{d-2}|.$$

Proof. Using the previous theorem we have for all $d \in \mathbb{N}$

$$|B_d(R)| = \int_{\mathbb{R}^d} \mathbb{1}_{B_d(R)} dx = \int_0^R \int_{S_{d-1}} r^{d-1}\ dv\ dr = |S_{d-1}| \left[\frac{1}{d} r^d\right]_0^R = \frac{|S_{d-1}|}{d} R^d.$$

This implies $|B_d(R)| = |B_d(1)|R^d$ for all $R > 0$. Hence using polar coordinates we obtain for all $d \geq 2$

$$\begin{aligned}
|B_d(1)| &= \int_0^1 \int_0^{2\pi} \int_{B_{d-2}(\sqrt{1-r^2})} r\ d(x_3, \ldots, x_d) d\phi dr \\
&= \int_0^1 \int_0^{2\pi} |B_{d-2}(1)|(1-r^2)^{\frac{d-2}{2}} r\ d\phi dr \\
&= 2\pi |B_{d-2}(1)| \left[-\frac{1}{d}(1-r^2)^{\frac{d}{2}}\right]_0^1 = \frac{2\pi}{d}|B_{d-2}(1)|.
\end{aligned}$$

The desired result is therefore obtained via

$$|S_d| = d|B_{d+1}(1)| = 2\pi|B_{d-1}(1)| = \frac{2\pi}{d}|S_{d-1}|.$$

□

A.2. Convex Optimisation

Since the regularised empirical risk functional is a convex function, finding its minimiser is a convex optimisation problem. Problems of this type have been studied in great detail and some results used in this work are presented here. We first provide some properties of convex functions.

Definition A.6 (Convex function)
Let X be a normed vector space and $C \subseteq X$ convex. $f : C \to \mathbb{R}$ is called ***convex*** *if*

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

for all $x, y \in X$ and for all $t \in [0,1]$.

Lemma A.7
Let $S \subseteq \mathbb{R}^d$ be a nonempty convex set, and let $f : S \to R$ be convex. Then f is continuous on the interior of S.

Proof. see [2, page 100]. □

The next lemma is essential for the discussion of the maximal risk $\mathcal{R}_{max}$. It is the main reason for this functional to yield a unique minimiser, as lower-semicontinuity and convexity are requirements of Theorem A.10.

Lemma A.8
Let X be a normed space, A be some index set and $g_a : X \to \mathbb{R}$ for all $a \in A$. Let $g := \sup_{a \in A} g_a$ be the pointwise supremum. Then the following statements hold true.

1. *If g_a is lower-semicontinuous for all $a \in A$, then so is g.*
2. *If g_a is convex for all $a \in A$, then so is g.*

Proof. For a proof of the first part see [1, page 43]. To show the second part let $t \in [0,1]$ and $x_1, x_2 \in X$. Then we have

$$\begin{aligned} g(tx_1 + (1-t)x_2) &= \sup_{a \in A} g_a(tx_1 + (1-t)x_2) \\ &\leq \sup_{a \in A} [t g_a(x_1) + (1-t) g_a(x_2)] \\ &\leq t \sup_{a \in A} g_a(x_1) + (1-t) \sup_{a \in A} g_a(x_2)] \\ &= t g(x_1) + (1-t) g(x_2), \end{aligned}$$

where we obtained the first inequality via using that g_a is convex for all $a \in A$. □

To ensure the existence of a minimiser in an unbounded vector space, we require a functional to be coercive.

Definition A.9 (Coercive functional)
Let X be a normed vector space and $f : X \to \mathbb{R}$. f is called ***coercive*** *if*

$$\|x\| \to \infty \Rightarrow |f(x)| \to \infty.$$

Theorem A.10 (Mazur-Schauder)
Let $\mathcal{E}$ be a reflexive Banach space, $C \neq \emptyset$ a closed and convex subset of $\mathcal{E}$. Let ϕ be a lower-semicontinuous, convex and coercive functional on C. If ϕ is bounded from below then it has a minimal solution.

Proof. See [9, page 37]. Definitions for the terms appearing in this theorem can be found there as well. □

The Karush-Kuhn-Tucker conditions (conditions $I - III$ in the next theorem) give necessary requirements for an optimum of a constrained optimisation problem. Since minimising the regularised empirical risk with a hinge loss function can be formulated in this way, Theorem A.11 is used to derive the dual formulation in Subsection 2.4.

Theorem A.11 (Karush-Kuhn-Tucker)
Let $\emptyset \neq X \subseteq \mathbb{R}^n$ be open and let $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ for all $i = 1, \ldots, m$. Consider the optimisation problem P

$$\begin{aligned} &\textit{minimize} \quad f(x) \\ &\textit{subject to} \quad g_i(x) \leq 0 \quad \forall i = 1, \ldots, m \\ &\textit{with respect to} \quad x \in X. \end{aligned}$$

Let $x^* \in X$ be a solution of P with f, g_i being differentiable at x^* for all $i = 1, \ldots, m$. Then there exists $u \in \mathbb{R}^m$ such that

$$\begin{aligned} I: &\quad \nabla f(x^*) + \textstyle\sum_{i=1}^m u_i \nabla g_i(x^*) = 0 \\ II: &\quad u_i g_i(x) = 0 \quad \forall i = 1, \ldots, n \\ III: &\quad u_i \geq 0 \quad \forall i = 1, \ldots, n. \end{aligned}$$

Proof. See [2, page 190] □

B. R-Package: convexdatasvm

The methods derived in the third and the forth section are implemented in **R**. The corresponding functions are collected in a package called **convexdatasvm**. This package should not be seen as fully functional and ready to use implementation, though. It is rather constructed to give an idea of how SVMs using a kernel for convex sets and minimax SVMs can be implemented. Some basic features are still missing, like proper input checking and a wider choice of loss and kernel functions. Furthermore, no test are executed. Nevertheless, the functions in **convexdatasvm** have already been used to create the examples throughout this work.

B.1. User Manual

The next pages present the user manual belonging to **convexdatasvm**.

Package 'convexdatasvm'

August 3, 2017

Type Package

Title SVM for classification of convex data

Version 0.1.0

Description Classification using a Support Vektor Machine based on a kernel for convex sets. For the case of linear separation a minimax approach is implemented as well.

License CC0

Encoding UTF-8

LazyData true

Imports kernlab,
SphericalCubature,
stats,
grDevices

Depends ggplot2

Suggests testthat

RoxygenNote 6.0.1

R topics documented:

`autoplot.convex_data_svm` *Plot function for convex_data_svm classifier*

Description

Plotting method for data of class 'convex_data_svm'.

Usage

```
## S3 method for class 'convex_data_svm'
autoplot(object, existing_plot = NULL,
  direction = TRUE, colours = c(1, 2), ...)
```

Arguments

`object`	object of class 'convex_data_svm', usually a result of a call to `get_convex_data_svm`.
`existing_plot`	ggplot object to which this plot should be added. NULL for new plot.
`direction`	TRUE if the direction of larger values for the decision function shoul be added.
`colours`	vector of colours for both labels.
`...`	further graphical arguments, to be passed on to ggplot.

Value

a ggplot object

Examples

```
set.seed(21)
d <- 2
n <- 20
lower <- matrix(rnorm(d*n, sd = 2.5), d)
upper <- lower + matrix(rnorm(d*n, mean = 1, sd = 0.2), d)^2
intervals <- lapply(1:n, function(i) cbind(lower[,i], upper[,i]))
labels <- sign(rnorm(n, sapply(intervals, mean)))
interval_data <- interval_data(intervals,labels)
convex_data_svm <- get_convex_data_svm(interval_data)
autoplot(convex_data_svm)
```

`autoplot.minimax_svm` *Plot function for minimax classifier*

Description

autoplot method for data of class 'minimax_svm'.

Usage

```
## S3 method for class 'minimax_svm'
autoplot(object, existing_plot = NULL,
  critical_points = FALSE, direction = TRUE, colours = c(1, 2), ...)
```

Arguments

object	object of class 'minimax_svm', usually a result of a call to get_minimax_svm.
existing_plot	ggplot object to which this plot should be added. NULL for new plot.
critical_points	TRUE if critical points should be added.
direction	TRUE if the direction of larger values for the decision function shoul be added.
colours	vector of colours for both labels.
...	further graphical arguments, to be passed on to ggplot.

Value

a ggplot object.

See Also

autoplot.convex_data and autoplot.interval_data

Examples

```
set.seed(21)
d <- 2
n <- 20
lower <- matrix(rnorm(d*n, sd = 2.5), d)
upper <- lower + matrix(rnorm(d*n, mean = 1, sd = 0.2), d)^2
intervals <- lapply(1:n, function(i) cbind(lower[,i], upper[,i]))
labels <- sign(rnorm(n, sapply(intervals, mean)))
interval_data <- interval_data(intervals,labels)
minimax_svm <- get_minimax_svm(interval_data)
autoplot(minimax_svm)
```

---

autoplot_data *Plot functions for two dimensional interval/convex data.*

Description

autoplot method for data of class 'interval_data' and 'convex_data'.

Usage

```
## S3 method for class 'interval_data'
autoplot(object, colours = c(1, 2), ...)

## S3 method for class 'convex_data'
autoplot(object, colours = c(1, 2), ...)
```

Arguments

object	object of class interval_data or convex_data.
colours	vector of length two defining the colours of the sets differenciated by label.
...	further graphical arguments, to be passed on to ggplot.

Value

a ggplot object

Examples

```
set.seed(21)
d <- 2
n <- 20
lower <- matrix(rnorm(d*n, sd = 2.5), d)
upper <- lower + matrix(rnorm(d*n, mean = 1, sd = 0.2), d)^2
intervals <- lapply(1:n, function(i) cbind(lower[,i], upper[,i]))
labels <- sign(rnorm(n, sapply(intervals, mean)))
interval_data <- interval_data(intervals,labels)
autoplot(interval_data)

interval_points <- lapply(intervals, function(interval) {
 t(expand.grid(lapply(1:d, function(i) interval[i,])))
 })
convex_data <- convex_data(interval_points,labels)
autoplot(convex_data)
```

---

constructor_functions *Construct objects of class 'interval_data' and 'convex_data'*

Description

Creates objects of class 'interval_data' and 'convex_data'

Usage

```
interval_data(intervals, labels)

convex_data(convex_sets, labels)
```

Arguments

intervals	list of d-dimensional intervals, each interval is a dx2 matrix with lower bound in the first coordinate and upper bound in the second.
labels	numerical vector of labels -1 and +1.
convex_sets	list of matrices with d-rows, coloums indicate extreme points of the convex set.

Value

an object of class 'interval data' or 'convex_data'.

Examples

```
set.seed(21)
d <- 2
n <- 20
lower <- matrix(rnorm(d*n, sd = 2.5), d)
upper <- lower + matrix(rnorm(d*n, mean = 1, sd = 0.2), d)^2
intervals <- lapply(1:n, function(i) cbind(lower[,i], upper[,i]))
labels <- sign(rnorm(n, sapply(intervals, mean)))
interval_data <- interval_data(intervals,labels)

interval_points <- lapply(intervals, function(interval) {
 t(expand.grid(lapply(1:d, function(i) interval[i,])))
 })
convex_data <- convex_data(interval_points,labels)
```

---

convexdatasvm *SVM for classification of convex data*

Description

Classification algorithm using a Support Vektor Machine based on a kernel for convex sets. For the case of linear separation, a minimax approach is implemented as well. See `get_convex_data_svm` and `get_minimax_svm`. Moreover, plotting methods for both approaches, as well as for interval and convex data sets are implemented.

---

get_convex_data_svm *get_convex_data_svm*

Description

Finds an optimal separating function using a kernel for convex data

Usage

```
get_convex_data_svm(data, loss = "hinge", lambda = 1, kernel = "linear",
  gamma = 1, offset = FALSE)
```

Arguments

`data`	data of class 'interval_data' or 'convex_data'
`loss`	character indicating a loss function, only for the "hinge" loss implemented at the moment.
`lambda`	a positiv tuning parameter
`kernel`	character indicating a kernel, must be one of "linear", "affine_linear" or "gaussian" at the moment.
`gamma`	tuning parameter for the Gaussian kernel.
`offset`	logical, should an additional offset be used.

Value

an object of class 'convex_data_svm', that is a list containing

w_optim	The optimal parameter in case of interval data.
alpha_optim	Weights for the input vectors.
offset	optimal offset, FALSE when no offset is used.
data	Input data.
loss	The loss function used.
type	Type of input data.
kernel	The kernel used.

See Also

interval_data and convex_data for the construction of suitable data.

Examples

```
set.seed(3)
d <- 2
n <- 15
middle_points <- matrix(runif(2*n, -8, 8), 2)
labels <- sign(rnorm(n, c(1,1)%*%middle_points) + 1)
data_points <- lapply(1:n, function(i) middle_points[,i] + matrix(rnorm(16, 0, 0.5), 2))
convex_data <- convex_data(data_points, labels)
get_convex_data_svm(convex_data)
```

---

get_minimax_svm *get_minimax_svm*

Description

Finds an optimal separating function using the minimax rule.

Usage

```
get_minimax_svm(data, loss = "hinge", offset = FALSE, lambda = 1)
```

Arguments

data	data of class 'interval_data'.
loss	character indicating a loss function, only for the "hinge" loss implemented at the moment.
offset	logical, should an additional offset be used.
lambda	a positiv tuning parameter.

Value

an object of class 'minimax_svm', that is a list containing

w_optim	The optimal parameter in case of interval data.
x_optim	Critical points for minimisation.
offset	optimal offset, FALSE when no offset is used.
data	Input data.
loss	The loss function used.

See Also

interval_data for the construction of suitable data.

Examples

```
set.seed(21)
d <- 2
n <- 20
lower <- matrix(rnorm(d*n, sd = 2.5), d)
upper <- lower + matrix(rnorm(d*n, mean = 1, sd = 0.2), d)^2
intervals <- lapply(1:n, function(i) cbind(lower[,i], upper[,i]))
labels <- sign(rnorm(n, sapply(intervals, mean)))
interval_data <- interval_data(intervals,labels)
minimax_svm <- get_minimax_svm(interval_data)
```

---

predict_functions *Predictions for convex data and minimax SVMs.*

Description

predict method for objects of class 'convex_data_svm' and 'minimax_svm'.

Usage

```
## S3 method for class 'minimax_svm'
predict(object, newdata, ...)

## S3 method for class 'convex_data_svm'
predict(object, newdata, ...)
```

Arguments

object	object of class 'convex_data_svm' or 'minimax_svm'.
newdata	list of new interval data, that are a dx2 matrices with lower bounds in the first column and upper bound in the second.
...	additional arguments, not used here.

Value

vector of predicted labels.

Examples

```
set.seed(21)
d <- 2
n <- 20
lower <- matrix(rnorm(d*n, sd = 2.5), d)
upper <- lower + matrix(rnorm(d*n, mean = 1, sd = 0.2), d)^2
intervals <- lapply(1:n, function(i) cbind(lower[,i], upper[,i]))
labels <- sign(rnorm(n, sapply(intervals, mean)))
interval_data <- interval_data(intervals,labels)
minimax_svm <- get_minimax_svm(interval_data)
predict(minimax_svm, intervals)

interval_points <- lapply(intervals, function(interval) {
 t(expand.grid(lapply(1:d, function(i) interval[i,])))
 })
convex_data <- convex_data(interval_points,labels)
convex_data_svm <- get_convex_data_svm(interval_data)
predict(convex_data_svm, intervals)
```

`supplementary_functions` *Supplementary functions for objects of type 'convex_data_svm'.*

Description

Calculating the decision functions restricted to point sets for objects of type 'convex_data_svm'. `get_w_projected` calculates the separating hyperplane in case of a linear kernel, `get_decision_fkt_parameter` calculates the weights for the Gaussian kernel.

Usage

```
get_w_projected(convex_data_svm)

get_decision_fkt_parameter(convex_data_svm)
```

Arguments

`convex_data_svm` object of class 'convex_data_svm', usually a result of a call to `get_convex_data_svm`.

Value

the optimal parameter projected on point sets; weights for the Gaussian kernel.

Examples

```
set.seed(3)
d <- 2
n <- 10
middle_points <- matrix(runif(2*n, -8, 8), 2)
labels <- sign(rnorm(n, c(1,1)%*%middle_points) + 1)
data_points <- lapply(1:n, function(i) middle_points[,i] + matrix(rnorm(16, 0, 0.5), 2))
```

```
convex_data <- convex_data(data_points, labels)
convex_data_svm <- get_convex_data_svm(convex_data)
get_w_projected(convex_data_svm)
```

Index

B.2. Electronic Appendix

The CD-R enclosed contains a electronic version of this document in PDF format. Moreover, several R-scripts are included, which can be sourced to produce all figures in this work. They are named according to the corresponding section.

- `plots_theory.R` produces all plots in Section 2,
- `plots_convex_kernel_1.R`, `plots_convex_kernel_2.R` and `plots_convex_kernel_3.R` produce the plots in Section 3,
- and `plots_minimax_1.R`, `plots_minimax_2.R` and `plots_minimax_3.R` produce every plot in Section 4.

To run these scripts the package 'convexdatasvm' needs to be installed and added to the local library. This package is provided on the CD-R as well.