Approximate Methods in Geometry
Lecture Notes

Bernd Gärtner Joachim Giesen Emo Welzl

Summer term 2005

Contents

Chapter 1

Some Basic Geometry

This chapter reviews some material we will need during the course and tries to get you acquainted with some unusual phenomena of high-dimensional objects.

1.1 Euclidean Space

The d-dimensional Euclidean space $\mathbb{R}^d$ is the d-dimensional vector space over the real numbers $\mathbb{R}$, equipped with the *scalar product*

$$p \cdot q := \sum_{i=1}^{d} p_i q_i, \quad p, q \in \mathbb{R}^d.$$

Members of $\mathbb{R}^d$, for example $p = (p_1, \ldots, p_d)$ and $q = (q_1, \ldots, q_d)$ are called *points* or *vectors*, depending on how we think about them. Typically, a point denotes some absolute location in space (relative to the origin), while a vector stands for the difference $p - q$ of two points. Two special points we frequently need are $\mathbf{0} := (0, \ldots, 0)$ (the origin) and $\mathbf{1} := (1, \ldots, 1)$, with dimensions clear from the context.

The scalar product defines the *Euclidean norm* of a point (or the *length* of a vector) via

$$\|p\| := \sqrt{p \cdot p}, \quad p \in \mathbb{R}^d.$$

It holds that

$$\|tp\| = |t|\|p\|, \quad p \in \mathbb{R}^d, t \in \mathbb{R}.$$

A basic and important fact is the

Triangle Inequality. $\|q - p\| \leq \|q - r\| + \|r - p\|, \quad p, q, r \in \mathbb{R}^d.$

It says that in any triangle, any of the three sides is at most as long as the sum of the two other sides, see Figure 1.1.

A simple but very useful fact is the

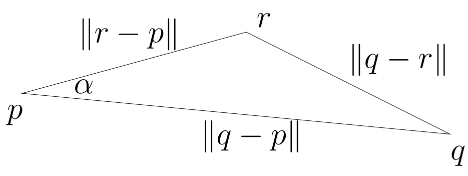

Figure 1.1: The triangle inequality and angles

Cauchy-Schwarz inequality.

$$|p \cdot q| \leq \|p\|\|q\|, \quad p, q \in \mathbb{R}^d.$$

The scalar product also defines *angles*, according to

$$\cos(\alpha) = \frac{(q-p)\cdot(r-p)}{\|q-p\|\|r-p\|},$$

see Figure 1.1. We frequently need the

Cosine Theorem. $\|q-r\|^2 = \|r-p\|^2 + \|q-p\|^2 - 2\|r-p\|\|q-p\|\cos(\alpha)$.

For $\alpha = 90^o$, this is Pythagoras's Theorem.

1.2 Hyperplanes

Any $(d+1)$-tuple $(h_1, \ldots, h_d, h_{d+1}) \in \mathbb{R}^{d+1}$ with $(h_1, \ldots, h_d) \neq \mathbf{0}$ defines a *hyperplane*

$$h = \{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} h_i x_i = h_{d+1}\}. \tag{1.1}$$

For $d = 2$, hyperplanes are *lines* (see Figure 1.2), and for $d = 3$, we get *planes*. Note that h is invariant under scaling its defining $(d+1)$-tuple by any nonzero constant.

The vector $\underline{h} = (h_1, \ldots, h_d) \in \mathbb{R}^d$ is the so-called *normal vector* of h. It is orthogonal to the hyperplane in the sense that

$$\underline{h} \cdot (p - q) = 0, \quad p, q \in h,$$

a fact that immediately follows from (1.1). It is not hard to prove (basic calculus) that the distance of h to the origin is $|h_{d+1}|/\|\underline{h}\|$, attained by the unique point $(h_{d+1}/\|\underline{h}\|^2)\underline{h}$.

Any hyperplane h comes with two *halfspaces*

$$\begin{aligned} h^+ &:= \{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} h_i x_i \geq h_{d+1}\}, \\ h^- &:= \{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} h_i x_i \leq h_{d+1}\}. \end{aligned}$$

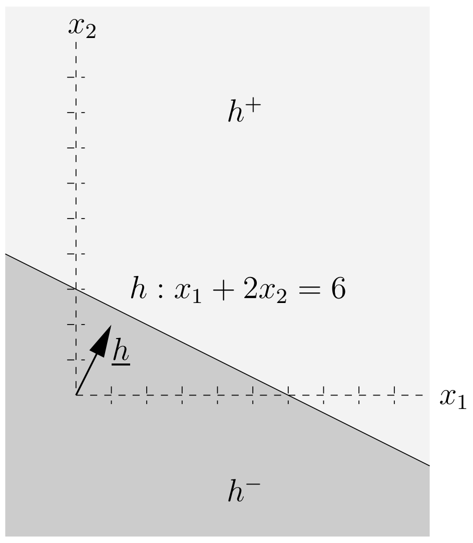

Figure 1.2: A hyperplane h in $\mathbb{R}^2$ along with its two halfspaces

In the following, we make the convention that $h_{d+1} \geq 0$ (to achieve this, we can scale by -1, if necessary). In this case, h^- contains the origin. Note that h^+ and h^- are well-defined only if $h_{d+1} \neq 0$, equivalently, if h does not contain the origin.

In a slight abuse of notation, we identify the hyperplane h with its defining $(d+1)$-tuple $(h_1, \ldots, h_{d+1})$, i.e. we write $h = (h_1, \ldots, h_{d+1})$.

Non-vertical hyperplanes. Hyperplanes h with $h_d \neq 0$ are called *non-vertical* and have an alternative definition in terms of only d parameters. Any d-tuple $(g_1, \ldots, g_d)$ defines a non-vertical hyperplane

$$g = \{x \in \mathbb{R}^d \mid x_d = \sum_{i=1}^{d-1} g_i x_i + g_d\}. \tag{1.2}$$

In this form, the line from Figure 1.2 has the equation

$$x_2 = -\frac{1}{2}x_1 + 3.$$

The non-vertical hyperplane g defines the two halfspaces

$$\begin{aligned} g^+ &:= \{x \in \mathbb{R}^d \mid x_d \geq \sum_{i=1}^{d-1} g_i x_i + g_d\}, \\ g^- &:= \{x \in \mathbb{R}^d \mid x_d \leq \sum_{i=1}^{d-1} g_i x_i + g_d\}. \end{aligned}$$

g^+ is the halfspace *above* g, while g^- is *below* g. As above, we identify the non-vertical hyperplane g with its defining d-tuple $(g_1, \ldots, g_d)$ by writing $g = (g_1, \ldots, g_d)$.

1.3 Duality

Points and hyperplanes behave in the same way. Even if it is not clear what this exactly means, the statement may appear surprising at first sight. Here are two *duality transforms* that map points to hyperplanes and vice versa, in such a way that relative positions of points w.r.t. hyperplanes are preserved.

The origin-avoiding case. For $p = (p_1, \ldots, p_d) \in \mathbb{R}^d \setminus \{\mathbf{0}\}$, the origin-avoiding hyperplane

$$p^* = (p_1, \ldots, p_d, 1) = \{x \in \mathbb{R}^d \mid \sum_{i=1}^{d} p_i x_i = 1\} \tag{1.3}$$

is called the hyperplane *dual to* p. Vive versa, for an origin-avoiding hyperplane $h = (h_1, \ldots, h_{d+1})$, the point

$$h^* = \left(\frac{h_1}{h_{d+1}}, \ldots, \frac{h_d}{h_{d+1}}\right) \in \mathbb{R}^d \setminus \{\mathbf{0}\} \tag{1.4}$$

is called the point *dual to* h. We get $(p^*)^* = p$ and $(h^*)^* = h$ (modulo scaling of coordinates by a positive multiple), so this duality transform is an *involution* (a mapping satisfying $f(f(x)) = x$ for all x).

It follows from the above facts about hyperplanes that p^* is orthogonal to p and has distance $1/\|p\|$ from the origin. Thus, points close to the origin are mapped to hyperplanes far away, and vice versa. p is actually on p^* if and only if $\|p\| = 1$, i.e. if p is on the so-called *unit sphere*, see Figure 1.3.

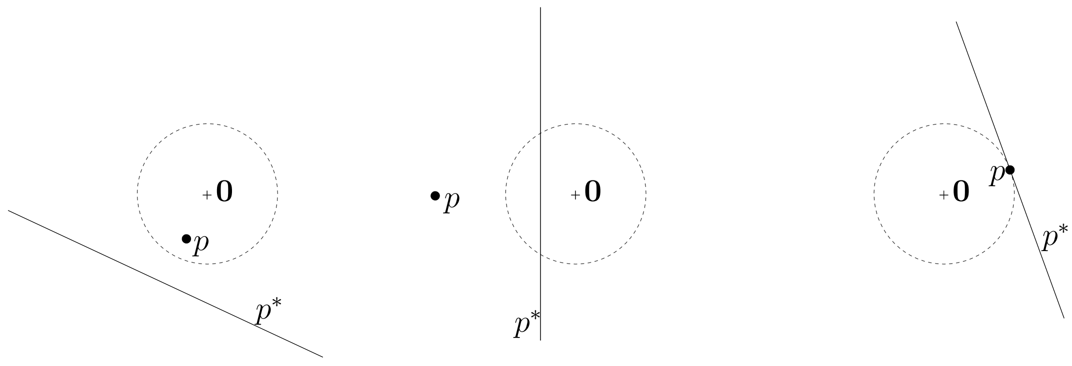

Figure 1.3: Duality in the origin-avoiding case

The important fact about the duality transform is that relative positions of points w.r.t. hyperplanes are maintained.

Lemma 1.3.1 *For all points* $p \neq \mathbf{0}$ *and all origin-avoiding hyperplanes* h*, we have*

$$p \in \left\{ \begin{array}{c} h^+ \\ h^- \\ h \end{array} \right\} \quad \Leftrightarrow \quad h^* \in \left\{ \begin{array}{c} (p^*)^+ \\ (p^*)^- \\ p^* \end{array} \right\}.$$

Proof. Really boring, but still useful in order to see what happens (or rather, that nothing happens). Let's look at h^+, the other cases are the same.

$$p \in h^+ :\Leftrightarrow \sum_{i=1}^{d} h_i p_i \geq h_{d+1} \Leftrightarrow \sum_{i=1}^{d} p_i \frac{h_i}{h_{d+1}} \geq 1 \Leftrightarrow: h^* \in (p^*)^+.$$

$\square$

The non-vertical case. The previous duality has two kinds of singularities: it does not work for the point $p = \mathbf{0}$, and it does not work for hyperplanes containing $\mathbf{0}$. The following duality has only one kind of singularity: it does not work for vertical hyperplanes, but it works for *all* points.

For $p = (p_1, \ldots, p_d) \in \mathbb{R}^d$, the non-vertical hyperplane

$$p^* = (2p_1, \ldots, 2p_{d-1}, -p_d) = \{x \in \mathbb{R}^d \mid x_d = \sum_{i=1}^{d-1} 2p_i x_i - p_d\} \tag{1.5}$$

is called the hyperplane *dual to* p.[1] Vice versa, given a non-vertical hyperplane $g = (g_1, \ldots, g_d)$, the point

$$g^* = \left(\frac{1}{2}g_1, \ldots, \frac{1}{2}g_{d-1}, -g_d\right) \tag{1.6}$$

is called the point *dual to* g.

Here is the analog of Lemma 1.3.1.

Lemma 1.3.2 *For all points p and all non-vertical hyperplanes g, we have*

$$p \in \left\{ \begin{array}{c} g^+ \\ g^- \\ g \end{array} \right\} \quad \Leftrightarrow \quad g^* \in \left\{ \begin{array}{c} (p^*)^+ \\ (p^*)^- \\ p^* \end{array} \right\}.$$

Proof. Again, this is really easy (and we only do the g^+-case).

$$\begin{aligned} p \in g^+ \quad &:\Leftrightarrow \quad p_d \geq \sum_{i=1}^{d-1} g_i p_i + g_d \\ &\Leftrightarrow \quad -g_d \geq \sum_{i=1}^{d-1} 2p_i \, \frac{1}{2} g_i - p_d \\ &\Leftrightarrow: \quad g^* \in (p^*)^+ \end{aligned}$$

$\square$

[1] We could use another symbol to distinguish this from the previous duality, but since we never mix both dualities, it will always be clear to which one we refer.

It turns out that this duality has a geometric interpretation involving the *unit paraboloid* instead of the unit sphere [2]. Which of the two is more practical depends on the application.

Duality allows us to translate statements about hyperplanes into statements about points, and vice versa. Sometimes, the statement is easier to understand after such a translation. Exercise 2 gives a nontrivial example. Here is one very easy translation in the non-vertical case. In the origin-avoiding case, the essence is the same, but the precise statement is slightly different (Exercise 3).

Observation 1.3.3 *Let p, q, r be points in $\mathbb{R}^2$. The following statements are equivalent, see Figure 1.4.*

(i) The points p, q, r are collinear *(lie on the common line ℓ).*

(ii) The lines p^*, q^*, r^* are concurrent *(go through the common point ℓ^*), or are* parallel *to each other, if ℓ is vertical).*

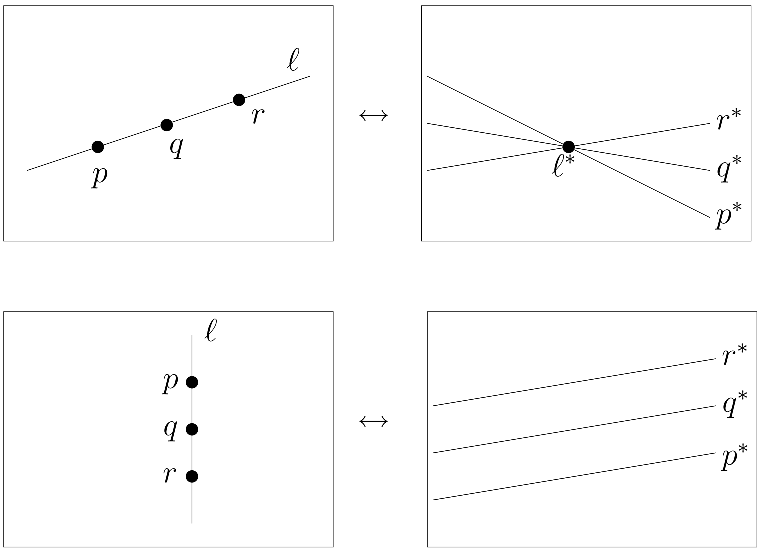

Figure 1.4: Duality: collinear points translate to concurrent lines (top) or parallel lines (bottom)

1.4 Convex Sets

A set $K \subseteq \mathbb{R}^d$ is called *convex* if for all $p, q \in K$ and for all $\lambda \in [0, 1]$, we also have

$$(1 - \lambda)p + \lambda q \in K.$$

Geometrically, this means that for any two points in K, the connecting *line segment* is completely in K, see Figure 1.5.

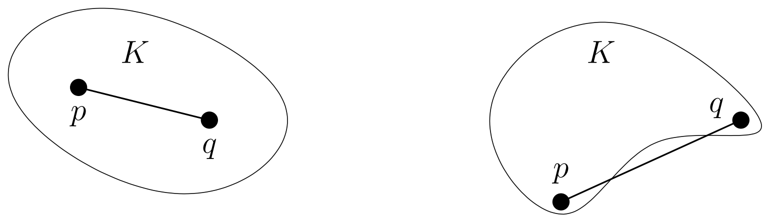

Figure 1.5: A convex set (left) and a non-convex set (right)

It immediately follows that the intersection of convex sets is convex. Convex sets are "nice" sets in many respects, and we often consider the *convex hull* of a set.

Lemma 1.4.1 *Let $X, K \subseteq \mathbb{R}^d$. The following statements are equivalent, and if K satisfies any of them, K is the* convex hull $\operatorname{conv}(X)$ *of X.*

(i) K is the intersection of all *convex sets containing X,*

$$K = \bigcap_{\substack{C \supseteq X \\ C \text{ convex}}} C.$$

(ii) K is the intersection of all halfspaces *containing X,*

$$K = \bigcap_{\substack{H \supseteq X \\ H \text{ halfspace}}} H.$$

(iii) K is the set of all convex combinations *of elements of X,*

$$K = \{\sum_{x \in S} \lambda_x x \mid S \subset X \textit{ finite}, \sum_{x \in S} \lambda_x = 1, \forall x \in S : \lambda_x \geq 0\}.$$

Of particular interest for us are convex hulls of point clouds (finite point sets), see Figure 1.6 for an illustration in $\mathbb{R}^2$.

Here is one very important statement about convex sets.

> **Helly's Theorem.** Let $C_1, \ldots, C_n$ be $n \geq d+1$ convex sets in $\mathbb{R}^d$. If every $d+1$ of the sets have a non-empty common intersection, the common intersection of all sets is nonempty.

For a proof, see Edelsbrunner [2], and for an application, see Exercise 2.

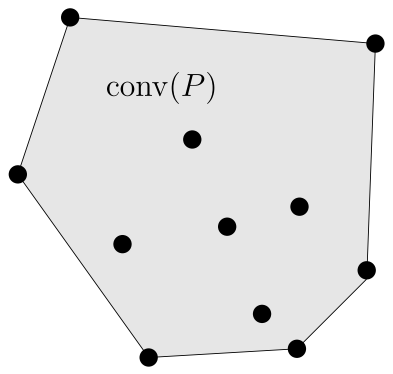

Figure 1.6: The convex hull of a point cloud $P \subseteq \mathbb{R}^2$

1.5 Balls and Boxes

Here are the most fundamental convex sets in $\mathbb{R}^d$ (see also Exercise 4).

Definition 1.5.1 *Fix $d \in \mathbb{N}, d \geq 1$.*

(i) Let $\underline{b} = (\underline{b}_1, \ldots, \underline{b}_d) \in \mathbb{R}^d$ and $\overline{b} = (\overline{b}_1, \ldots, \overline{b}_d) \in \mathbb{R}^d$ be two d-tuples such that $\underline{b}_i \leq \overline{b}_i$ for $i = 1, \ldots, d$. The box $Q_d(\underline{b}, \overline{b})$ *is the d-fold Cartesian product*

$$Q_d(\underline{b}, \overline{b}) := \prod_{i=1}^{d} [\underline{b}_i, \overline{b}_i] \subseteq \mathbb{R}^d.$$

(ii) $Q_d := Q_d(\mathbf{0}, \mathbf{1})$ is the unit box*, see Figure 1.7 (left).*

(iii) Let $c \in \mathbb{R}^d, \rho \in \mathbb{R}^+$. The ball $B_d(c, \rho)$ *is the set*

$$B_d(c, \rho) = \{x \in \mathbb{R}^d \mid \|x - c\| \leq \rho\}.$$

(iv) $B_d := B_d(\mathbf{0}, 1)$ is the unit ball*, see Figure 1.7 (right).*

While we have a good intuition concerning balls and boxes in dimensions 2 and 3, this intuition does not capture the behavior in higher dimensions. Let us discuss a few counterintuitive phenomena.

Diameter. The *diameter* of a compact[2] set $X \subseteq \mathbb{R}^d$ is defined as

$$\text{diam}(X) = \max_{x,y \in X} \|x - y\|.$$

What can we say about the diameters of balls and boxes?

[2]a set that is closed and bounded

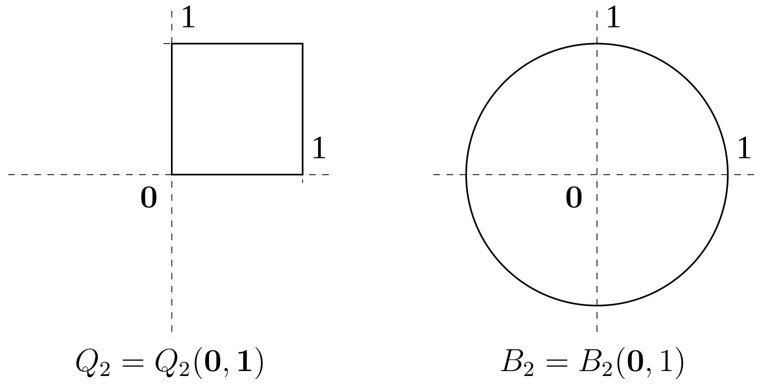
1
1
0
$Q_2 = Q_2(\mathbf{0}, \mathbf{1})$
1
1
0
$B_2 = B_2(\mathbf{0}, 1)$

Figure 1.7: The unit box (left) and the unit ball (right)

Lemma 1.5.2 *For $d \in \mathbb{N}, d \geq 1$,*

(i) $\operatorname{diam}(Q_d) = \sqrt{d}$*, and*

(ii) $\operatorname{diam}(B_d) = 2$.

Proof. This is not difficult, but it is instructive to derive it using the material we have. For $x, y \in Q_d$, we have $|x_i - y_i| \leq 1$ for $i = 1, \ldots, d$, from which

$$\|x - y\|^2 = (x - y) \cdot (x - y) = \sum_{i=1}^{d} (x_i - y_i)^2 \leq d$$

follows, with equality for $x = \mathbf{0}, y = \mathbf{1}$. This gives (i). For (ii), we consider $x, y \in B_d$ and use the triangle inequality to obtain

$$\|x - y\| \leq \|x - \mathbf{0}\| + \|\mathbf{0} - y\| = \|x\| + \|y\| \leq 2,$$

with equality for $x = (1, 0, \ldots, 0), y = (-1, 0, \ldots, 0)$. This is (ii). $\square$

The counterintuitive phenomenon is that the unit box contains points which are arbitrarily far apart, if d only gets large enough. For example, if our unit of measurement is cm (meaning that the unit box has side length $1\mathrm{cm}$), we find that $Q_{10{,}000}$ has two opposite corners which are $1\mathrm{m}$ apart; for $Q_{10^{10}}$, the diameter is already $1\mathrm{km}$.

Volume. The *volume* of a compact set $X \subseteq \mathbb{R}^d$ is defined as

$$\operatorname{vol}(X) = \int_{\mathbb{R}^d} \chi_X(x) dx,$$

where χ_X is the *characteristic function* of X,

$$\chi_X(x) = \begin{cases} 1, & \text{if } x \in X, \\ 0, & \text{otherwise.} \end{cases}$$

Lemma 1.5.3 *Let* $d \in \mathbb{N}, d \geq 1$.

(i) $\mathrm{vol}(Q_d) = 1$, *and*

(ii) $\mathrm{vol}(B_d) = \pi^{d/2}/\Gamma(\frac{d}{2}+1)$, *where* Γ *is the* Gamma *function. In particular,*

$$\Gamma\left(\frac{d}{2}+1\right) = \begin{cases} \frac{d}{2}!, & \textit{if } d \textit{ is even}, \\ \sqrt{\pi} \prod_{m=0}^{(d-1)/2} \left(m + \frac{1}{2}\right), & \textit{if } d \textit{ is odd}. \end{cases}$$

We skip the proof, because it takes us too far away from our actual topic; here is just the rough idea for (ii): *Cavalieri's principle* says that the volume of a compact set in $\mathbb{R}^d$ can be calculated by integrating over the $(d-1)$-dimensional volumes of its *slices*, obtained by cutting the set orthogonal to some fixed direction. In case of a ball, these slices are balls again, so we can use induction to reduce the problem in $\mathbb{R}^d$ to the problem in $\mathbb{R}^{d-1}$.

Let us discuss the counterintuitive implication of Lemma 1.5.3. The intuition tells us that the unit ball is larger than the unit box, and for $d = 2$, Figure 1.7 clearly confirms this. B_2 is larger than Q_2 by a factor of π (the volume of B_2). You might recall (or derive from the lemma) that

$$\mathrm{vol}(B_3) = \frac{4}{3}\pi,$$

meaning that B_3 is larger than Q_3 by a factor of more than four. Next we get

$$\mathrm{vol}(B_4) \approx 4.93, \quad \mathrm{vol}(B_5) \approx 5.26,$$

so $\mathrm{vol}(B_d)/\mathrm{vol}(Q_d)$ seems to grow with d. Calculating

$$\mathrm{vol}(B_6) \approx 5.17$$

makes us sceptical, though, and once we get to

$$\mathrm{vol}(B_{13}) \approx 0.91,$$

we have to admit that the unit ball in dimension 13 is in fact *smaller* than the unit box. From this point on, the ball volume rapidly decreases (Table 1.1), and in the limit, it even vanishes:

$$\lim_{d\to\infty} \mathrm{vol}(B_d) = 0,$$

because $\Gamma(d/2+1)$ grows faster than $\pi^{d/2}$.

d	13	14	15	16	17	$\cdots$	20
$\text{vol}(B_d)$	0.91	0.6	0.38	0.24	0.14	$\cdots$	0.026

Table 1.1: Unit ball volumes

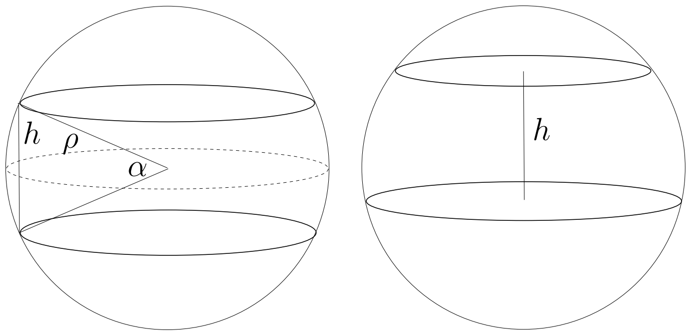

Figure 1.8: The giant lawn mower covers a stripe of width h

Area. Suppose the surface of the earth is completely covered with grass, and your task is to mow it. You have a giant lawn mower able to mow a stripe that spans a spherical angle of α, say (where α is small in order not to make your task too easy). What percentage of the grass have you mowed after you have gone around the equator once? See Figure 1.8 (left) for an illustration of the situation.

Obviously, this is a question about the surface area of (parts of) a threedimensional ball. We will not get into the business of formally defining area here; luckily, others have done the work for us, and the lawn mower question is easy to solve using a known fact. The area covered by pushing the lawn mower around the equator is

$$2\pi\rho h,$$

with ρ the radius of the ball (in our case, $\rho \approx 6{,}378\text{km}$) and h the width of the stripe. Interestingly, this area does not depend on the stripe being centered around the equator—any stripe of width h has area $2\pi\rho h$, see Figure 1.8 (right). For $h = 2\rho$, the stripe covers the whole surface, and the area is $4\pi\rho^2$, the known formula for the surface area of $B_3(\mathbf{0}, \rho)$.

Because the stripe spans spherical angle α, we get $h = 2\rho\sin(\alpha/2)$, meaning that the fraction of the earth's surface you have mowed is

$$\frac{2\pi\rho h}{4\pi\rho^2} = \frac{h}{2\rho} = \sin(\alpha/2).$$

If $\alpha = 10^o$, for example (a pretty big mower, the stripe is more than $1,000\text{km}$ wide), the fraction covered is 8.7%.

The counterintuitive phenomenon is that your task would be much simpler if the earth were of higher dimension. For sufficiently large dimension, one round with your

10^o-mower (or any α-mower, for fixed α) covers 90% (or any desired percentage) of the surface. This means, the surface area of B_d is concentrated around the equator for large d. Not only that: by symmetry of B_d, the surface area is concentrated around *any* equator. Figure 1.9 shows the (width of the) stripe around the equator that contains 90% of the area, for three values of d, see Matoušek's book [5]).

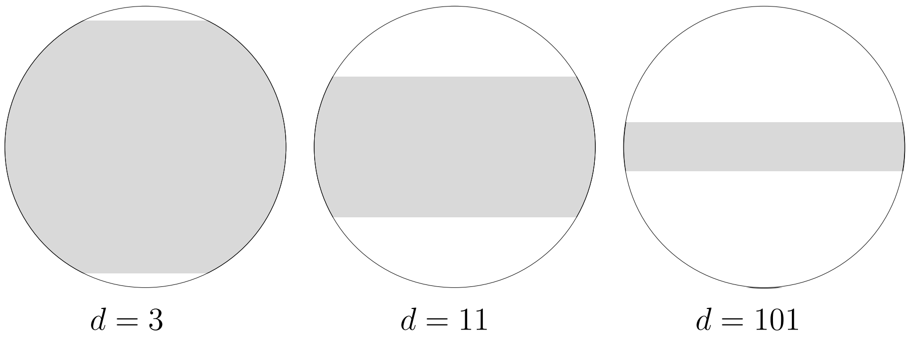

Figure 1.9: Stripe around the equator containing 90% of the area

Without seeing the connection yet, you already know a similar phenomenon involving the unit box Q_d. Let the "equator" of Q_d be the set

$$\{x \in Q_d \mid \sum_{i=1}^{d} x_i = \frac{d}{2}\},$$

see Figure 1.10 (left) for a picture in dimension 3.

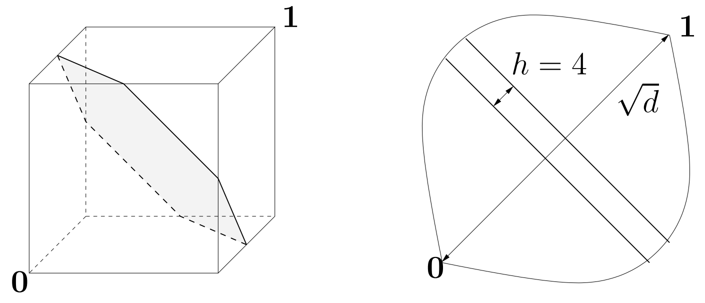

Figure 1.10: The "equator" of the unit box (left); symbolic drawing of a stripe of width h around the equator that contains 96.3% of all box corners (right)

Note that the equator is the intersection of Q_d with a hyperplane. Motivated by Figure 1.9, we plan to prove now that the stripe around the equator containing 90% of the 2^d box corners becomes thinner and thinner (compared to the diameter of the box), as d grows. It turns out that this is nothing else than the well-known

Chernoff Bound. Let $X_1, \ldots, X_d$ be independent random variables with $\mathrm{prob}(X_i = 0) = \mathrm{prob}(X_i = 1) = 1/2$ for all i, and let $X = X_1 + \cdots + X_d$. For all $\delta > 0$, we have

$$\text{prob}\left(\left|X - \frac{d}{2}\right| > \delta\frac{d}{2}\right) < 2\exp\left(-\delta^2\frac{d}{4}\right).$$

This follows from the *lower tail bound* for independent Poisson trials (see for example [6, Theorem 4.2]), together with the fact that upper and lower tails have the same distribution in our symmetric case.

Setting $\delta = 4/\sqrt{d}$, for example, yields

$$\text{prob}\left(\left|X - \frac{d}{2}\right| > 2\sqrt{d}\right) < 2\exp(-4) \approx 0.037.$$

Because X is the sum of coordinates of a randomly chosen unit box corner, it follows that a fraction of no more than 3.7% of all box corners is outside the stripe

$$\{x \in \mathbb{R}^d \mid \frac{d}{2} - 2\sqrt{d} \leq \sum_{i=1}^{d} x_i \leq \frac{d}{2} + 2\sqrt{d}\}$$

around the equator. It follows from our earlier material on hyperplanes (calculation of distance to the origin) that the width of this stripe is

$$\frac{d/2 + 2\sqrt{d}}{\sqrt{d}} - \frac{d/2 - 2\sqrt{d}}{\sqrt{d}} = 4,$$

a constant! As d gets larger, the stripe therefore becomes thinner and thinner compared to the stripe

$$\{x \in \mathbb{R}^d \mid 0 \leq \sum_{i=1}^{d} x_i \leq d\}$$

of width $\sqrt{d}$ containing the whole unit box, see Figure 1.10 (right) for a symbolic picture.

1.6 Exercises

Exercise 1 *A finite point set $P \subseteq \mathbb{R}^d$ is called* affinely independent *if the two equations*

$$\sum_{p\in P} \lambda_p p = \mathbf{0}, \quad \sum_{p\in P} \lambda_p = 0$$

imply $\lambda_p = 0$ for all $p \in P$. Prove that if P is an affinely independent point set with $|P| = d$, then there exists a unique hyperplane containing all points in P. (This generalizes the statement that there is a unique line through any two distinct points.)

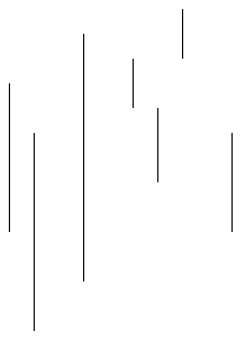

Figure 1.11: A set of vertical line segments in $\mathbb{R}^2$

Exercise 2 *Let S be a set of vertical* line segments[3] *in $\mathbb{R}^2$, see Figure 1.11. Prove the following statement: if for every three of the line segments, there is a line that intersects all three segments, then there is a line that intersects* all *segments.*

Hint. Use the duality transform (non-vertical case) and Helly's Theorem. For this, you need to understand the following: (i) what is the set of lines dual to the set of points on a (vertical) segment? (ii) if a line intersects the segment, what can we say about the point dual to this line?

Exercise 3 *State and prove the analog to Observation 1.3.3 for the origin-avoiding case.*

Exercise 4 *Prove that all boxes $Q_d(\underline{b}, \overline{b})$ and all balls $B(c, \rho)$ are convex sets.*

Exercise 5 *In order to generate a random point p in B_d, we could proceed as follows: first generate a random point p in $Q_d(-\mathbf{1}, \mathbf{1})$ (this is easy, because it can be done coordinatewise); if $p \in B_d$, we are done, and if not, we repeat the choice of p until $p \in B_d$ holds. Explain why this is not necessarily a good idea. For $d = 20$, what is the expected number of trials necessary before the event '$p \in B_d$' happens?*

[3]a line segment is the convex hull of a set of two points

Chapter 2

Approximate Smallest Enclosing Balls

2.1 Bounding Volumes

A *bounding volume* for a set $S \subseteq \mathbb{R}^d$ is a superset of S with a simple shape, for example a box, a ball, or an ellipsoid.

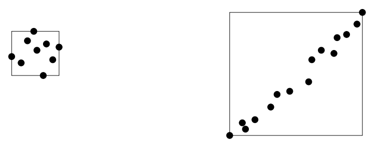

Figure 2.1: Bounding boxes $Q(P)$ of point sets $P \subseteq \mathbb{R}^2$

Bounding volumes which are smallest among the ones of a given shape (with respect to volume, diameter, or some other criteria) are very useful, because they 'approximate' the possibly complicated set S by a simple superset. Then, whenever you want to know something about S, you first try to use the bounding volume to answer the question (this is usually cheap), and only if this fails, you analyze S in more detail. Here, we focus on the situation in which S is a *point cloud* (finite point set) which we usually denote by P.

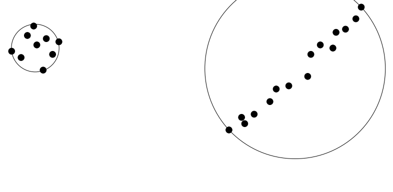

Figure 2.2: Smallest enclosing balls $B(P)$ of point sets $P \subseteq \mathbb{R}^2$

What shapes to use depends on the questions you want to ask, of course. If you want to know, for example, how the set $P \subseteq \mathbb{R}^2$ must be translated and scaled such that a printout of it fills a sheet of paper, you want to use *axes-parallel bounding boxes*. Finding the smallest axes-parallel bounding box (or simply the bounding box) is easy. For every $i \in \{1, \ldots, d\}$, iterate over the i-th coordinate of all points to find the smallest and largest one. This results in an algorithm of total runtime $O(dn)$ for $|P| = n$. See Figure 2.1 for two examples. The resulting box $Q(P)$ is smallest with respect to volume and diameter.

Alternatively, you may consider other bounding volumes. A popular one is the *smallest enclosing ball*, see Figure 2.2.

Here, *smallest* refers to the radius (or equivalently, the volume) of the ball. Exercise 7 asks you to prove that the diameter of the optimal ball $B(P)$ is never larger than the diameter of the bounding box $Q(P)$. This means, if you consider the diameter as a measure of how well the bounding volume approximates the point set, balls are better than boxes. On the other hand, if you consider the *volume* as the measure of quality, there is no clear winner. Superimposing Figures 2.1 and 2.2, you see that the ball has smaller volume than the box in the left situation, but larger volume in the right situation.

There is another aspect where the ball is the winner: rotating P changes the shape and diameter of the bounding box, but the smallest enclosing ball stays the same (up to translation). This is a desirable property in many geometric applications, because it means that the smallest enclosing ball does not have to be recomputed when we apply an *isometry* (any affine transformation[1] that preserves lengths of difference vectors) to P.

Other popular bounding volumes are boxes of arbitrary orientation (images of some $Q_d(\underline{b}, \overline{b})$ under an isometry) and ellipsoids, mostly because they approximate the volume of $\text{conv}(P)$ well. They are more difficult to compute than (axes-paralell) boxes and balls, though.

2.2 Finding an almost optimal ball

For $P \subseteq \mathbb{R}^d, |P| = n > 0$, we let $B(P)$ denote the ball of smallest radius that contains P. It is well-known that $B(P)$ exists and is unique [10], but how do we (efficiently) compute it? Recall that the bounding box was easy to find in time $O(dn)$. Can we also compute $B(P)$ in time $O(dn)$? There is no rigorous argument that this is not possible, but there are good reasons to believe that one cannot do it. At least, all known algorithms for computing $B(P)$ are much slower—so slow in fact that they will not be able to solve the problem for $n = d = 1,000$, say.

The goal of this chapter is to prove that time $O(dn)$ suffices to find a ball containing P whose radius is only 1% larger than the radius of $B(P)$. If this is not good enough, the same algorithm can produce a ball whose radius is only 0.1% larger, at the extent of running (essentially) ten times as long. In fact, any desired percentage can be

[1]a mapping $x \to Ax + b$, with $A \in \mathbb{R}^{d \times d}$ a matrix and $b \in \mathbb{R}^d$ a translation vector

achieved in time $O(dn)$, but with the constant behind the big O depending on the percentage. The algorithm (actually, a combination of two algorithms) is due to Bădoiu and Clarkson [1].

2.2.1 Basics

Here is what we mean by (the center of) an almost optimal ball, with respect to a given constant $\varepsilon \geq 0$.

Definition 2.2.1 *Let R_P be the radius of $B(P)$, $\varepsilon \geq 0$. A point $c \in \mathbb{R}^d$ is called $(1+\varepsilon)$-approximation of $B(P)$, if*

$$\max_{q \in P} \|q - c\| \leq (1+\varepsilon) R_P.$$

Note that the center c_P of $B(P)$ is a 1-approximation of $B(P)$. We will need the following statement about c_P. The origin of it is unknown to me, and often it is conceived as a more or less obvious fact. Bădoiu and Clarkson remark that the statement is proved in a paper by Goel et al. [4] which is not true. A simple proof is due to Seidel [9, 3].

Lemma 2.2.2 *For any $c \in \mathbb{R}^d$, there exists a point $p \in P$ such that*

(i) $\|p - c_P\| = R_P$*, and*

(ii) $\|p - c\|^2 \geq \|c - c_P\|^2 + \|p - c_P\|^2$,

see Figure 2.3 (left).

If $c \neq c_P$ and $|P| > 1$ (which is the case we are interested in), this lemma says that we can find a point on the boundary of $B(P)$ such that the nonzero vectors $p - c_P$ and $c - c_P$ span an obtuse angle (an angle of at least 90^o). This interpretation of the inequality in Lemma 2.2.2 (ii) is a consequence of cosine theorem.

In more geometric terms, and with Exercise 8, the inequality can also be interpreted as follows: if $c \neq c_P$, the unique hyperplane

$$h = \{x \in \mathbb{R}^d \mid (x - c_P) \cdot (c - c_P) = 0\}$$

through c_P with normal vector $c - c_P$ must have a boundary point of $B(P)$ in the halfspace

$$h^c := \{x \in \mathbb{R}^d \mid (x - c_P) \cdot (c - c_P) \leq 0\}$$

not containing c, see Figure 2.3 (left). The right part of the figure shows on an intuitive level why this condition is necessary: if no point p satisfies the criteria of the Lemma, we can shrink $B(P)$ and get a still smaller ball containing P, which is a contradiction to $B(P)$ being smallest.

Using Lemma 2.2.2, we can easily prove that a $(1+\varepsilon)$-approximation cannot be too far away from c_P, a fact we need later (Exercise 10).

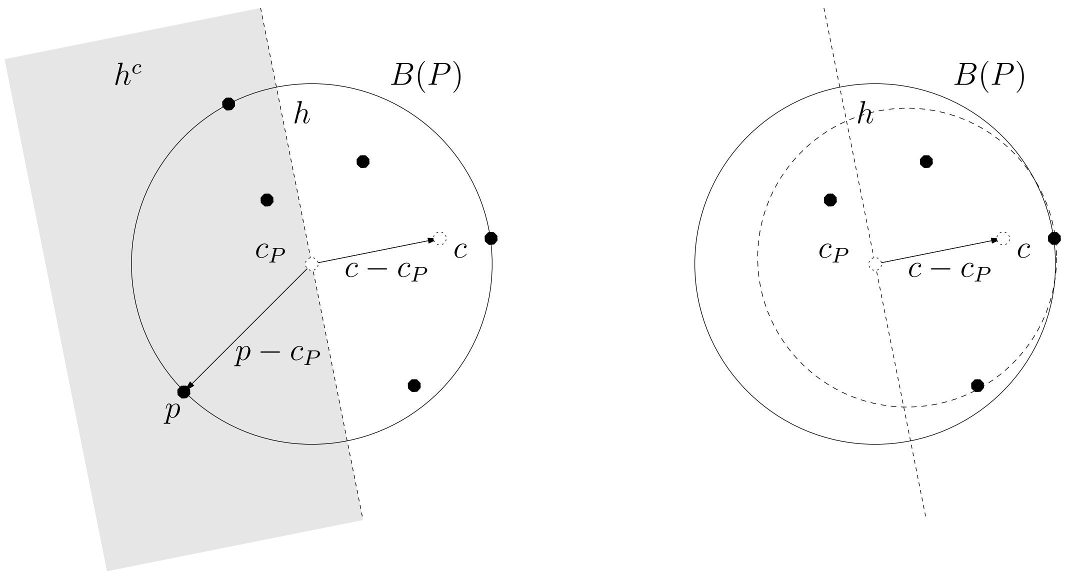

Figure 2.3: Illustration of Lemma 2.2.2

Lemma 2.2.3 *Let $c \in \mathbb{R}^d$ be a $(1+\varepsilon)$-approximation of $B(P)$. Then*

$$\|c - c_P\| \leq \sqrt{2\varepsilon + \varepsilon^2} R_p.$$

Proof. Choose p according to Lemma 2.2.2. Because p is on the boundary of $B(P)$ and c is a $(1+\varepsilon)$-approximation, we get

$$\|c - c_P\|^2 \leq \|p - c\|^2 - \|p - c_P\|^2 \leq (1+\varepsilon)^2 R_P^2 - R_P^2 = (2\varepsilon + \varepsilon^2) R_P^2.$$

□

2.2.2 Algorithm 1

Here is our first algorithm for approximating $B(P)$. For given P and $\varepsilon > 0$, this algorithm computes a sequence of centers $c_i, i = 1, \ldots, \lceil 1/\varepsilon^2 \rceil$, with the property that the last one is a $(1+\varepsilon)$-approximation of c_P.

```
Miniball_Approx1($P, \varepsilon$):
    (* computes $(1+\varepsilon)$-approximation of $B(P)$ *)
    choose $p \in P$ arbitrarily and set $c_1 := p$
    FOR $i = 2$ TO $\lceil 1/\varepsilon^2 \rceil$ DO
        choose $q \in P$ such that $\|q - c_{i-1}\|$ is maximum
        $c_i := c_{i-1} + \frac{1}{i}(q - c_{i-1})$
    END
    RETURN $c_{\lceil 1/\varepsilon^2 \rceil}$
```

Theorem 2.2.4 *For $i = 1, \ldots, \lceil 1/\varepsilon^2 \rceil$,*

$$\|c_i - c_P\| \le \frac{R_P}{\sqrt{i}}.$$

This invariant immediately implies the approximation factor of the algorithm: using the triangle inequality, we get that for all $s \in P$,

$$\|s - c_i\| \le \|s - c_P\| + \|c_i - c_P\| \le R_P + \frac{R_P}{\sqrt{i}} = \left(1 + \frac{1}{\sqrt{i}}\right) R_P. \tag{2.1}$$

It follows that c_i is a $(1 + 1/\sqrt{i})$-approximation; setting $i = \lceil 1/\varepsilon^2 \rceil$ gives the desired $(1 + \varepsilon)$-approximation.

Proof. If $|P| = 1$, we have $c_i = c_P$ in any iteration and the result follows. Otherwise, we proceed by induction. The statement holds for $i = 1$ because of $c_1 \in P$. Now assume that $i > 1$ and that we have already verified the theorem for $i - 1$. Let us apply Lemma 2.2.2 with $c = c_{i-1}$, providing us with a point p. Let q be the point chosen in iteration i.

Claim.

$$\|q - c_{i-1}\|^2 \ge \|c_{i-1} - c_P\|^2 + \|q - c_P\|^2. \tag{2.2}$$

To see this, we use the way q was chosen, together with p's properties (i) and (ii) from Lemma 2.2.2 to conclude that

$$\|q - c_{i-1}\|^2 \ge \|p - c_{i-1}\|^2 \overset{(ii)}{\ge} \|c_{i-1} - c_P\|^2 + \|p - c_P\|^2 \overset{(i)}{\ge} \|c_{i-1} - c_P\|^2 + \|q - c_P\|^2.$$

Note that for $c_{i-1} \ne c_P$, equation (2.2) is equivalent to $q \in h^{c_{i-1}}$, see Figure 2.4 and the discussion after Lemma 2.2.2.

The claimed bound for $\|c_i - c_P\|$ is obvious if $c_i = c_P$, so we will assume that $c_i \ne c_P$. Moreover, $|P| > 1$ implies $c_i \ne c_{i-1}$ (why?), and $i > 1$ implies $q \ne c_i$, see the situation in Figure 2.4.

We now restrict attention to the triangle spanned by c_P, c_{i-1} and q (this triangle also contains c_i); assume $c_P - c_i$ and $c_{i-1} - c_i$ span some angle β, so that $c_P - c_i$ and $q - c_i$ span angle $180^o - \beta$.[2] Let us introduce the following shortcuts.

$$\begin{aligned}
x &:= \|c_i - c_P\|, \\
a &:= \|c_{i-1} - c_P\|, \\
b &:= \|q - c_P\|, \\
\kappa &:= \|c_{i-1} - c_i\|, \\
\mu &:= \|q - c_i\|, \\
\lambda &:= \|q - c_{i-1}\| = \kappa + \mu.
\end{aligned}$$

[2]Because all involved vectors are nonzero, these angles are well-defined, even if the triangle spanned by c_P, c_{i-1} and q is flat.

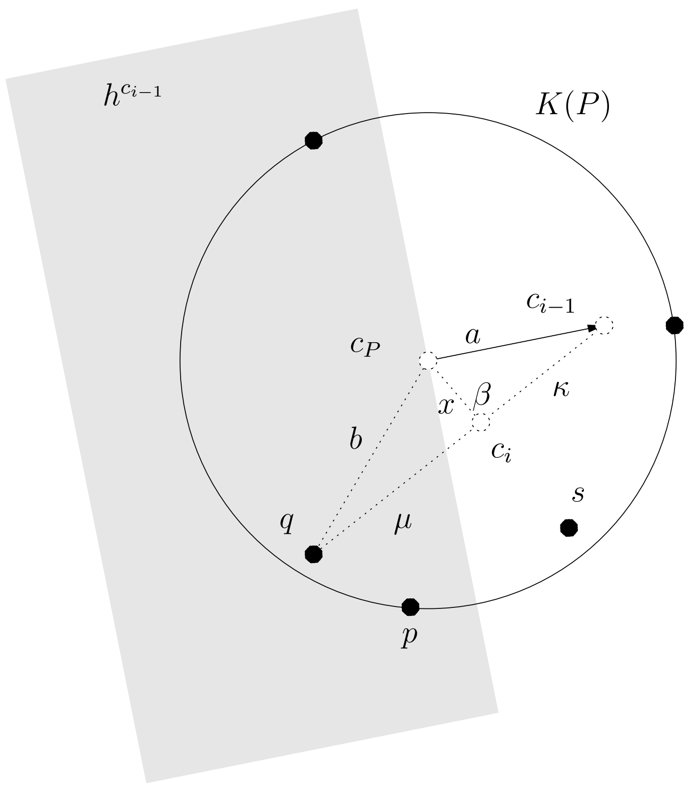

Figure 2.4: Proof of Theorem 2.2.4

Recalling that $\cos(180^o - \beta) = -\cos(\beta)$, the cosine theorem (applied to the two subtriangles involving $c_i - c_P$) yields

$$x^2 = a^2 - \kappa^2 + 2x\kappa\cos\beta, \tag{2.3}$$
$$x^2 = b^2 - \mu^2 - 2x\mu\cos\beta. \tag{2.4}$$

By definition of c_i, we have

$$\kappa = \frac{1}{i}\lambda, \quad \mu = \frac{i-1}{i}\lambda.$$

Therefore, (2.3) and (2.4) can be rewritten as follows.

$$x^2 = a^2 - \left(\frac{1}{i}\right)^2\lambda^2 + 2x\frac{1}{i}\lambda\cos\beta, \tag{2.5}$$
$$x^2 = b^2 - \left(\frac{i-1}{i}\right)^2\lambda^2 - 2x\frac{i-1}{i}\lambda\cos\beta. \tag{2.6}$$

Multiplying (2.5) with $i-1$ and adding up the two equations removes the contribution of β and x on the right hand side, and we get

$$ix^2 = (i-1)a^2 + b^2 - \frac{i-1}{i}\lambda^2. \tag{2.7}$$

Inequality (2.2) exactly says that $\lambda^2 \geq a^2 + b^2$, so that (2.7) further yields

$$\begin{aligned} x^2 &\leq \frac{1}{i}\left((i-1)a^2 + b^2 - \frac{i-1}{i}(a^2+b^2)\right) \\ &= \left(\frac{i-1}{i}\right)^2 a^2 + \left(\frac{1}{i}\right)^2 b^2. \end{aligned}$$

Inductively, we know that $a^2 = \|c_{i-1} - c_P\|^2 \leq R_P^2/(i-1)$, and because $b^2 = \|q - c_P\|^2 \leq R_P^2$, the desired inequality

$$x^2 \leq \frac{R_P^2}{i}$$

follows. $\Box$

For $|P| = n$, the runtime of Algorithm `Miniball_Approx1` is $O(dn)$ for one iteration (we need to find the largest among n scalar products), meaning that the total runtime is

$$O\left(\frac{dn}{\varepsilon^2}\right).$$

2.2.3 Algorithm 2

The denominator of ε^2 in the runtime of Algorithm `Miniball_Approx1` is not very satisfactory. Even a moderate error bound of $\varepsilon = 0.01$ (1%) leads to $10,000$ iterations in the algorithm; that way, we cannot compete with the bounding box.

Here is a second algorithm that achieves a denominator of ε. The prize to pay is that this algorithm requires as a black box the computation of the *exact* smallest enclosing ball of a small point set (by Exercise 10, a call to this black box can be replaced with a call to `Miniball_Approx1`, though).

```
Miniball_Approx2($P, \varepsilon$):
  (* computes $(1+\varepsilon)$-approximation of $B(P)$ *)
  choose $p \in P$ arbitrarily and set $c_1 := p, S_1 := \{p\}, j := 0, \Delta := \infty$
  FOR $i = 2$ TO $\lceil 2/\varepsilon \rceil$ DO
      choose $q \in P$ such that $\|q - c_{i-1}\|$ is maximum
      IF $\|q - c_{i-1}\| < \Delta$ THEN
          $j := i - 1$
          $\Delta := \|q - c_{i-1}\|$
      END
      $S_i := S_{i-1} \cup \{q\}$
      compute the smallest enclosing ball $B(S_i) = B_d(c_i, R_i)$
  END
  IF $\max_{s \in P} \|s - c_{\lceil 2/\varepsilon \rceil}\| < \Delta$ THEN
      $j = \lceil 2/\varepsilon \rceil$
  END
  RETURN $c_j$
```

In the analysis of the algorithm, we work with the following symbols.

$$\begin{aligned}
R &:= \text{radius } R_P \text{ of } B(P),\\
\bar{R} &:= (1+\varepsilon)R,\\
c_i &:= \text{center of } B(S_i),\\
R_i &:= \text{radius of } B(S_i) \text{ (note that } R_1 = 0),\\
\lambda_i &:= R_i/\bar{R},\\
k_i &:= \|c_i - c_{i-1}\|.
\end{aligned}$$

Observe that for all i,

$$\lambda_i \leq R/\bar{R} = 1/(1+\varepsilon), \tag{2.8}$$

because $S_i \subseteq P$, so $B(S_i)$ cannot have larger radius than $B(P)$. Let us assume that no c_i is a $(1+\varepsilon)$-approximation of $B(P)$. We will show that under this assumption, $\lambda_{\lceil 2/\varepsilon \rceil + 1}$ exceeds the bound in (2.8), a contradiction.[3]

To analyze the development of λ_i in iteration $i \geq 2$, we first apply Lemma 2.2.2 to the set S_{i-1} and the point $c = c_i$ to deduce the existence of $p \in S_{i-1}$ such that

$$\|p - c_i\|^2 \geq \|c_i - c_{i-1}\|^2 + \|p - c_{i-1}\|^2 = k_i^2 + R_{i-1}^2,$$

meaning that

$$\|p - c_i\| \geq \sqrt{\lambda_{i-1}^2 \bar{R}^2 + k_i^2}. \tag{2.9}$$

Furthermore, for the point q chosen in iteration i, the triangle inequality yields $\|q - c_{i-1}\| \leq \|q - c_i\| + \|c_i - c_{i-1}\|$, implying

$$\|q - c_i\| \geq \|q - c_{i-1}\| - \|c_i - c_{i-1}\| = \|q - c_{i-1}\| - k_i > \bar{R} - k_i. \tag{2.10}$$

In the last inequality (and only here), we use the assumption that c_{i-1} is not a $(1+\varepsilon)$-approximation, meaning that

$$\|q - c_{i-1}\| = \max_{s \in P} \|s - c_{i-1}\| > (1+\varepsilon)R = \bar{R}.$$

We are now approaching a recursive lower bound for λ_i. Because both p and q are in S_i, we get—using (2.9) and (2.10)—that

$$\lambda_i \bar{R} = R_i \geq \max\left(\|p - c_i\|, \|q - c_i\|\right) \geq \max\left(\sqrt{\lambda_{i-1}^2 \bar{R}^2 + k_i^2}, \bar{R} - k_i\right). \tag{2.11}$$

The first term of the maximum increases with k_i, while the second term decreases. It follows that the maximum is minimized when both terms are equal, so when

$$k_i = \frac{1 - \lambda_{i-1}^2}{2} \bar{R}$$

[3]you may object that $\lambda_{\lceil 2/\varepsilon \rceil + 1}$ does not appear in the algorithm, but for this argument, we simply consider a hypothetical last iteration with $i = \lceil 2/\varepsilon \rceil + 1$.

and consequently

$$\sqrt{\lambda_{i-1}^2 \bar{R}^2 + k_i^2} = \bar{R} - k_i = \frac{1+\lambda_{i-1}^2}{2}\bar{R}.$$

Substituting this into (2.11) yields

$$\lambda_i \geq \frac{1+\lambda_{i-1}^2}{2}, \quad i \geq 2, \tag{2.12}$$

with $\lambda_1 = R_1/\bar{R} = 0$. Equation (2.12) is equivalently written as

$$1 - \lambda_i \leq \frac{1-\lambda_{i-1}^2}{2}, \quad i \geq 2,$$

which in turn implies

$$\frac{1}{1-\lambda_i} \geq \frac{2}{(1-\lambda_{i-1})(1+\lambda_{i-1})} = \frac{1}{1-\lambda_{i-1}} + \frac{1}{1+\lambda_{i-1}} > \frac{1}{1-\lambda_{i-1}} + \frac{1}{2}, \quad i \geq 2,$$

because $\lambda_{i-1} < 1$. By expanding this, we get

$$\frac{1}{1-\lambda_i} > \frac{i-1}{2} + \frac{1}{1-\lambda_1} = 1 + \frac{i-1}{2}, \quad i \geq 1.$$

In other words,

$$\lambda_i > 1 - \frac{1}{1+(i-1)/2}, \quad i \geq 1.$$

For $i = \lceil 2/\varepsilon \rceil + 1$, this gives

$$\lambda_i > 1 - \frac{1}{1+1/\varepsilon} = \frac{1}{1+\varepsilon},$$

a contradiction to (2.8). Therefore, our initial assumption was wrong, and there must be some $c_{i_0}, i_0 \in \{1, \ldots, \lceil 2/\varepsilon \rceil\}$, which yields a $(1+\varepsilon)$-approximation. By the choice of j, the point c_j returned by the algorithm satisfies

$$\max_{s \in P} \|s - c_j\| \leq \max_{s \in P} \|s - c_{i_0}\| \leq (1+\varepsilon)R,$$

so c_j is itself a $(1+\varepsilon)$-approximation.

The runtime of the algorithm is bounded by

$$O\left(\frac{dn}{\varepsilon} + \frac{1}{\varepsilon} f_d\left(\lceil 2/\varepsilon \rceil - 1\right)\right),$$

with $f_d(n)$ the time necessary to compute the exact smallest enclosing ball of a set of n points in $\mathbb{R}^d$. The known bounds for $f_d(n)$ are exponential in d, but with Exercise 10, we can replace this by a bound which is polynomial in d and $1/\varepsilon$.

2.2.4 Core sets

Our analysis of Algorithm `Miniball_Approx2` has a very interesting consequence which we want to state explicitly (choose $S = S_j$ to get it).

Corollary 2.2.5 *For any finite point set $P \subseteq \mathbb{R}^d$ and $\varepsilon > 0$, there exists a subset $S \subseteq P$, $|S| \le \lceil 2/\varepsilon \rceil$, such that the center of $B(S)$ is a $(1+\varepsilon)$-approximation of $B(P)$.*

Such a set is called a *core set*, and we will encounter core sets for other bounding volumes later in the course.

Recall from Exercise 9 that a subset with the *same* smallest enclosing ball as P may require up to $d+1$ points (and this is tight). If you are willing to accept a small error of $\varepsilon = 0.01$, say, you can find a subset S of *constant* size 200 with 'the same' smallest enclosing ball as P, in *any* dimension. This is quite remarkable, and also exceptional. While other bounding volumes for P do have core sets whose sizes do not depend on $n = |P|$, there is usually an (exponential) dependence on d, on top of the dependence on ε.

2.3 Exercises

Exercise 6 *Let $P \subseteq \mathbb{R}^d, |P| = d+1$ be an affinely independent point set (see Exercise 1). Prove that there exists a unique ball $B_d(c,\rho)$ with all points of P on its boundary, meaning that*

$$\|p - c\| = \rho, \quad p \in P.$$

Exercise 7 *Prove that for any finite point set P, the diameter of the smallest enclosing ball $B(P)$ is at most the diameter of the smallest enclosing axes-parallel box $Q(P)$.*

Exercise 8 *Prove that the inequality in Lemma 2.2.2 (ii) is equivalent to the inequality*

$$(p - c_P) \cdot (c - c_P) \le 0.$$

Prove that

$$h = \{x \in \mathbb{R}^d \mid (x - c_P) \cdot (c - c_P) = 0\}$$

is a hyperplane containing c_P whose normal vector is a multiple of $c - c_P$.

Exercise 9 *Let $P \subseteq \mathbb{R}^d, |P| = n$. Prove the following statements about smallest enclosing balls and boxes.*

(i) There exists $T \subseteq P, |T| \le 2d$ such that

$$Q(P) = Q(T).$$

(ii) There exists $T \subseteq P, |T| \le d+1$ such that

$$B(P) = B(T).$$

Hint: For (ii), use Helly's Theorem, with balls of radius $R_P - \varepsilon$ centered at the points in P.

Exercise 10 *Consider the following variant of* `Miniball_Approx2` *that uses the previous algorithm* `Miniball_Approx1` *as a subroutine.*

```
Miniball_Approx2($P, \varepsilon', \delta$):
    (* computes $(1+\varepsilon')(1+\sqrt{2\delta+\delta^2})$-approximation of $B(P)$ *)
    choose $p \in P$ arbitrarily and set $c'_1 := p, S_1 := \{p\}, j := 0, \Delta := \infty$
    FOR $i = 2$ TO $\lceil 2/\varepsilon' \rceil$ DO
        choose $q \in P$ such that $\|q - c'_{i-1}\|$ is maximum
        IF $\|q - c'_{i-1}\| < \Delta$ THEN
            $j := i - 1$
            $\Delta := \|q - c'_{i-1}\|$
        END
        $S_i := S_{i-1} \cup \{q\}$
        $c'_i :=$ Miniball_Approx1($S_i, \delta$)
    END
    IF $\max_{s \in P} \|s - c'_{\lceil 2/\varepsilon' \rceil}\| < \Delta$ THEN
        $j = \lceil 2/\varepsilon' \rceil$
    END
    RETURN $c'_j$
```

Prove that the algorithm achieves the indicated approximation ratio. How would you choose ε', δ in order to achieve a $(1+\varepsilon)$-approximation in the end, with best possible runtime?

Hint: Use Lemma 2.2.3 to compare c'_i with the actual center c_i of $B(S_i)$. For the analysis, work with c_i and establish adapted versions of the bounds in (2.9) and (2.10).

Exercise 11 *Establish a notion of $(1+\varepsilon)$-approximation for bounding boxes $Q(P) \subseteq \mathbb{R}^d$. Can you always find a subset $S \subseteq P$ whose size only depends on ε such that the center of $Q(S)$ is a $(1+\varepsilon)$-approximation of $Q(P)$?*

Chapter 3

Quadratic Programming

In this chapter, we show that the problem of computing the smallest enclosing ball (as well as another interesting problem) can be formulated as a *quadratic program* (QP). The implications are twofold. On the one hand, there are (at least practically) efficient algorithms for computing (approximate) solutions to QP, even in high dimensions; on the other hand, the QP approach shows that the smallest enclosing ball is fully determined by the n^2 pairwise scalar products of the n input points. We will make use of this surprising fact in the next chapter.

3.1 Simple convex programming

A function $f : \mathbb{R}^m \to \mathbb{R}$ is called *convex*, if for all $x, x' \in \mathbb{R}^m$ and all $\lambda \in [0,1]$,

$$f((1-\lambda)x + \lambda x') \leq (1-\lambda)f(x) + \lambda f(x'). \tag{3.1}$$

Geometrically, this means that any segment connecting two points on the *graph* of f lies above the graph, see Figure 3.1. The graph of f is the set $\{(x, f(x)) \mid x \in \mathbb{R}^m\} \subseteq \mathbb{R}^{m+1}$, and if the function is convex, the graph looks like a 'bowl'.

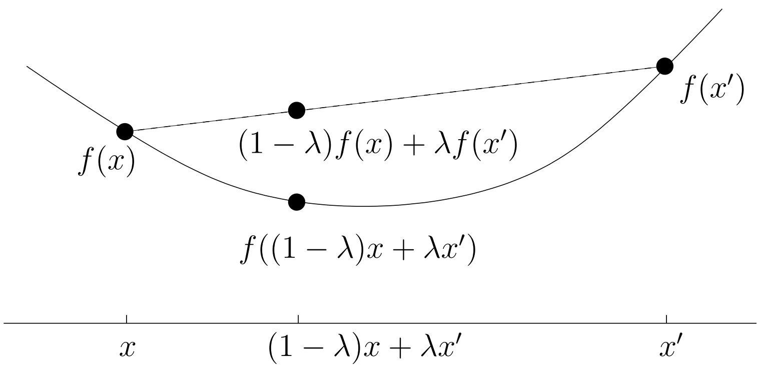
$f(x')$
$(1-\lambda)f(x) + \lambda f(x')$
$f(x)$
$f((1-\lambda)x + \lambda x')$
x
$(1-\lambda)x + \lambda x'$
x'

Figure 3.1: A convex function

We do not require strict inequality in (3.1), so the graph of f may be 'flat' in certain parts. In particular, a linear function (whose graph can be considered as a nonvertical hyperplane in $\mathbb{R}^{m+1}$) is convex. Convex functions are continuous.

In the sequel, we will deal with differentiable convex functions that have continuous partial derivatives. In this case, we have

Fact 3.1.1 *f is convex if and only if*

$$f(x) + \nabla f(x)(x' - x) \leq f(x'),$$

for all $x, x' \in \mathbb{R}^m$, where $\nabla f : \mathbb{R}^m \to \mathbb{R}^m$ is the gradient operator (whose values are row vectors by convention).

For a proof (and other interesting material about convex functions), see the very nice book by Peressini, Sullivan and Uhl [7]. Geometrically, this fact says that f is convex if and only if all tangential hyperplanes to the graph of f are below the graph.

Fixing f, we now consider the *simple convex program*

$$\begin{array}{lll} \text{(SCP)} & \text{minimize} & f(x) \\ & \text{subject to} & \sum_{i=1}^m x_i = 1, \\ & & x_i \geq 0, \quad i = 1, \ldots, m. \end{array}$$

This means, we are looking for a point $x \in \mathbb{R}^m$ that satisfies the *constraints*

$$\sum_{i=1}^{m} x_i = 1, \quad x_i \geq 0, i = 1, \ldots, m$$

and has minimum function value among all such points. Any point satisfying the constraints is a *feasible solution*, and the set of all feasible solutions is the *feasible region*. It is an $m-1$-dimensional *simplex* (see Figure 3.2); in particular, it is a nonempty compact set, and so f as a continuous function does have a minimum over this simplex. Any feasible solution x for which $f(x)$ achieves this minimum value is called a *minimizer* of f over the feasible region, or an *optimal solution* to (SCP).

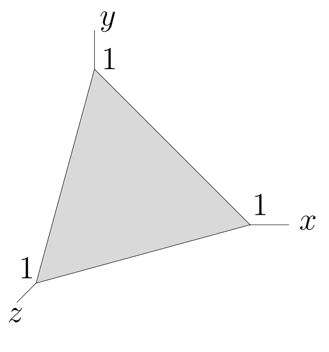

Figure 3.2: The feasible region of (SCP), $m = 3$

The convexity of f lets us characterize the set of optimal solutions to (SCP). Because the feasible region of (SCP) is convex, the following is actually a more general statement.

Lemma 3.1.2 *Let $C \subseteq \mathbb{R}^m$ be a nonempty convex set. $x \in C$ is a minimizer of f over C if and only if for all $x' \in C$,*

$$\nabla f(x)(x' - x) \geq 0. \tag{3.2}$$

Proof. If condition (3.2) holds, Fact 3.1.1 yields for all $x' \in C$

$$f(x') \geq f(x) + \nabla f(x)(x' - x) \geq f(x),$$

so x is a minimizer of f over C. Vice versa, assume that x is a minimizer and choose $x' \in C$. By convexity, we have $x(\lambda) = (1 - \lambda)x + \lambda x' \in C$, $\lambda \in [0, 1]$, and because x is a minimizer of f over C, we obtain

$$\frac{\partial}{\partial \lambda} f(x(\lambda))|_{\lambda=0} := \lim_{\lambda \searrow 0} \frac{f(x(\lambda)) - f(x)}{\lambda} \geq 0. \tag{3.3}$$

On the other hand, the chain rule yields

$$\frac{\partial}{\partial \lambda} f(x(\lambda)) = \frac{\partial}{\partial \lambda} f((1 - \lambda)x + \lambda x') = \nabla f(x(\lambda))(x' - x).$$

With $\lambda = 0$ and (3.3), condition (3.2) follows. $\square$

In our specific case, we can refine the optimality condition (3.2) as follows.

Lemma 3.1.3 *Let $x \in \mathbb{R}^m$ be a feasible solution to (SCP). Then x is an optimal solution to (SCP) if and only if for all $i = 1, \ldots, m$,*

$$\nabla f(x)(\mathbf{e}_i - x) \begin{cases} = 0, & \text{if } x_i > 0, \\ \geq 0 & \text{otherwise}, \end{cases} \tag{3.4}$$

where $\mathbf{e}_i$ is the i-th unit vector.

Proof. If x is optimal, '≥ 0' in (3.4) follows from (3.2), because $x' = \mathbf{e}_i$ is a feasible solution. If $x_i > 0$, there exists $\varepsilon > 0$ such that $x' = (1 - \lambda)x + \lambda \mathbf{e}_i$ is feasible for all $\lambda \in [-\varepsilon, \varepsilon]$. From Lemma 3.1.2, we then get

$$0 \leq \nabla f(x)(x' - x) = \lambda \nabla f(x)(\mathbf{e}_i - x).$$

As this in particular holds for $\lambda = -\varepsilon, \varepsilon$, we must have $\nabla f(x)(\mathbf{e}_i - x) = 0$.

If, on the other hand, (3.4) holds for all i, we get for any other feasible solution x' that

$$\nabla f(x)(x' - x) = \nabla f(x) \left(\sum_{i=1}^{m} x'_i \mathbf{e}_i - x \right) = \sum_{i=1}^{m} x'_i \nabla f(x)(\mathbf{e}_i - x) \geq 0,$$

because all coordinates x'_i of x' are nonnegative. With Lemma 3.1.2, it follows that x is optimal. $\square$

3.2 Smallest enclosing balls

We now show that the problem of finding the smallest enclosing ball of an n-point set $P \subseteq \mathbb{R}^d$ can be formulated in the form of (SCP), with f being a *quadratic* function. In this case, (SCP) is called a *quadratic program.*

Theorem 3.2.1 *Let $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d$. Let Q be the $(d \times n)$-matrix whose columns are the points of P, i.e.*

$$Q = \begin{pmatrix} p_{11} & \cdots & p_{n1} \\ \vdots & & \vdots \\ p_{1d} & \cdots & p_{nd} \end{pmatrix}. \tag{3.5}$$

Then the following statements hold.

(i) $f : \mathbb{R}^n \to \mathbb{R}$ defined by

$$f(x) = x^T Q^T Q x - \sum_{i=1}^n p_i^T p_i x_i$$

is a convex function (Exercise 12). [1]

(ii) Let x^* be any optimal solution to (SCP) with the function f from (i). Then the point

$$c^* = Qx^*$$

is the center c_P of the smallest enclosing ball $B(P)$ of P, and the value $-f(x^*)$ is the squared radius R_P^2 of $B(P)$.

Proof. We only prove (ii) here, assuming (i). We first show that c^* is the center of an enclosing ball with squared radius $-f(x^*)$. Using

$$\nabla f(x) = 2x^T Q^T Q - (p_1^T p_1, \ldots, p_n^T p_n)$$

(you might want to check this), Lemma 3.1.3 yields

$$\begin{aligned} 0 &\leq \nabla f(x^*)(\mathbf{e}_i - x^*) \\ &= \left(2x^{*T} Q^T Q - (p_1^T p_1, \ldots, p_n^T p_n)\right)(\mathbf{e}_i - x^*) \\ &= \left(2c^{*T} Q - (p_1^T p_1, \ldots, p_n^T p_n)\right)(\mathbf{e}_i - x^*) \\ &= \left(2c^{*T} p_i - p_i^T p_i\right) - \left(2c^{*T} c^* - (p_1^T p_1, \ldots, p_n^T p_n) x^*\right) \\ &= \left(2c^{*T} p_i - p_i^T p_i - c^{*T} c^*\right) - \left(c^{*T} c^* - \sum_{i=1}^n p_i^T p_i x_i^*\right) \\ &= -(p_i - c^*)^T (p_i - c^*) - f(x^*). \end{aligned} \tag{3.6}$$

[1]For $x, y \in \mathbb{R}^d$, the expression $x^T y$ is just a different way of writing the scalar product $x \cdot y$.

In other words,

$$\|p_i - c^*\|^2 \leq -f(x^*), \quad i = 1, \ldots, n,$$

and this proves that $B_d(c^*, \sqrt{-f(x^*)})$ is a ball containing all points of P. It remains to prove that there is no smaller ball with this property.

Given a potential center $c \neq c^*$, we can uniquely write it in the form

$$c = c^* + \lambda u,$$

where u is some vector of length 1 and $\lambda > 0$. Define $r^2 := -f(x^*)$ and

$$F := \{i \in \{1, \ldots, n\} \mid x_i^* > 0\}.$$

F is nonempty because $\sum_{i=1}^n x_i^* = 1$. With $i \in F$, we get

$$\begin{array}{rcl} (p_i - c)^T(p_i - c) & = & (p_i - c^* - \lambda u)^T(p_i - c^* - \lambda u) \\ & = & (p_i - c^*)^T(p_i - c^*) + \lambda^2 u^T u - 2\lambda u^T(p_i - c^*) \\ & = & r^2 + \lambda^2 - 2\lambda u^T(p_i - c^*), \end{array}$$

where the last equality holds because $x_i^* > 0$ implies equality in (3.6) via Lemma 3.1.3.

In order for c to define a ball of squared radius at most r^2 which contains all points—in particular the points $p_i, i \in F$—we must have

$$u^T(p_i - c^*) > 0, \quad i \in F, \tag{3.7}$$

because whenever this fails for some $i \in F$, we get $(p_i - c)^T(p_i - c) > r^2$.

Using $x_i^* > 0, i \in F$, inequality (3.7) then yields

$$\sum_{i \in F} x_i^* u^T(p_i - c^*) > 0.$$

On the other hand, using $\sum_{i \in F} x_i^* = 1$ and $c^* = Qx^* = \sum_{i=1}^n x_i^* p_i = \sum_{i \in F} x_i^* p_i$, we have

$$\sum_{i \in F} x_i^* u^T(p_i - c^*) = u^T \left(\sum_{i \in F} x_i^* p_i - \sum_{i \in F} x_i^* c^* \right) = u^T(c^* - c^*) = 0,$$

a contradiction. This means, c cannot define an enclosing ball of squared radius at most r^2, so c^* is indeed the center of $B(P)$, and $r^2 = -f(x^*)$ is its squared radius. □

The proof actually shows that all points $p_i, i \in F$, are *on the boundary* of $B(P)$, because they satisfy $\|p_i - c^*\|^2 = -f(x^*)$, due to equality in (3.6). Even more is true: the equations

$$c^* = \sum_{i \in F} x_i^* p_i, \quad \sum_{i \in F} x_i^* = 1,$$

along with $x_i^* > 0, i \in F$ show that c^* is a *convex combination* of the $p_i, i \in F$, see Lemma 1.4.1(iii). This means, the solution x^* of the convex program does not only give us $B(P)$, it also provides us with a set of boundary points whose convex hull contains c^*, see Figure 3.3.

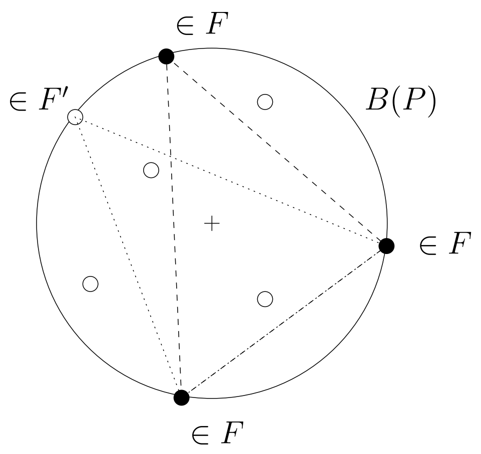

Figure 3.3: P and $B(P)$, with the points $p_i, i \in F$ filled and their convex hull dashed; x^* is not unique in general: there is another optimal solution x'^* defining a set $F' \neq F$ (dotted convex hull)

3.3 Quadratic programming

There are methods for approximately solving (SCP) with a quadratic function efficiently in practice, even for large d and n; these methods are *heuristics*, though, and they do not come with an approximation guarantee. In certain cases, there are fast methods for which some quality of the solution can be guaranteed. This in particular applies to the situation in which the matrix Q from (3.5) is *sparse*, meaning that it has only few nonzero entries. All these methods work for more general problems than we have discussed so far. A general convex quadratic program assumes the form

$$\text{(QP)} \quad \begin{array}{ll} \text{minimize} & x^T D x + c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0, \end{array}$$

where $D \in \mathbb{R}^{m \times m}$ is a *positive-semidefinite matrix*,[2], $c \in \mathbb{R}^m$, $A \in \mathbb{R}^{k \times m}$ any matrix, and $b \in \mathbb{R}^k$. $x \geq 0$ is a shortcut for $x_i \geq 0, i = 1, \ldots, m$. Optimality conditions similar to the ones in Lemma 3.1.3 exist, and the main reason why quadratic programs are still 'easy' to solve is that the gradient of a quadratic function is linear. General convex programs (CP), where the quadratic function of (QP) is replaced with an arbitrary convex function, still have nice optimality conditions, but these are nonlinear in general, making the problem much harder to solve in practice.

[2]meaning that $x^T D x \geq 0$ for all x

3.4 Polytope Distance

There is another problem which is solvable via the quadratic programming approach. Let $P, Q \subseteq \mathbb{R}^d$ be two point sets. The polytope distance problem is concerned with the computation of the distance between $\text{conv}(P)$ and $\text{conv}(Q)$,

$$\delta(P, Q) = \min\{\|p - q\| \mid p \in \text{conv}(P), q \in \text{conv}(Q)\},$$

see Figure 3.4.[3]

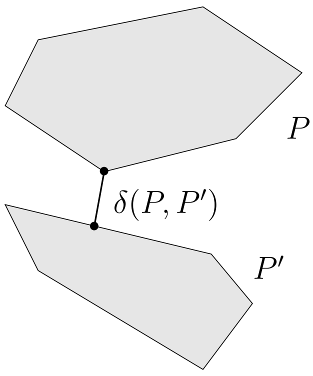

Figure 3.4: The polytope distance problem

If we can compute $\delta(P, Q)$, we can in particular decide whether the convex hulls intersect—this is the case if and only if $\delta(P, Q) = 0$.

Exercise 14 asks you to write down a formulation of the polytope distance problem in the form of (QP) and to show that the shortest difference vector $p - q, p \in \text{conv}(P), q \in \text{conv}(Q)$ is unique.

3.5 It's all about scalar products

Looking at the problem (SCP) with the function f used for smallest enclosing balls (Theorem 3.2.1), we see that the actual coordinates of the points p_i are never needed. The entries of the matrix Q^TQ are scalar products of the form $p_i \cdot p_j$, and there is an additional linear term in f involving the scalar products $p_i \cdot p_i$. This implies

Corollary 3.5.1 *Let $P \subseteq \mathbb{R}^d$, $|P| = n$. The smallest enclosing ball $B(P)$ can be computed in time*

$$O(dn^2 + g(n)),$$

where $g(n)$ is the time necessary to solve (SCP) with the function f of Theorem 3.2.1.

[3] $\delta(P, Q)$ exists, because we are minimizing a continuous function $g : \mathbb{R}^{2d} \to \mathbb{R}$ (which one?) over a compact set $K \subseteq \mathbb{R}^{2d}$ (which one?)

Proof. It takes $O(dn^2)$ time to compute all scalar products, $g(n)$ time to solve (SCP) and another $O(dn)$ time to compute $B(P)$ from the solution, according to Theorem 3.2.1(ii). □

If $n \ll d$, this is a major improvement over other exact methods. Exercise 15 gives a scenario, where the dependence on d can be removed altogether.

3.6 Exercises

Exercise 12 *Prove Theorem 3.2.1 (i).*

Exercise 13 *Use the arguments in the proof of Theorem 3.2.1(ii) to prove Lemma 2.2.2.*

Exercise 14 *Formulate the polytope distance problem in the form of a quadratic program (QP). Can you do it in such a way that the formulation only involves* scalar products *of points from $\mathbb{R}^d$? Also, prove that the shortest difference vector $p - q, p \in \operatorname{conv}(P), q \in \operatorname{conv}(Q)$ is unique.*

Exercise 15 *The* moment curve *in $\mathbb{R}^d$ is the point set*

$$M_d = \{x \in \mathbb{R}^d \mid x = (t, t^2, \ldots, t^d), t \in \mathbb{R}\}.$$

Let $P \subseteq M_d, |P| = n$. Prove that the radius R_P of $B(P)$ can be computed in time independent from d.

Chapter 4

Cuboids

We have already seen that we can efficiently find the bounding box $Q(P)$ and an arbitrarily good approximation to the smallest enclosing ball $B(P)$ of a set $P \subseteq \mathbb{R}^d$. Unfortunately, both bounding volumes are bad when the task is to approximate the volume of $\mathrm{conv}(P)$. As you can see from Figures 2.1 and 2.2, the ratios

$$\frac{\mathrm{vol}(Q(P))}{\mathrm{vol}(\mathrm{conv}(P))} \text{ and } \frac{\mathrm{vol}(B(P))}{\mathrm{vol}(\mathrm{conv}(P))}$$

can get arbitrarily large even for $d = 2$, and if $\mathrm{conv}(P)$ has nonzero volume: as the points in P get closer and closer to some fixed—nonvertical and nonhorizontal—line segment, the volume of the convex hull becomes smaller and smaller, while the volumes of $Q(P)$ and $B(P)$ converge to nonzero constants.

In this chapter, we show that boxes of *arbitrary orientations* are better with respect to volume. To distinguish them from the (axis-parallel) boxes, we call them *cuboids*, see Figure 4.1. Formally, a cuboid is any set of the form

$$C = \{Mx \mid x \in Q_d(\underline{b}, \overline{b}), M^{-1} = M^T\}. \tag{4.1}$$

A matrix M with $M^{-1} = M^T$ is called *orthogonal*, and in the orthogonal coordinate system defined by the columns of M, C is an axis-parallel box.

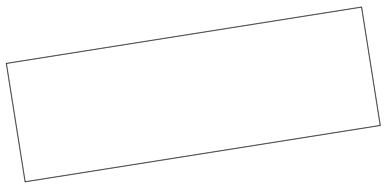

Figure 4.1: A cuboid in $\mathbb{R}^2$

4.1 Approximating the smallest enclosing cuboid

The exact smallest enclosing cuboid $C(P)$ of set $P \subseteq \mathbb{R}^d$ can be computed in time $O(n \log n)$ for $d = 2$ and $O(n^3)$ for $d = 3$. No better algorithms are known. In contrast, $Q(P)$ and $B(P)$ can be computed in optimal time $O(n)$ for $d = 2, 3$ [10]. This already shows that $C(P)$ is a more complicated object then $B(P)$, and that we should be more modest regarding the quality of approximation we can expect. Actually, $C(P)$ is not even well-defined, because there is not necessarily a unique smallest enclosing cuboid, see Figure 4.2. When we are writing $C(P)$, we therefore mean *some* smallest enclosing cuboid, and with a little care, this poses no problems. For example, the quantity $\text{vol}(C(P))$ is well-defined, because it does not depend on the particular choice of $C(P)$.

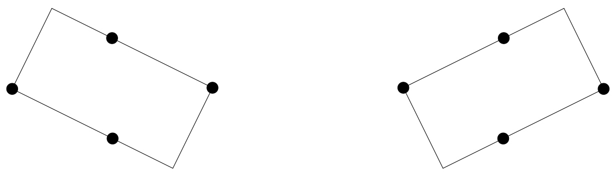

Figure 4.2: A set may have more than one smallest enclosing cuboid

Here is the main result of this section.

Theorem 4.1.1 *For $P \subseteq \mathbb{R}^d$ with $|P| = n$, we can compute in $O(d^2 n)$ time a cuboid C that contains P and satisfies*

$$\text{vol}(C) \leq 2^d d! \text{vol}(C(P)).$$

This means, for any constant dimension d, we can approximate the volume of the smallest cuboid that contains P up to some constant factor; however, this factor is already pretty bad when the dimension is only moderately high. On the other hand, the result is not as bad as it might look like, and the exponential dependence on d seems very hard to avoid.

To see this, let's look at smallest enclosing balls again. In Chapter 2, we have shown that we can find an enclosing ball B of P whose radius is at most $(1 + \varepsilon)$ times the radius of $B(P)$, for any $\varepsilon > 0$. With respect to *volume*, this means that

$$\text{vol}(B) \leq (1 + \varepsilon)^d \text{vol}(B(P)),$$

which is exponential in d. In view of this, the bound in Theorem 4.1.1 starts to look somewhat more reasonable.

In order to prove the theorem, we first describe an algorithm for computing some bounding cuboid C and then argue about the quality of C. Actually, the algorithm

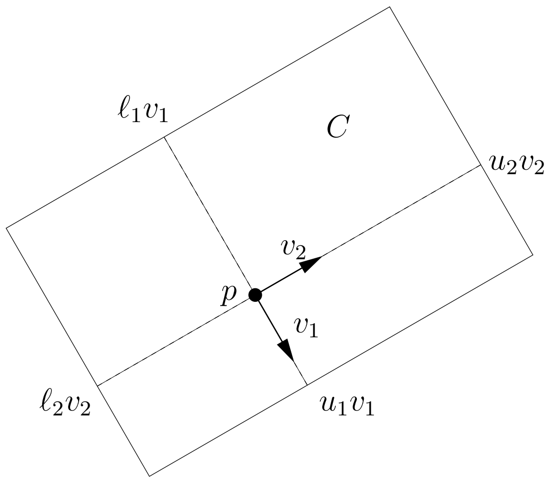

Figure 4.3: Cuboid defined by the algorithm, $d = 2$

computes a point $p \in \mathbb{R}^d$, a set of *pairwise orthogonal*[1] unit vectors $v_1, \ldots, v_d$, and numbers $\ell_k \leq u_k, k = 1, \ldots, d$ such that

$$C = \{p + \sum_{k=1}^{d} \lambda_k v_k \mid \ell_k \leq \lambda_k \leq u_k, k = 1, \ldots, d\}, \tag{4.2}$$

see Figure 4.3. Exercise 16 asks you to verify that this set C is indeed a cuboid.

We will assume that $\mathbf{0} \in P$. This can be achieved through a translation of P in time $O(dn)$. Now we call the following recursive algorithm with parameters (P, d), meaning that the global invariant holds in the beginning. The algorithm constructs v_k, l_k, u_k one after another, starting with $k = d$. The point p used in (4.2) will be $p = 0$.

```
MinCuboid_Approx($P, k$):
    (* Global invariant: $p \cdot v_i = 0$ for $p \in P, i = k+1, \ldots, d$ *)
    choose $q \in P$ such that $\|q\|$ is maximum
    IF $q \neq \mathbf{0}$ THEN
        $v_k = q/\|q\|$
        $\ell_k = \min_{p \in P} p \cdot v_k$
        $u_k = \|q\|$
        $P' = \{p - (p \cdot v_k)v_k \mid p \in P\}$
        (* Local invariant: $p' \cdot v_k = 0$ for $p' \in P'$ *)
        MinCuboid_Approx($P', k-1$)
    END
```

Before we analyze the algorithm, let us try to get a geometric intuition for the top-level call with $k = d$ (Figure 4.4). The vector v_d is a unit vector pointing in the direction

[1]Two vectors v, w are orthogonal if $v \cdot w = 0$.

of some point q farthest away from $\mathbf{0} \in P$. Because $p \cdot v_d$ is the (signed) length of p's projection onto the direction v_d, the value $u_d - \ell_d = \|q\| - \ell_d$ is the extent of P along direction v_d. P' is the projection of P onto the unique hyperplane h through the origin that is orthogonal to v_d. For P', the recursive call finds a $(d-1)$-dimensional bounding cuboid C' within h, which we then extend along direction v_d to a cuboid C containing P.

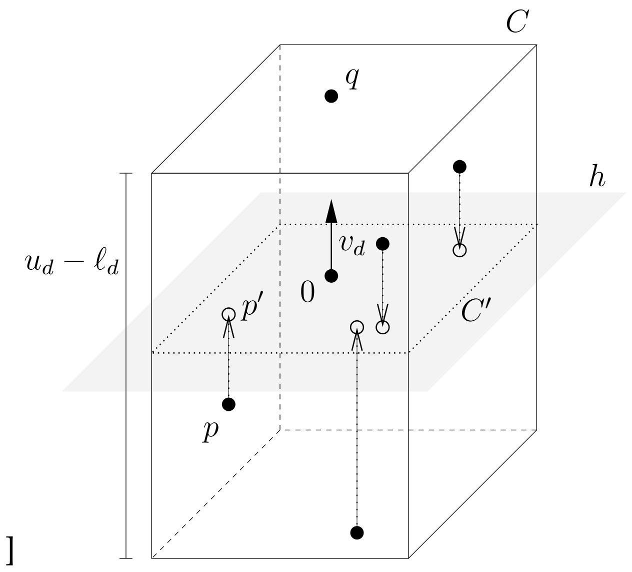

Figure 4.4: Geometric picture of the algorithm

4.1.1 Correctness and runtime of the algorithm.

Because $\|v_k\| = 1$ and $p' \cdot v_k = p \cdot v_k - (p \cdot v_k)\|v_k\|^2 = 0$, the local invariant holds for all $p' \in P'$. The global invariant also holds in the recursive call, because of the just established local invariant, and because the global invariant for $p, q \in P$ shows that for $i = k+1, \ldots, d$,

$$\begin{aligned} p' \cdot v_i &= (p - (p \cdot v_k)v_k) \cdot v_i \\ &= \underbrace{p \cdot v_i}_{=0} - (p \cdot v_k)(v_k \cdot v_i) = -(p \cdot v_k)\underbrace{(q \cdot v_i)}_{=0}/\|q\| = 0. \end{aligned}$$

The latter equation also shows that $v_k \cdot v_i = 0, i = k+1, \ldots, d$ which yields the pairwise orthogonality of all the v_k. Because there are only d pairwise orthogonal nonzero vectors, the recursion bottoms out for some value $\underline{k} \geq 0$. This implies the runtime of $O(d^2 n)$.

To prove that the cuboid

$$C = \{ \sum_{i=\underline{k}+1}^{d} \lambda_i v_i \mid \ell_i \leq \lambda_i \leq u_i, i = \underline{k}+1, \ldots, d\} \tag{4.3}$$

contains P, we proceed by induction on d. For $d = 0$, or if the condition of the `IF` clause fails, we have $P = C = \{\mathbf{0}\}$ and $\underline{k} = d$, so the claim holds. Now assume $d > 0$, and we have already proved the claim for $d - 1$. Inductively, we then know that all points $p' \in P'$ can be written in the form

$$p' = \sum_{i=k_0+1}^{d-1} \lambda_i v_i, \quad \ell_i \leq \lambda_i \leq u_i, i = \underline{k} + 1, \ldots, d - 1. \tag{4.4}$$

Furthermore, the definition of p' yields

$$p = p' + (p \cdot v_d) v_d, \tag{4.5}$$

where

$$\ell_d \leq p \cdot v_d \leq |p \cdot v_d| \leq \|p\| \|v_d\| = \|p\| \leq \|q\| = u_d,$$

by the Cauchy-Schwarz inequality. Therefore, setting $\lambda_d = (p \cdot v_d)$ and plugging (4.4) into (4.5) gives the desired conclusion $p \in C$.

4.1.2 Quality of the algorithm

It remains to bound the volume of the cuboid C resulting from the algorithm according to (4.3). If the value of $\underline{k}$ for which the recursion bottoms out satisfies $\underline{k} > 0$, we have $\mathrm{vol}(C) = 0$ and are done, so we will assume that $\underline{k} = 0$.

Let $q_k, k = 1, \ldots, d$ be the point chosen in the recursive call with second parameter k, and let $p_k \in P, k = 1, \ldots, d$ be the point of P whose (iterated) projection q_k is. We have

$$q_d = p_d, \quad q_{d-1} = p_{d-1} - (p_{d-1} \cdot v_d) v_d,$$

and using the pairwise orthogonality of the v_k, we can inductively verify the general formula

$$q_k = p_k - \sum_{i=k+1}^{d} (p_k \cdot v_i) v_i, \quad k = 1, \ldots, d. \tag{4.6}$$

The approximation ratio of Theorem 4.1.1 is derived using two ingredients: a *lower* bound on the volume of the smallest enclosing cuboid $C(P)$, and an *upper* bound on the volume of the cuboid C computed by the algorithm.

A lower bound for $\mathrm{vol}(C(P))$. We consider the two sets

$$S_q = \mathrm{conv}(\{\mathbf{0}, q_d, \ldots, q_1\}), \quad S_p = \mathrm{conv}(\{\mathbf{0}, p_d, \ldots, p_1\}).$$

Because the q_i are pairwise orthogonal, they are in particular linearly independent. In this case, we can apply the well-known formula for the volume of a *simplex* in $\mathbb{R}^d$ to obtain

$$\mathrm{vol}(S_q) = \frac{1}{d!} |\det(q_d, \ldots, q_1)|.$$

On the other hand, (4.6) shows that q_k equals p_k plus some linear combination of the $q_i, i > k$ (note that v_i is a multiple of q_i). Recalling that the determinant of a set of vectors does not change when we add to some vector a linear combination of the other vectors, we consequently get (how?) that

$$\det(q_d, \ldots, q_1) = \det(p_d, \ldots, p_1),$$

and therefore

$$\mathrm{vol}(S_p) = \mathrm{vol}(S_q) = \frac{1}{d!} \prod_{k=1}^{d} \|q_k\|,$$

using orthogonality of the q_k. Geometrically, this shows that we can transform the simplex S_q into the simplex S_p, by always moving one of the points *parallel* to its opposite side. During such a movement, the volume stays constant, see Figure 4.5.

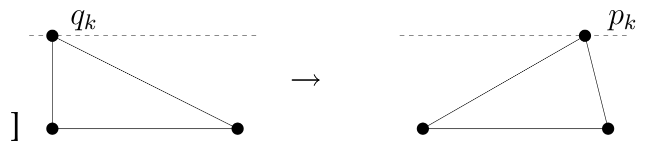

Figure 4.5: Moving a vertex parallel to the opposite side does not change the volume

Now we have a lower bound for the volume of the smallest cuboid $C(P)$. Because $S_p \subseteq \mathrm{conv}(P) \subseteq C(P)$, we get

$$\mathrm{vol}(C(P)) \geq \mathrm{vol}(\mathrm{conv}(P)) \geq \mathrm{vol}(S_p) = \frac{1}{d!} \prod_{k=1}^{d} \|q_k\|. \tag{4.7}$$

An upper bound for $\mathrm{vol}(C)$**.** By construction of C, we have

$$\mathrm{vol}(C) = \prod_{k=1}^{d} (u_k - \ell_k),$$

because $u_k - \ell_k$ is the extent of C in direction of the unit vector v_k. Let $\underline{p}$ be the point defining ℓ_k in the recursive call of the algorithm with second parameter k. We get

$$\begin{array}{rcll} u_k - \ell_k & \leq & |u_k| + |\ell_k| = \|q_k\| + |\underline{p} \cdot v_k| & \\ & \leq & \|q_k\| + \|\underline{p}\| \|v_k\| & \text{(Cauchy-Schwarz inequality)} \\ & = & \|q_k\| + \|\underline{p}\| & (\|v_k\| = 1) \\ & \leq & 2\|q_k\|. & \text{(choice of } q\text{)} \end{array}$$

It follows that

$$\mathrm{vol}(C) \leq 2^d \prod_{k=1}^{d} \|q_k\| \overset{(4.7)}{\leq} 2^d d! \mathrm{vol}(C(P)),$$

which is what we wanted to prove in Theorem 4.1.1.

Looking at (4.7), we see that we have actually proved a little more.

Corollary 4.1.2 *Let C be the cuboid computed by a call to* `MinCuboid_Approx`(P, d)*,* $P \subseteq \mathbb{R}^d$*. Then*

$$\frac{\mathrm{vol}(C(P))}{\mathrm{vol}(\mathrm{conv}(P))} \leq \frac{\mathrm{vol}(C)}{\mathrm{vol}(\mathrm{conv}(P))} \leq 2^d d!.$$

We therefore have shown that $C(P)$ is strictly better than $Q(B)$ and $B(P)$ when it comes to approximating the volume: the volume of any smallest enclosing cuboid of P is only by a *constant factor* (depending on d) larger than the volume of $\mathrm{conv}(P)$. The same factor even holds for the cuboid C that we can easily compute in $O(d^2 n)$ time.

This cuboid satisfies an additional property that we mention without proof. This goes back to a paper by Barequet and Har-Peled [**?**] that also contains a sketch of the algorithm `MinCuboid_Approx`.

Theorem 4.1.3 *Let C be the cuboid computed by a call to* `MinCuboid_Approx`(P, d)*,* $P \subseteq \mathbb{R}^d$*. There exists a constant $\alpha_d > 0$ (only depending on d) such that C, scaled with α_d and suitably translated, is contained in* $\mathrm{conv}(P)$*, see Figure 4.6.*

This means that we will find a not-too-small copy of C inside $\mathrm{conv}(P)$, and this is exploited in a number of approximation algorithms for other problems.

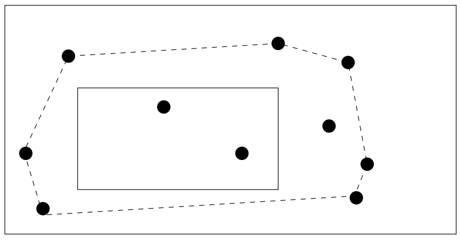

Figure 4.6: Bounding and *inscribed* cuboid of the same shape

The proof of Theorem 4.1.3 (and similar statements) involves properties of an even nicer family of bounding volumes that we introduce next.

4.2 Ellipsoids

Definition 4.2.1

(i) An ellipsoid *in $\mathbb{R}^d$ is any set of the form*

$$E = \{x \in \mathbb{R}^d \mid (x-c)^T A(x-c) \leq z,$$

where $c \in \mathbb{R}^d$ is the center *of the ellipsoid, A is a positive definite matrix,*[2] *and $z \in \mathbb{R}$.*

[2] $x^T Ax > 0$ for $x \neq 0$.

(ii) *For $\lambda \geq 0$, the* scaled *ellipsoid λE is the set*

$$\lambda E = \{c + \lambda(x - c) \mid x \in E\}.$$

This scales E relative to its center, and it is easy to prove (Exercise 17) that λE is an ellipsoid again.

If A is the identity matrix, for example, E is a ball with center c and squared radius z. In general, any ellipsoid is the affine image of a ball, and the image of any ellipsoid under a nonsingular affine transformation is again an ellipsoid (Exercise 18).

Here is what makes ellipsoids attractive as bounding (and also as inscribed) volumes. This is classic material [**?**].

Theorem 4.2.2 *Let $K \subseteq \mathbb{R}^d$ be a convex body (this is a convex and compact set of positive volume).*

(i) *There is a unique ellipsoid $\overline{E}(K)$ of smallest volume such that $K \subseteq \overline{E}(K)$ and a unique ellipsoid $\underline{E}(K)$ of largest volume such that $\underline{E}(K) \subseteq K$.*

(ii) $1/d\,\overline{E}(K) \subseteq K$

(iii) $d\underline{E}(K) \supseteq K$.

(iv) *If K is* centrally symmetric *($p \in K \Rightarrow -p \in K$), we have $1/\sqrt{d}\;\overline{E}(K) \subseteq K$ and $\sqrt{d}\underline{E}(K) \supseteq K$.*

This is a theorem, similar in spirit to Theorem 4.1.3. The difference is that the scaling factors are explicit (and small), and that no additional translation is required. The theorem says that any convex body is 'wedged' between two concentric ellipsoids whose scale only differs by a factor of d at most.

4.3 Exercises

Exercise 16 *Prove (using the definition 4.1) that for any $p \in \mathbb{R}^d$, any set of pairwise orthogonal vectors $v_1, \ldots, v_d$, and any numbers $\ell_k \leq u_k$, $k = 1, \ldots, d$, the set*

$$C = \{p + \sum_{i=1}^{d} \lambda_k v_k \mid \ell_k \leq \lambda_k \leq u_k, k = 1, \ldots, d\}$$

is a cuboid.

Exercise 17 *Prove that the set λE from Definition 4.2.1 is an ellipsoid again.*

Exercise 18 *Prove that any ellipsoid is the image of a ball under some affine transformation, and that the image of any ellipsoid under a nonsingular affine transformation is again an ellipsoid.*

Chapter 5

Support Vector Machines

Support vector machines are universal tools in machine learning, where they are used for almost every task imaginable. Still the most prominent application for support vector machines is discriminant analysis. In discriminant analysis we are given labeled training data, where the label indicates to which class a datum belongs. The task is to compute a classifier from the labeled training data that allows to categorize new data, i.e., attach a label to it. In a first phase, we want to focus on geometric aspects of this task.

5.1 Maximum Margin Hyperplane

Here we study a version of the discriminant problem, where we assume that the data are points in the Euclidean space $\mathbb{E}^d$ and that there are only two classes P and Q with labels 1 and -1, respectively. We assume that $P \cup Q = \{x_1, \ldots, x_n\}$. At first we want to further assume that the two classes are linearly separable, i.e., that there exists a hyperplane that has all points with negative label strictly on one side and all points with positive label strictly on the other side. Such a hyperplane is given by two parameters, a unit normal $\tilde{w} \in \mathbb{E}^d$ and an offset $\tilde{b} \in \mathbb{R}$. Thus we have

$$\begin{aligned} \tilde{w}^T p_i - \tilde{b} &> 0, \quad p_i \in P \\ \tilde{w}^T p_i - \tilde{b} &< 0, \quad p_i \in Q. \end{aligned}$$

The classifier associated with a hyperplane $h = \{x \in \mathbb{E}^d \,|\, w^T x = b\}$ is the function $\operatorname{sign}(w^T x - b)$, where $\operatorname{sign}(z) = 1$ if $z > 0$, $\operatorname{sign}(z) = -1$ if $z < 0$ and $\operatorname{sign}(0) = 0$. Among hyperplanes that separate P and Q we are looking for one that has a *maximal margin*, i.e., a hyperplane that we can move the farthest in both directions along the normal $\tilde{w}$ before we meet a point from either P or Q. The naive intuition that the separation by such a hyperplane has good generalization properties, i.e., unseen data are likely to fall on the right side of the hyperplane, can be made more precise in statistical learning theory. Here we simply postulate that it is desirable to have a separating hyperplane with large margin, see also Figure 5.1. In order to compute the margin of a hyperplane

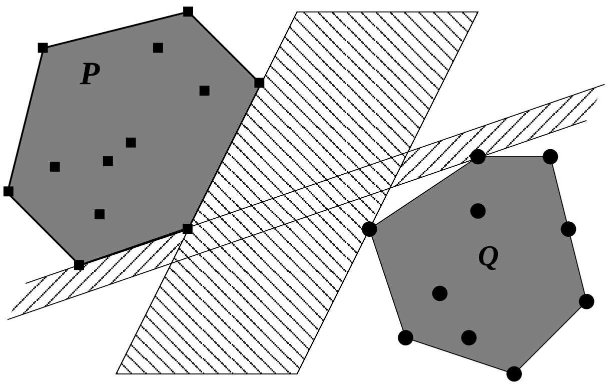

Figure 5.1: A large (good) and and a small (bad) margin.

let

$$c := \min_{p_i \in P \cup Q} |\tilde{w}^T p_i - \tilde{b}| > 0.$$

That is,

$$\begin{array}{rcll} \tilde{w}^T p_i - \tilde{b} & \geq & c, & p_i \in P \\ \tilde{w}^T p_i - \tilde{b} & \leq & -c, & p_i \in Q. \end{array}$$

By scaling the normal $\tilde{w}$ and the offset $\tilde{b}$ by $1/|c|$ we get

$$\begin{array}{rcll} w^T p_i - b & \geq & 1, & p_i \in P \\ w^T p_i - b & \leq & -1, & p_i \in Q, \end{array}$$

where $w := \tilde{w}/|c|$ and $b := \tilde{b}/|c|$. The margin of the hyperplane described by $\tilde{w}$ and $\tilde{b}$ is defined as the distance between the hyperplanes $h = \{x \in \mathbb{E}^d \mid w^T x = b + 1\}$ and $h' = \{x \in \mathbb{E}^d \mid w^T x = b - 1\}$. From the material in Section 1.2, it is easy to deduce that this distance is $2/\|w\|$. In order to maximize the margin we can therefore minimize $\|w\|/2$. This leads to the convex quadratic program

$$\begin{array}{llcll} \min_{w,b} & \frac{1}{2} w^T w & & & \\ \text{s.t.} & w^T p_i - b & \geq & 1, & p_i \in P, \\ & w^T p_i - b & \leq & -1, & p_i \in Q. \end{array} \tag{5.1}$$

Defining class labels y_i with $y_i = 1$ if $p_i \in P$ and $y_i = -1$ if $p_i \in Q$, the constraints of (5.1) become

$$y_i(w^T p_i - b) - 1 \geq 0, \quad i = 1, \ldots, n.$$

We could solve the optimization problem directly but for reasons that become apparent later we want to move to a dual formulation.

5.2 Lagrangian and dual problem

Lagrangian. Consider a (primal) optimization problem of the form

$$\begin{array}{lll} \min_x & f(x) & \\ \text{s.t.} & c_i(x) \leq 0, & i = 1, \ldots, n, \end{array} \tag{5.2}$$

where $f, c_i : \mathbb{R}^d \to \mathbb{R}$. The *Lagrangian* of (5.2) is the function $L : \mathbb{R}^d \times \mathbb{R}^n_+ \to \mathbb{R}$, defined by

$$L(x, \alpha_i) = f(x) + \sum_{i=1}^{n} \alpha_i c_i(x), \quad x \in \mathbb{R}^d, \alpha \in \mathbb{R}^n_+.$$

The Dual problem. Let us assume that there exist $\hat{x} \in \mathbb{R}^d, \hat{\alpha} \geq 0$ such that

$$L(\hat{x}, \alpha) \leq L(\hat{x}, \hat{\alpha}) \leq L(x, \hat{\alpha}), \quad \forall x \in \mathbb{R}^d, \forall \alpha \geq 0, \tag{5.3}$$

meaning that $(\hat{x}, \hat{\alpha})$ is a *saddle point* of the Lagrangian. From Exercise 20, we get that in this situation, $\hat{x}$ is an optimal solution to (5.2), with

$$f(\hat{x}) = L(\hat{x}, \hat{\alpha}).$$

We further get

$$\max_{\alpha \geq 0} \min_{x \in \mathbb{R}^d} L(x, \alpha) \leq \max_{\alpha \geq 0} L(\hat{x}, \alpha) \overset{(5.3)}{=} L(\hat{x}, \hat{\alpha}) \overset{(5.3)}{=} \min_{x \in \mathbb{R}^d} L(x, \hat{\alpha}) \leq \max_{\alpha \geq 0} \min_{x \in \mathbb{R}^d} L(x, \alpha),$$

where the first and last inequality have nothing to do with (5.3). This means, the optimum value of the primal problem (5.2) coincides with the optium value of the *dual problem*

$$\begin{array}{ll} \max_\alpha & \min_x L(x, \alpha) \\ \text{s.t.} & \alpha \geq 0 \end{array} \tag{5.4}$$

This dual problem has the nice feature that the constraints $c_i(x) \leq 0$ have 'disappeared'. In return, the objective function looks more complicated now than in the primal, because it has a nested minimum.
If for any fixed α, the Lagrangian is a *convex* differentiable function in x, with continuous partial derivatives, Fact 3.1.1 implies that

$$L(x^*, \alpha) = \min_x L(x, \alpha) \quad \Leftrightarrow \quad \partial_x L(x^*, \alpha) = 0,$$

where ∂_x is the vector of partial derivatives with respect to the variables $x_1, \ldots, x_d$. In this case, we can get rid of the nested minimum in the dual problem, by simply stipulating the additional constraint $\partial_x L(x, \alpha) = 0$. This leads to the following equivalent formulation of (5.4).

$$\begin{array}{ll} \max_\alpha & L(x,\alpha) \\ \text{s.t.} & \alpha \geq 0 \\ & \partial_x L(x,\alpha) = 0. \end{array} \tag{5.5}$$

In the maximum margin hyperplane problem, the Lagrangian is the convex quadratic function (in $x = (w, b)$)

$$L(w,b,\alpha) = \frac{1}{2}w^T w - \sum_i \alpha_i (y_i(w^T p_i - b) - 1),$$

so we can apply the previous machinery. The condition $\partial_x L(x,\alpha) = 0$ reads as

$$\begin{aligned} \frac{\partial L}{\partial w} = 0 \quad &\Leftrightarrow \quad w = \sum_i \alpha_i y_i p_i \\ \frac{\partial L}{\partial b} = 0 \quad &\Leftrightarrow \quad \sum_i \alpha_i y_i = 0. \end{aligned}$$

We can use the first equation to eliminate w and b from the objective function of the dual problem, i.e., from the Lagrangian L, and get the dual problem

$$\begin{array}{ll} \max_\alpha & -\frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j p_i^T p_j + \sum_i \alpha_i \\ \text{s.t.} & \alpha \geq 0 \\ & \sum_{i=1}^n \alpha_i y_i = 0. \end{array} \tag{5.6}$$

The constraint $w = \sum_i \alpha_i y_i p_i$ can be omitted from the problem; later, we are going to use it, of course, to derive w from an optimal solution $\hat{\alpha}$ to (5.6).
The important fact is that in our case, there is a saddle point of the Lagrangian according to (5.3), so that the maximum value of the dual problem (5.6) indeed coincides with the minimum value of the primal problem (5.1): there is no *duality gap*. This is implied by the following very general result.

Theorem 5.2.1 (Karush-Kuhn-Tucker) *Given a (primal) optimization problem (5.2) with convex objective function f and convex constraints c_i. Under some mild additional conditions (no conditions are needed if the c_i are linear), $\hat{x}$ is an optimal solution to (5.2) if and only if there exists $\hat{\alpha} \geq 0$ such that $(\hat{x}, \hat{\alpha})$ is a saddle point of the Lagrangian.*

If P and Q can be linearly separated, there is a feasible and therefore also an optimal solution $\hat{w}, \hat{b}$ to the primal problem (5.1).[1]
By the Karush-Kuhn-Tucker conditions, there is a saddle point of the Lagrangian which in turn implies that the dual problem (5.6) has an optimal solution whose value

[1]This follows from a standard compactness argument: if there is a feasible solution w_0, then we only need to look for an optimal solution within the ball $\{w \mid w^T w \leq w_0^T w_0\}$. This ball is compact, and so is its intersection with the closed halfspaces induced by the constraints. Within this compact intersection, the continuos objective function $w^T w/2$ assumes a minimum.

coincides with the optimum value of the primal. We can use the latter solution to compute the normal $\hat{w}$ of a maximum margin hyperplane as $\sum_i \hat{\alpha}_i y_i p_i$, where $\hat{\alpha}_i$ is an optimal solution of the dual problem. To compute the offset of a maximum margin hyperplane we can use the *complementarity conditions* implied by the saddle point, see Exercise 20 (i): given that $\alpha_i > 0$ for some index i, we find that $y_i(\hat{w}^T p_i - \hat{b}) - 1 = 0$. From this we get

$$\hat{b} = \hat{w}^T p_i - y_i = \sum_j \hat{\alpha}_j y_j p_j^T p_i - y_i.$$

The data points p_i for which $\alpha_i > 0$ are called *support vectors*.
Let us now have a closer look at the solution $\hat{w}$ of the dual problem.

Lemma 5.2.2 *Let $\hat{w}$ be a normal of a maximum margin hyperplane that separates point sets P and Q and let $\|p-q\|$ with $p \in \mathrm{conv}(P)$ and $q \in \mathrm{conv}(Q)$ the minimal distance between $\mathrm{conv}(P)$ and $\mathrm{conv}(Q)$. Then $p - q = \hat{w} / \sum_{\{i \,|\, p_i \in P\}} \hat{\alpha}_i$, where the $\hat{\alpha}_i$ are optimal for (5.6).*

Proof. Consider the dual of the maximum margin hyperplane problem. The second constraint can also be written as

$$\sum_{\{i \,|\, p_i \in P\}} \alpha_i = \sum_{\{i \,|\, p_i \in Q\}} \alpha_i.$$

Let $c := \sum_{\{i \,|\, p_i \in P\}} \hat{\alpha}_i$. We must have $c \neq 0$ (why?), so if we re-scale the vector $\hat{w} = \sum_i \hat{\alpha}_i y_i p_i$ by the factor $1/c$ we get $\hat{w}/c = p - q$, where

$$p = \sum_{\{i \,|\, p_i \in P\}} \bar{\alpha}_i p_i \in \mathrm{conv}(P) \quad \text{and} \quad q = \sum_{\{i \,|\, p_i \in Q\}} \bar{\alpha}_i p_i \in \mathrm{conv}(Q),$$

and $\bar{\alpha}_i := \hat{\alpha}_i / c$. We have

$$\|p - q\|^2 = \sum_{i,j} \bar{\alpha}_i \bar{\alpha}_j y_i y_j p_i^T p_j$$

and claim that this is the squared distance of the polytopes $\mathrm{conv}(P)$ and $\mathrm{conv}(Q)$. To see this assume the contrary, i.e., $\|p - q\|$ is larger than the distance between $\mathrm{conv}(P)$ and $\mathrm{conv}(Q)$. Recall the quadratic programming formulation of the polytope distance problem, see Exercise 14:

$$\begin{array}{rl} \min\limits_{\beta} & \sum_{i,j} \beta_i \beta_j y_i y_j p_i^T p_j \\ \text{s.t.} & \beta \geq 0 \\ & \sum_{\{i \,|\, p_i \in P\}} \beta_i = 1 \\ & \sum_{\{i \,|\, p_i \in Q\}} \beta_i = 1. \end{array}$$

Note that a solution $\hat{\beta}$ for this problem is feasible for the dual of the maximum margin hyperplane problem. The same holds for $\tilde{\beta} := c\hat{\beta}$. By our assumption, we have

$$\|p - q\|^2 = \sum_{i,j} \bar{\alpha}_i \bar{\alpha}_j y_i y_j p_i^T p_j > \sum_{i,j} \hat{\beta}_i \hat{\beta}_j y_i y_j p_i^T p_j.$$

This implies

$$\begin{aligned}
-\frac{1}{2}\sum_{i,j}\hat{\alpha}_i\hat{\alpha}_j y_i y_j p_i^T p_j + \sum_i \hat{\alpha}_i &= -\frac{c^2}{2}\sum_{i,j}\bar{\alpha}_i\bar{\alpha}_j y_i y_j p_i^T p_j + 2c \\
&< -\frac{c^2}{2}\sum_{i,j}\hat{\beta}_i\hat{\beta}_j y_i y_j p_i^T p_j + 2c \\
&= -\frac{1}{2}\sum_{i,j}\tilde{\beta}_i\tilde{\beta}_j y_i y_j p_i^T p_j + \sum_i \tilde{\beta}_i,
\end{aligned}$$

which is not possible since $-\frac{1}{2}\sum_{i,j}\hat{\alpha}_i\hat{\alpha}_j y_i y_j p_i^T p_j + \sum_i \hat{\alpha}_i$ is the optimum value of (5.6). Thus, $\|p - q\|$ is the optimal value for the objective function of the polytope distance problem and the statement of the lemma follows through uniqueness of $p - q$ (Exercise 14). $\square$

That is, the maximum margin problem is essentially the polytope distance problem, see also Figure 5.2.

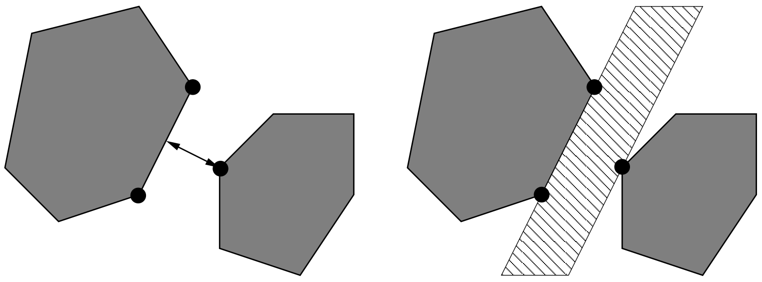

Figure 5.2: The maximum margin problem and the polytope problem are related. The highlighted vertices are the support vectors for both problems.

5.3 Relaxed Maximum Margin Hyperplane

The assumption of linearly separable data sets is not realistic for most applications. Here we want to deal with the case that though the data are not linearly separable a linear separation still makes sense, because it classifies most of the data correctly. Figure 5.3 depicts an example where the data are not linearly separable, but a linear separation still makes sense.

For linearly inseparable data sets (this includes the case where the convex hulls of the two data sets just touch), any pair (w, b) will violate at least one of the constraints

$$y_i(w^T p_i - b) - 1 \geq 0$$

of the primal problem (5.1). The plan is now to relax these constraints by adding positive slack variables z_i. The i-th relaxed constraints now reads as

$$y_i(w^T p_i - b) + z_i - 1 \geq 0, \quad z_i \geq 0.$$

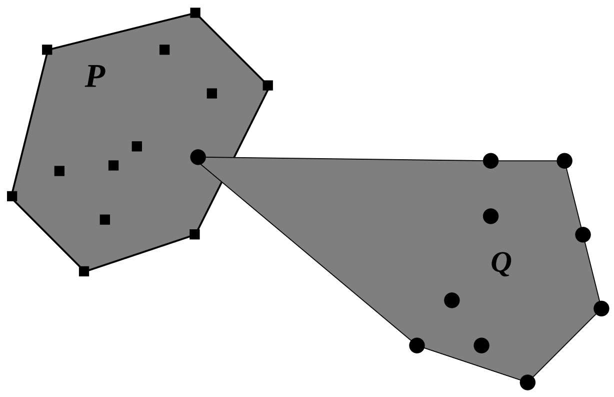

Figure 5.3: Inseparable data set for which a linear separation still is meaningful.

Relaxing the constraints means allowing outliers. We penalize outliers by adding another term to the objective function of the maximum margin hyperplane problem that contains the slack variables. The relaxed maximum margin hyperplane problem becomes

$$\begin{array}{lrcll} \min_{w,b} & \frac{1}{2}w^T w + C\sum_i z_i & & & \\ \text{s.t.} & y_i(w^T p_i - b) + z_i - 1 & \geq & 0, & i = 1,\ldots,n \\ & z_i & \geq & 0, & i = 1,\ldots,n. \end{array} \tag{5.7}$$

Here $C \geq 0$ is a parameter that controls the trade-off between maximizing the margin and penalizing the outliers. The problem still is a convex quadratic optimization problem that we can dualize as we did with the non-relaxed problem. Exercise 22 asks you to prove that the dual problem to (5.7) is the problem

$$\begin{array}{ll} \max_\alpha & -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j p_i^T p_j + \sum_i \alpha_i \\ \text{s.t.} & 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,n \\ & \sum_i \alpha_i y_i = 0. \end{array} \tag{5.8}$$

That is, the only difference to the non-relaxed dual problem (5.6) is that the coefficients α_i are also upper-bounded by the trade-off parameter C. The geometric interpretation of this situation is that instead of separating the convex hulls of the data, *reduced convex hulls* get separated, see Figure 5.4 for an example. To see this, we can argue as before that the vector $\hat{w} = \sum_i \hat{\alpha}_i y_i p_i$ resulting from an optimal solution to (5.8) satisfies

$$\hat{w}/c = p - q,$$

where $c = \sum_{\{i|x_i\in P\}} \hat{\alpha}_i$ and $p - q$ is the shortest vector with $p \in \text{conv}_{C/c}(P), q \in \text{conv}_{C/c}(Q)$,

$$\text{conv}_t(X) := \{\sum_{x\in X}\lambda_x x \mid \sum_{x\in X}\lambda_x = 1, 0 \leq \lambda_x \leq t \;\; \forall x \in X\}.$$

Note that $\text{conv}_t(X)$ becomes empty for $t < 1/|X|$.

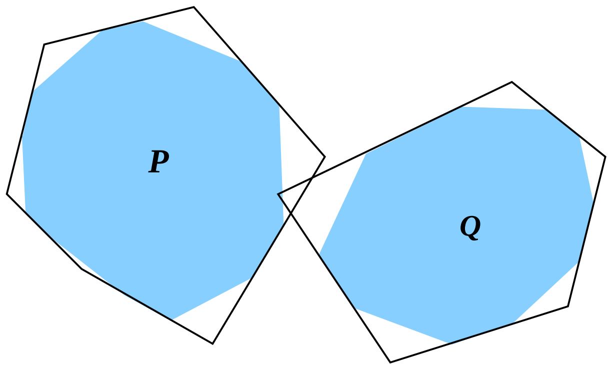

Figure 5.4: Reduced convex hulls.

5.4 Kernel trick

In many cases a linear classifier simply does not do the job even if we allow outliers. For example the data in Figure 5.5 can be separated meaningfully only with a non-linear discriminant function.

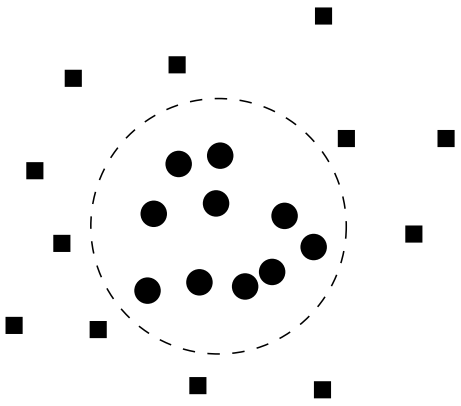

Figure 5.5: Data set is "best" separated by a non-linear classifier.

The key idea behind support vector machines is to map the data points non-linearly into some higher dimensional space, where they (hopefully) can be separated linearly. Let $\varphi : \mathbb{R}^d \to \mathbb{R}^{d'}$ be such a mapping. The dual of the relaxed maximum margin problem in $\mathbb{R}^{d'}$ looks as follows

$$\begin{array}{rc} \max\limits_{\alpha} & -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \varphi(p_i)^T\varphi(p_j) + \sum_i \alpha_i \\ \text{s.t.} & 0 \le \alpha_i \le C, \quad i = 1,\ldots,n \\ & \sum_i \alpha_i y_i = 0. \end{array}$$

That is, the constraints remain exactly the same, only the objective function changes.

The classifier that we get from a solution $\hat{\alpha}_i$ of this problem is

$$f(x) := \operatorname{sign}\left(\underbrace{\sum_i \hat{\alpha}_i y_i \varphi(p_i)^T}_{\hat{w}} \varphi(x) - \hat{b}\right).$$

Note that (depending on φ) this is now a non-linear classifier on $\mathbb{E}^d$ though it is linear on $\mathbb{E}^{d'}$. In Figure 5.6 it is schematically shown how this non-linear classification works.

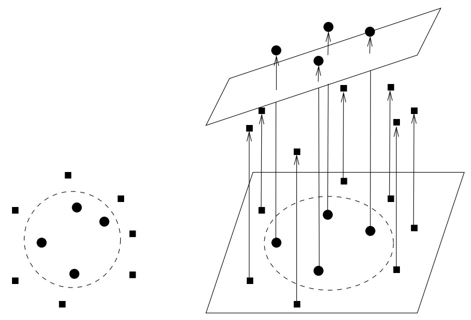

Figure 5.6: Linear separation of lifted data and the resulting non-linear classifier in input space.

In practice it is often infeasible to map the data explicitly to some higher dimensional space. Instead it is done implicitly using a *kernel* function. A kernel function is a positive semi-definite function $k : \mathbb{E}^d \times \mathbb{E}^d \to \mathbb{R}$, and we will use $k(x,y)$ to replace the scalar product $\varphi(x)^T\varphi(y)$ in d'-dimensional space.
It can be shown that for certain kernel functions, this actually works, meaning that there is a mapping φ from $\mathbb{R}^d$ to some higher (possibly infinite) dimensional Hilbert space such that

$$k(x,y) = \varphi(x)^T\varphi(y), \quad x, y \in \mathbb{R}^d.$$

This is Mercer's theorem, also known as the *kernel trick*, see [8]) for more information. Popular kernel function are

(1) Polynomials of degree d': $k(x,y) = (x^Ty + c)^{d'}$,

(2) Gaussian functions: $k(x,y) = e^{-c\|x-y\|^2}$,

(3) Sigmoid functions: $k(x,y) := \tanh(x^Ty + c)$.

The resulting classifier when using the kernel trick is

$$\operatorname{sign}\left(\sum_i \hat{\alpha}_i y_i k(p_i, x) - b\right).$$

Note that the kernels in the sum of the classifier only have to be evaluated for support vectors, i.e., for data points p_i with $\hat{\alpha}_i > 0$.

5.5 Exercises

Exercise 19 *The Karush-Kuhn-Tucker conditions are a generalization of the Lagrange multiplier theorem from equality to inequality constraints. Given a differentiable function $f : \mathbb{E}^d \to \mathbb{R}$ and differentiable constraints $c_i : \mathbb{E}^d \to \mathbb{R}$ then a solution $\bar{x}$ of the problem*

$$\begin{aligned} \max_x \quad & f(x) \\ \textit{s.t.} \quad & c_i(x) = 0, \end{aligned}$$

fulfills

$$\nabla f(\bar{x}) = \sum_i \alpha_i \nabla c_i(\bar{x}),$$

for some $\alpha_i \in \mathbb{R}$. Use this to determine the axis parallel box with maximal volume and prescribed surface area a.

Exercise 20 *For a constrained optimization problem*

$$\begin{aligned} \min_x \quad & f(x) \\ \textit{s.t.} \quad & c_i(x) \leq 0, \end{aligned}$$

with functions $f : \mathbb{E}^d \to \mathbb{R}$ and $c_i : \mathbb{E}^d \to \mathbb{R}$ let

$$L(x, \alpha) = f(x) + \sum_{i=1}^{n} \alpha_i c_i(x)$$

be its Lagrangian. Assume that $\hat{x} \in \mathbb{E}^d$ and $\hat{\alpha} \geq 0$ exist such that for all $x \in \mathbb{E}^d$ and $\alpha \geq 0$

$$L(\hat{x}, \alpha) \leq L(\hat{x}, \hat{\alpha}) \leq L(x, \hat{\alpha}).$$

Prove that the following two facts are implied.

(i) $\hat{\alpha}_i c_i(\hat{x}) = 0$ *for all i, and*

(ii) $\hat{x}$ *is an optimal solution to the optimization problem.*

Exercise 21 *What are the benefits of introducing the dual of the maximum margin hyperplane problem?*

Exercise 22 *Prove that problem (5.8) is the dual problem of the primal relaxed maximum margin problem (5.7).*

Bibliography

[1] M. Bădoiu and K. L. Clarkson. Optimal core-sets for balls. submitted, 2002.

[2] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin Heidelberg, 1987.

[3] K. Fischer and B. Gärtner. The smallest enclosing ball of balls: combinatorial structure and algorithms. In *Proc. 19th annual ACM Symposium on Computational Geometry (SCG)*, pages 292–301, 2003.

[4] A. Goel, P. Indyk, and K. R. Varadarajan. Reductions among high-dimensional proximity problems. In *Proc. 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 769–778, 2001.

[5] J. Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin Heidelberg, 2002.

[6] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, 1995.

[7] A. L. Peressini, F. E. Sullivan, and J. J. Uhl. *The Mathematics of Nonlinear Programming*. Undergraduate Texts in Mathematics. Springer-Verlag, 1988.

[8] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge Massachusetts, 2002.

[9] R. Seidel. Personal communication, 1997.

[10] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes in Computer Science*, pages 359–370. Springer-Verlag, 1991.