Project Optimization 1: Location Analysis

Tasks & Solutions

February 2026

Submission

You are strongly encouraged to work on this project in groups of up to three. Each member of the group must submit an (identical) copy of the project before the deadline **23:59 Sunday 22.3.2026** on Inspera.

All members are expected to contribute on, and should be ready to answers questions to, all parts of the project. For full marks, all numbered points should be addressed.

The project should be completed as a written report, including a typeset pdf file. Background, calculations, justifications and conclusions should be clearly stated. All group members names must be displayed clearly on the pdf file.

The coding component must be runnable. Please submit your code as a Jupyter notebook with minimal external dependencies. In Inspera, the pdf is to be delivered separately, and the code as a zip file. The clarity of your report counts towards the final grade.

Please direct questions about the project towards Anja Ringstad, who can be reached via email at anjarin@stud.ntnu.no.

1 Introduction

Many important problems from industry, engineering-technical or social area, logistics and the energy industry lead to location problems. Examples of this are the planning of new industrial plants, production facilities or warehouses, of hospitals, rescue stations and other public facilities, to landscaping or the placement of sensors on technical components. For example, in order to avoid hospital spatial inefficiency, it is important to learn how to optimize the spaces such that relevant services such as surgical equipment or emergency services are located at the best points of access to other stations. As another example, for a decision-maker deciding the best location to build a warehouse relative to factories, it is important to find the best place to locate this warehouse in order to minimize costs of travel and transportations. Thus, location problems appear in many variants and with different constraints depending on the practical application.

Other examples of location analysis appear for instance in the following areas: [cite: 20]

- Urban and regional planning (e.g., locations for emergency facilities)
- Technology (e.g., placement of sensors on technical components)
- Economy (e.g., planning new production facilities)
- Geography (e.g., landscape design)
- Environment-oriented project management (e.g., development of mining landscapes)
- Engineering.

This project is intended to provide the mathematical tools to model and solve location problems. The main goal in location analysis is to find one or more new locations with minimal distance to a set of known locations.

2 Problem description

Let us consider m given points in the plane

$$a^i = (a_1^i, a_2^i)^\top \in \mathbb{R}^2, \quad i = 1, ..., m.$$

$A = \{a^1, ..., a^m\}$ is the set of all given locations. Let $\mathcal{M} := \{1, ..., m\}$. We denote the new location, which we want to determine, by

$$x = (x_1, x_2)^\top \in \mathbb{R}^2.$$

When modeling a location problem, distance measures as well as the selected objective function play a key role. We consider the following distances:

$$\begin{aligned} d_1(x, y) &:= |x_1 - y_1| + |x_2 - y_2| && \text{(Manhattan or Taxicab distance)} \\ d_2(x, y) &:= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} && \text{(Euclidean distance)} \\ d_\infty(x, y) &:= \max\{|x_1 - y_1|, |x_2 - y_2|\} && \text{(Maximum distance)} \end{aligned}$$

The individual distances between the unknown x and a given a^i are set differently according to the concrete application. The selection o the distance measure determines whether the distance should be measured by the linear distance (Eucludean) or by a blocknorm (Maximum or Manhattan). The Maximum distance is suitable if movement in both directions is possible and only the larger one of these two determines the length of the movement.

We want to minimize a given objective function $f : \mathbb{R}^2 \to \mathbb{R}$ and thus consider the problem

$$\min_{x \in \mathbb{R}^2} f(x),$$

where we consider the following objective functions:

$$f_d^C(x) := \max_{a \in A} d(a, x), \qquad f_d^M(x) := \sum_{a \in A} d(a, x), \qquad f_{d_2^2}^M(x) := \sum_{a \in A} d_2^2(a, x).$$

This gives the following minimization / location problems

$$\min_{x \in \mathbb{R}^2} f_d^C(x) = \min_{x \in \mathbb{R}^2} \max_{a \in A} d(a, x) \qquad \text{(Center problem)} \tag{1}$$

$$\min_{x \in \mathbb{R}^2} f_d^M(x) = \min_{x \in \mathbb{R}^2} \sum_{a \in A} d(a, x) \qquad \text{(Median problem, also called Fermat-Weber problem)} \tag{2}$$

2.1 Metrics and Norms

Mathematically, the distance between two points in the plane can be measured using an appropriate metric.

Definition 1. *Let Y be a non-empty set of $\mathbb{R}^2$. A function $d : Y \times Y \to \mathbb{R}$ is called metric on Y if d fulfils the following conditions for all $x, y, z \in Y$:*

- $d(x, y) = 0 \iff x = y$ *(definiteness)*
- $d(x, y) = d(y, x)$ *(symmetry)*
- $d(x, y) \leq d(x, z) + d(z, y)$ *(triangle inequality)*

Note that $d(x, y)$ represents the distance between the points x and y. As one would expect, distances are always non-negative. This is often an extra assumption, but not required:

$$0 = d(x, x) \leq d(x, y) + d(y, x) = 2d(x, y)$$

by the triangle inequality and symmetry, proving $d(x, y) \geq 0$ for all points x, y.

Definition 2. *A function $\|\cdot\| : \mathbb{R}^2 \to \mathbb{R}$ is called norm on $\mathbb{R}^2$ if $\|\cdot\|$ fulfills the following conditions for all $x, y \in \mathbb{R}^2$ and for all $\alpha \in \mathbb{R}$:*

- $\|x\| = 0 \iff x = 0$ *(definiteness)*
- $\|\alpha x\| = |\alpha| \cdot \|x\|$ *(positive homogeneity)*
- $\|x + y\| \leq \|x\| + \|y\|$ *(triangle inequality)*

Again, norms are always non-negative, which follows from the triangle inequality and the positive homogeneity:

$$0 = \|x - x\| \leq \|x\| + \|-x\| = 2\|x\|$$

If $\|\cdot\|$ is a norm, then it is possible to define a metric on $\mathbb{R}^2$ that is induced by the norm by defining for all $x, y \in \mathbb{R}^2$

$$d(x, y) \coloneqq \|x - y\|.$$

2.2 Project Tasks - Theoretical Work

1. Show that d_1, d_2 and d_∞ are metrics.

Solution:

For each of the metrics we have to show the properties from Definition 1. We will continuously use the fact that the absolute value $|\cdot| : \mathbb{R} \to \mathbb{R}$ is a norm on $\mathbb{R}$ (same properties as in Definition 2) and the following result:

Proposition 3. *A sum of non-negative values is zero if and only if all values are zero:*

$$\sum_{i=1}^{n} |\lambda_i| = 0 \iff \forall i \in \{1, \ldots, n\} : \ \lambda_i = 0.$$

In the following, let $x, y, z \in \mathbb{R}^2$.

- **Manhattan (d_1)**
  - $d_1(x, y) = 0 \iff |x_1 - y_1| + |x_2 - y_2| = 0 \iff (x_1 - y_1 = 0 \text{ **and** } x_2 - y_2 = 0) \iff x = y$ *(1 P.)*
  - $d_1(x, y) = |x_1 - y_1| + |x_2 - y_2| = |y_1 - x_1| + |y_2 - x_2| = d_1(y, x)$ *(1 P.)*
  - $d_1(x, y) = |x_1 - y_1| + |x_2 - y_2| \leq |x_1 - z_1| + |z_1 - y_1| + |x_2 - z_2| + |z_2 - y_2| = d_1(x, z) + d_1(z, y)$, because the absolute value satisfies the triangle inequality. *(1 P.)*
- **Euclidean (d_2)**
  - $d_2(x, y) = 0 \iff d_2^2(x, y) = 0 \iff (x_1 - y_1)^2 + (x_2 - y_2)^2 = 0 \iff (x_1 - y_1)^2 = 0 = (x_2 - y_2)^2 \iff x_1 - y_1 = 0 = x_2 - y_2 \iff x = y$ *(1 P.)*
  - $d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2} = d(y, x)$ *(1 P.)*
  - **Variant A: Using Cauchy-Schwarz Inequality:** Consider the squared norm $d_2^2(x, y) = \|x - y\|_2^2 = \langle x - y, x - y \rangle$:

$$\begin{aligned}
\|x - y\|_2^2 &= \|x - z + z - y\|_2^2 \\
&= \langle (x - z) + (z - y), (x - z) + (z - y) \rangle \\
&= \langle x - z, x - z \rangle - 2\langle x - z, z - y \rangle + \langle z - yz-, y \rangle \\
&= \|x - z\|_2^2 - 2\langle x - z, z - y \rangle + \|z - y\|_2^2 \\
&\leq \|x - z\|_2^2 + 2|\langle x - z, z - y \rangle| + \|z - y\|_2^2 \\
&\leq \|x - z\|_2^2 + 2\|x - z\|_2 \|z - y\|_2 + \|z - y\|_2^2 \qquad \text{(Cauchy-Schwarz)} \\
&= (\|x - z\|_2 + \|z - y\|_2)^2
\end{aligned}$$

Taking the square root on both sides yields

$$d_2(x,y) = \|x-y\|_2 \le \|x-z\|_2 + \|z-y\|_2 = d_2(x,z) + d_2(z,y).$$

Variant B: Using Convexity (Jensen's Inequality / Power Mean) We use the fact that $f(t) = t^2$ is a convex function.
Let $\|x-z\|_2 = U$ and $\|z-y\|_2 = V$. If $U = 0$, we get $x = z$ and thus $d_2(x,y) \le d_2(x,z) + d_2(z,y) = 0 + d_2(x,y)$, and similarly for $V = 0$ we get $d_2(x,y) = d_2(x,z)$. So from now on we assume $U, V > 0$. Let $\lambda = \frac{U}{U+V}$. Then $1-\lambda = \frac{V}{U+V}$.

$$\begin{aligned}
\left\| \frac{x-z+z-y}{U+V} \right\|_2^2 &= \sum_{i=1}^{2} \left(\frac{x_i - z_i + z_i - y_i}{U+V} \right)^2 \\
&= \sum_{i=1}^{2} \left(\frac{U}{U+V} \frac{x_i - z_i}{U} + \frac{V}{U+V} \frac{z_i - y_i}{V} \right)^2 \\
&\le \sum_{i=1}^{2} \left[\frac{U}{U+V} \left(\frac{x_i - z_i}{U} \right)^2 + \frac{V}{U+V} \left(\frac{z_i - y_i}{V} \right)^2 \right] \quad \text{(convexity of } x^2\text{)} \\
&= \frac{U}{U+V} \frac{1}{U^2} \sum_{i=1}^{2} (x_i - z_i)^2 + \frac{V}{U+V} \frac{1}{V^2} \sum_{i=1}^{2} (z_i - y_i)^2 \\
&= \frac{U}{U+V} \frac{\|x-z\|_2^2}{U^2} + \frac{V}{U+V} \frac{\|z-y\|_2^2}{V^2} \\
&= \frac{U}{U+V}(1) + \frac{V}{U+V}(1) = 1
\end{aligned}$$

Thus, $\|x-y\|_2^2 \le (U+V)^2$, which implies

$$d_2(x,y) = \|x-y\|_2 \le \|x-z\|_2 + \|z-y\|_2 = d_2(x,z) + d_2(z,y).$$

Variant C: Using an orthogonal decomposition Let $u = x - z$ and $v = z - y$. If $v = 0$, then $z = y$ and the triangle inequality is trivial. So assume $v \neq 0$. We decompose u into a part parallel to v and a part orthogonal to v:

$$u = u_{\parallel} + u_{\perp}, \qquad u_{\parallel} \in \operatorname{span}(v), \quad u_{\perp} \perp v.$$

Since $u_{\perp}$ is orthogonal to both $u_{\parallel}$ and v, it is also orthogonal to $u_{\parallel} + v$. Hence, by the Pythagorean theorem,

$$\begin{aligned}
\|u+v\|_2^2 &= \|u_{\perp} + (u_{\parallel} + v)\|_2^2 \\
&= \|u_{\perp}\|_2^2 + \|u_{\parallel} + v\|_2^2.
\end{aligned}$$

Now $u_{\parallel}$ and v are parallel, so there exists $\lambda \in \mathbb{R}$ with $u_{\parallel} = \lambda v$. Therefore,

$$\|u_{\parallel} + v\|_2 = \|(\lambda+1)v\|_2 = |\lambda+1|\,\|v\|_2 \le (|\lambda|+1)\|v\|_2 = \|u_{\parallel}\|_2 + \|v\|_2.$$

Using this in the previous identity yields

$$\|u+v\|_2^2 \le \|u_{\perp}\|_2^2 + (\|u_{\parallel}\|_2 + \|v\|_2)^2.$$

Moreover, since $u_{\parallel} \perp u_{\perp}$, we have again by the Pythagorean theorem

$$\|u\|_2^2 = \|u_{\parallel}\|_2^2 + \|u_{\perp}\|_2^2.$$

Hence

$$\begin{aligned}\|u+v\|_2^2 &\le \|u_\perp\|_2^2 + \|u_\parallel\|_2^2 + 2\|u_\parallel\|_2\|v\|_2 + \|v\|_2^2 \\ &= \|u\|_2^2 + 2\|u_\parallel\|_2\|v\|_2 + \|v\|_2^2 \\ &\le \|u\|_2^2 + 2\|u\|_2\|v\|_2 + \|v\|_2^2 \\ &= (\|u\|_2 + \|v\|_2)^2.\end{aligned}$$

Taking square roots gives

$$\|u+v\|_2 \le \|u\|_2 + \|v\|_2.$$

Substituting back $u = x - z$ and $v = z - y$, we obtain

$$d_2(x,y) = \|x-y\|_2 = \|(x-z)+(z-y)\|_2 \le \|x-z\|_2 + \|z-y\|_2 = d_2(x,z) + d_2(z,y).$$

Thus, d_2 satisfies the triangle inequality. *(3 P.)*

- **Maximum (d_∞)**
  - $d_\infty(x,y) = 0 \iff \max\{|x_1-y_1|, |x_2-y_2|\} = 0 \iff |x_1-y_1|, |x_2-y_2| \le 0 \iff |x_1-y_1|, |x_2-y_2| = 0 \iff x = y$ *(1 P.)*
  - $d_\infty(x,y) = \max\{|x_1-y_1|, |x_2-y_2|\} = \max\{|y_1-x_1|, |y_2-x_2|\} = d(y,x)$ *(1 P.)*
  - **Variant A: Standard Computation (Element-wise)** By definition, $\|u-v\|_\infty = \max_i |u_i - v_i|$. For any specific index k:

$$\begin{aligned}|u_k - v_k| &\le |u_k| + |v_k| \qquad \text{(Triangle inequality in } \mathbb{R}) \\ &\le \max_i |u_i| + \max_j |v_j| \\ &= \|u\|_\infty + \|v\|_\infty\end{aligned}$$

    Since this holds for every k, it must also hold for the index that achieves the maximum:

$$d_\infty(x,y) = \max_i |x_i - y_i| = \max_i |x_i - z_i + z_i - y_i| \le \|x_i - z_i\|_\infty + \|z_i - y_i\|_\infty = d_\infty(x,z) + d_\infty(z,y).$$

    Variant B: Using an identity for the maximum We use the identity for $u, v \in \mathbb{R}$

$$\max(u,v) = \frac{u+v+|u-v|}{2}.$$

    This follows easily by the fact that $\frac{u+v}{2}$ is the middle of u and v and $\frac{|u-v|}{2}$ is half the distance between these, so their sum is the maximum. Then, the result follows from the triangle inequality for the absolute value and the reverse triangle inequality

$$||u| - |v|| \le |u - v|$$

by the following computation:

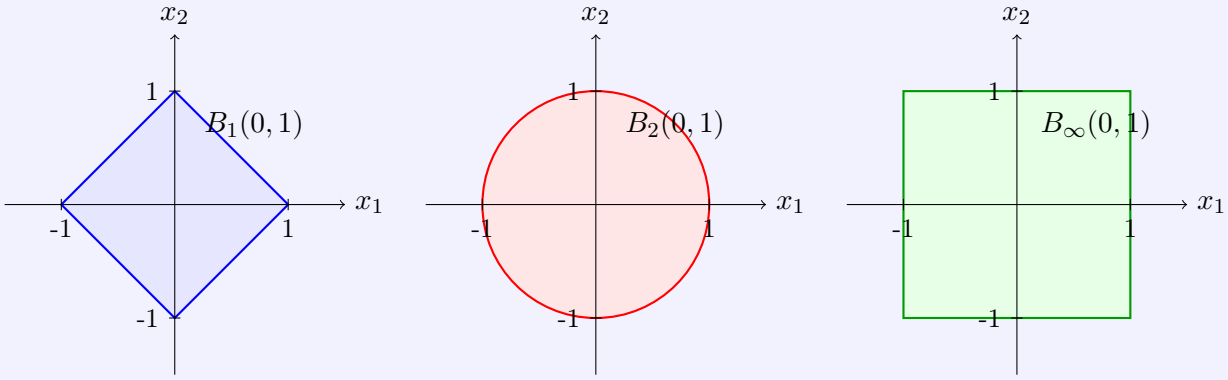
$$
\begin{aligned}
d_\infty(x,y) &= \max\{|x_1-y_1|, |x_2-y_2|\} \\
&= \frac{|x_1-y_1| + |x_2-y_2| + \big||x_1-y_1| - |x_2-y_2|\big|}{2} \\
&\le \frac{|x_1-z_1| + |z_1-y_1| + |x_2-z_2| + |z_2-y_2|}{2} \\
&\quad + \frac{\big||x_1-z_1| + |z_1-y_1| - |x_2-z_2| - |z_2-y_2|\big|}{2} \\
&\le \frac{|x_1-z_1| + |z_1-y_1| + |x_2-z_2| + |z_2-y_2|}{2} \\
&\quad + \frac{\big||x_1-z_1| - |x_2-z_2|\big| + \big||z_1-y_1| - |z_2-y_2|\big|}{2} \\
&= \frac{|x_1-z_1| + |x_2-z_2| + \big||x_1-z_1| - |x_2-z_2|\big|}{2} \\
&\quad + \frac{|z_1-y_1| + |z_2-y_2| + \big||z_1-y_1| - |z_2-y_2|\big|}{2} \\
&= d_\infty(x,z) + d_\infty(z,y).
\end{aligned}
$$

(2 P.)

Total: 12 Points

2. Plot the unit balls $B_i(0,1) \coloneqq \{x \in \mathbb{R}^2 \mid \|x\|_i \le 1\}$ for $i \in \{1, 2, \infty\}$.

Solution:



unit ball of 1-Norm | unit ball of 2-Norm | unit ball of ∞-Norm)

(3 P.)

Remark: You can always see whether a norm is strictly convex based on their unit ball: It is strictly convex if and only if the unit ball B is strictly convex, meaning for all $x, y \in \partial B$, so $\|x\|, \|y\| = 1$, with $x \neq y$ and all $\lambda \in (0,1)$ it is $\lambda x + (1-\lambda)y \in \text{int}(B)$, so $\|\lambda x + (1-\lambda)y\| < 1$. We see that $\|\cdot\|_2$ is strictly convex, while $\|\cdot\|_1$ (choose $(1,0)^T$ and $(0,1^T)$) and $\|\cdot\|_\infty$ (choose $(1,1)^T$ and $(-1,1)^T$) are not.

Total: 3 Points

3. Show that every norm is a convex function.

Solution:

For a norm $\|\cdot\|$ and all $x, y \in \mathbb{R}^n$ and all $\lambda \in [0,1]$, by the triangle inequality and positive homogeneity it is

$$\|\lambda x + (1-\lambda)y\| \leq \|\lambda x\| + \|(1-\lambda)y\| = \lambda\|x\| + (1-\lambda)\|y\|,$$

because $\lambda, 1-\lambda \geq 0$. This matches the definition of a convex function. *(2 P.)*

Total: 2 Points

4. Show that the objective function of problems (1) and (2) is convex.

Solution:

Variant A: Piece by Piece:
We will first show that for every fixed location $a \in A$ the function $d(a,\cdot) : \mathbb{R}^2 \to \mathbb{R}$ is convex and then prove the following result, which will give us the desired convexity of the objective functions.

Proposition 4. *Let $f_i : \mathbb{R}^2 \to \mathbb{R}$ for $i \in \{1,\dots,n\}$ be convex functions. Then, also*

$$f^C(x) \coloneqq \max_{i=1,\dots,n} f_i(x) \quad \text{and} \quad f^M(x) \coloneqq \sum_{i=1}^{n} f_i(x)$$

are convex.

- **$d_i(a,\cdot) : \mathbb{R}^2 \to \mathbb{R}$ is convex for $i \in \{1,2,\infty\}$:**

  Let $x, y \in \mathbb{R}^2$ and $\lambda \in [0,1]$. From task 3 we know that norms are convex, so we only have to see that the norm of the difference of a point and a convex combination of two points is the same as the weighted sum of the norm of the difference of the point and each of the two other points (You could also argue that translations of convex functions are convex, because the definition does not depend on any anchor point):

  $$\begin{aligned} d_i(a, \lambda x + (1-\lambda)y) &= \|a - \lambda x - (1-\lambda)y\|_i \\ &= \|\lambda a + (1-\lambda)a - \lambda x - (1-\lambda)y\|_i \\ &= \|\lambda(a-x) + (1-\lambda)(a-y)\|_i \\ &\overset{\text{task 3}}{\leq} \lambda\|a-x\|_i + (1-\lambda)\|a-y\|_i. \end{aligned}$$

  (1 P.)

  Remark 5. *We used the fact that our d_i are actually induced by norms and thus we can use the positive homogeneity of the norms. Also, this means that the same result holds true for any by a norm induced metric. For any general given metric d it is in general not true that $d(a,\cdot)$ is convex. A good example is $d : \mathbb{R} \to \mathbb{R}$, $d(x,y) \coloneqq \frac{|x-y|}{1+|x-y|}$. You can check that this really satisfies the axioms of a metric, but for $a = 0$, the function $d(a,\cdot)$ is $x \mapsto \frac{|x|}{1+|x|}$, which is concave!*

  Remark 6. *It is also correct to say that for a metric that is induced by a norm, $x \mapsto \|x-a\| = d(x,a)$ is convex as a translation of the norm, which by task 3 is convex. The proof of the statement that translated convex functions remain convex is the same as we showed though:*

  $$g(\lambda x + (1-\lambda)y - a) = g(\lambda(x-a) + (1-\lambda)(y-a)) \leq \lambda g(x-a) + (1-\lambda)g(y-a).$$

Another way without that proof would be to say that a function is convex if and only if its epigraph is a convex set and since the epigraph of a translated function is just the translation of the epigraph it remains convex and thus the translated function remains convex.

- **proof of proposition:**

  Let $x, y \in \mathbb{R}^2$ and $\lambda \in [0,1]$.
  For f^C since every f_j is convex we get

$$\begin{aligned} f^C(\lambda x + (1-\lambda)y) &= \max_{i=1,\dots,n} f_i(\lambda x + (1-\lambda)y) \\ &\leq \max_{i=1,\dots n} \lambda f_i(x) + (1-\lambda) f_i(y) \\ &\leq \lambda \max_{i=1,\dots,n} f_i(x) + (1-\lambda) \max_{i=1,\dots,n} f_i(y) \\ &= \lambda f^C(x) + (1-\lambda) f^C(y). \end{aligned}$$

  Thus, the maximum of convex functions is convex. *(2 P.)*

  Now, for the sum of convex functions we use that each of them is convex to get

$$\begin{aligned} f^M(\lambda x + (1-\lambda)y) &= \sum_{i=1}^{n} f_i(\lambda x + (1-\lambda)y) \\ &\leq \sum_{i=1}^{n} \lambda f_i(x) + (1-\lambda) f_i(y) \\ &= \lambda \sum_{i=1}^{n} f_i(x) + \sum_{i=1}^{n} f_i(y) \\ &= \lambda f^M(x) + (1-\lambda) f^M(y). \end{aligned}$$

  Thus, also the sum of convex functions is convex. □ *(2 P.)*

- **end result:**
  Since $d_i(a, \cdot)$ are convex for all $i \in \{1, 2, \infty\}$ and every $a \in \mathbb{R}^2$ the proposition gives us that the maximum of finitely many of these is also convex, thus the centre objective functions $f_{d_i}^C$ are convex, and that the sum of finitely many of these is convex as well, thus the median objective functions $f_{d_i}^M$ are convex. *(1 P.)*

Variant B: Direct proof for the objective functions:
We now prove the convexity of the centre and median objective functions directly from their definitions. Let $\|\cdot\|$ be a norm, let $x, y \in \mathbb{R}^2$ and let $\lambda \in [0,1]$.
For the centre objective function

$$f_d^C(x) = \max_{a \in A} \|a - x\|$$

we get

$$\begin{aligned} f_d^C(\lambda x + (1-\lambda)y) &= \max_{a \in A} \|a - (\lambda x + (1-\lambda)y)\| \\ &= \max_{a \in A} \|\lambda(a-x) + (1-\lambda)(a-y)\| \\ &\leq \max_{a \in A} \big(\lambda \|a - x\| + (1-\lambda)\|a - y\|\big) \\ &\leq \lambda \max_{a \in A} \|a - x\| + (1-\lambda) \max_{a \in A} \|a - y\| \\ &= \lambda f_d^C(x) + (1-\lambda) f_d^C(y), \end{aligned}$$

where we used the triangle inequality and the positive homogeneity of the norm. Thus f_d^C is convex.
For the median objective function

$$f_d^M(x) = \sum_{a \in A} \|a - x\|$$

we obtain similarly

$$\begin{aligned} f_d^M(\lambda x + (1-\lambda)y) &= \sum_{a \in A} \|a - (\lambda x + (1-\lambda)y)\| \\ &= \sum_{a \in A} \|\lambda(a-x) + (1-\lambda)(a-y)\| \\ &\leq \sum_{a \in A} \big(\lambda \|a-x\| + (1-\lambda)\|a-y\|\big) \\ &= \lambda \sum_{a \in A} \|a-x\| + (1-\lambda) \sum_{a \in A} \|a-y\| \\ &= \lambda f_d^M(x) + (1-\lambda) f_d^M(y). \end{aligned}$$

Thus f_d^M is convex as well.

Remark 7. *In the special case $m = 1$, both objective functions reduce to $x \mapsto \|a^1 - x\| = d(a, x)$, which is therefore convex for the same reason. We need this result for task 7.*

Total: 6 Points

5. Give a geometric interpretation of solving problem (1) with Euclidean distance function.

Solution:

Variant A: Minimum enclosing ball. Since $d(a, x)$ can be interpreted as the distance from x to a and the Euclidean distance d_2 is induced by the Euclidean norm, whose balls are circles, the problem can be interpreted as follows:
We want to find the *minimum enclosing ball*. In other words, we seek a centre x such that the radius needed to cover the furthest point in A is minimized.

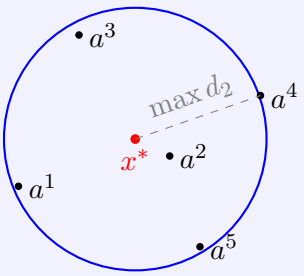


Figure 1: Variant A: smallest enclosing ball around x^*

Variant B: First intersection point. Equivalently, since $d(a, x) = d(x, a)$, we may also fix the radius at each given location a^i and enlarge around each a^i circles simultaneously until they

have a common intersection, that is, until

$$\emptyset \neq \bigcap_{i=1}^{m} B_2(a^i; r).$$

Any intersection point can then serve as a feasible centre, and the smallest such radius is the optimal value.

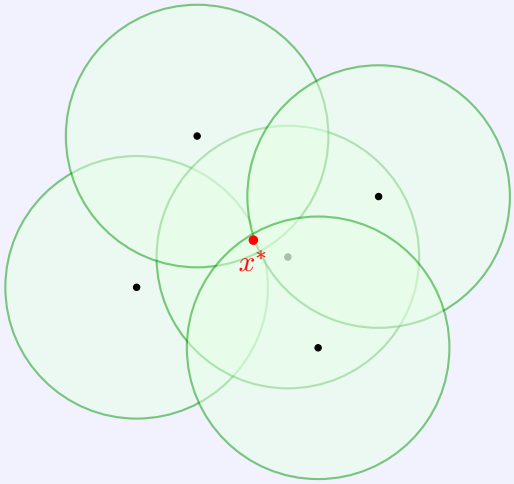


Figure 2: Variant B: equal balls around the data points intersect in x^*

(2 P.)

Remark 8. *The interpretation for other norms remains the same, we just have to take the respective balls of the norms (instead of taking circles).*

Total: 2 Points

6. Provide a solution approach for problem (2) with Manhattan distance (Hint: Note that the objective function can be separated (so-called separability). Just describe the idea of such an approach.).

Solution:

Setting in the definition of the Manhattan norm/distance in the median problem (2) we get *(1 P.)*

$$\min_{x\in\mathbb{R}^2} \sum_{i=1}^{m} d_1(a^i, x) = \min_{x\in\mathbb{R}^2} \sum_{i=1}^{m} |a_1^i - x_1| + |a_2^i - x_2| = \min_{x\in\mathbb{R}^2} \sum_{i=1}^{m} |a_1^i - x_1| + \min_{x\in\mathbb{R}^2} \sum_{i=1}^{m} |a_2^i - x_2|.$$

We can do this, because $|a_1^i - x_1|$ only depends on the first component x_1 while the other part only depends on the second component x_2. In order to find the desired $x \in \mathbb{R}^2$ we can see that the minimisations on the rights side each have two variables x_1, x_2, but only require one, so we can equivalently write

$$\min_{x\in\mathbb{R}^2} \sum_{i=1}^{m} |a_1^i - x_1| + \min_{x\in\mathbb{R}^2} \sum_{i=1}^{m} |a_2^i - x_2| = \min_{x_1\in\mathbb{R}} \sum_{i=1}^{m} |a_1^i - x_1| + \min_{x_2\in\mathbb{R}} \sum_{i=1}^{m} |a_2^i - x_2|$$

and construct the complete solution out of these parts *(1 P.)*

$$x = \left(\operatorname*{argmin}_{x_1\in\mathbb{R}} \sum_{i=1}^{m} |a_1^i - x_1| \,,\, \operatorname*{argmin}_{x_2\in\mathbb{R}} \sum_{i=1}^{m} |a_2^i - x_2| \right).$$

For solving any of the sub-problems, let us say $\min_{x_1 \in \mathbb{R}} \sum_{i=1}^{m} |a_1^i - x_1|$, we order the first components of the given locations a_1^i from small to big and get $\hat{a}_1^i$, so $\hat{a}_1^1 \leq \hat{a}_1^2 \leq \cdots \leq \hat{a}_1^m$. Then the solution is given by *(1 P.)*

$$x_1 \coloneqq \begin{cases} \hat{a}^{(n+1)/2_1} & \text{if } n \text{ is odd,} \\ \in [\hat{a}_1^{n/2}, \hat{a}_1^{n/2+1}] & \text{if } n \text{ is even.} \end{cases}$$

It is up to interpretation whether we want all solutions or only one of them. The same is done for the second component. In orer to see that this truly is the solution we can take a look at $\sum_{i=1}^{m} |a_1^i - x_1| = \sum_{i=1}^{m} |\hat{a}_1^i - x_1|$. As we proved this is a convex function. Besides in points $x_1 = a_1^i$ it is also differentiable and for $x_1 < \hat{a}_1^1$ all the absolute values when drawn goes down or in other words $|\hat{a}_1^i - x_1| = \hat{a}_1^i - x_1|$, so the derivative is $-n$. For $x_1 \in (\hat{a}_1^1, \hat{a}_1^2)$ the drawn absolute value of $|\hat{a}_1^1 - x_1|$ goes up or $|\hat{a}_1^1 - x_1 - x_1| = x_1 - \hat{a}_1^1$, while all others have the opposite sign, and thus the derivative is $1 - (n-1) = n - 2$. In general, in between $\hat{a}_1^i$ and $\hat{a}_1^{i+1}$ we have i absolute values going up and $n - i$ of them going down, so the derivative is $i - (n - i) = 2i - n$. If n is now even, we can see that for $i = \frac{n}{2}$ we get a derivative of $2\frac{n}{2} - n = 0$ and thus found our minimum as any point in between these two. The two points $\hat{a}_1^{n/2}$ and $\hat{a}_1^{n/2+1}$ are also solutions, because the objective function is continuous and thus has the same value there. If n is odd, we see that that in between $\hat{a}_1^{(n+1)2-1}$ and $\hat{a}_1^{(n+1)/2}$ the derivative is $2(\frac{n+1}{2} - 1) - n = -1$ and between $\hat{a}_1^{(n+1)2}$ and $\hat{a}_1^{(n+1)/2+1}$ the derivative is $2\frac{n+1}{2} - n = 1$. Thus, the unique solution is $\hat{a}_1^{(n+1)2}$.
The solution approach can thus be summarised as:

(a) Order the first components of the given locations: $a_1^1, \cdots, a_1^m$ into $\hat{a}_1^1 \leq \hat{a}_1^2 \leq \cdots \leq \hat{a}_1^m$.

(b) Choose (any or all)

$$x_1 \coloneqq \begin{cases} \hat{a}_1^{(n+1)/2} & \text{if } n \text{ is odd,} \\ \in [\hat{a}_1^{n/2}, \hat{a}_1^{n/2+1}] & \text{if } n \text{ is even.} \end{cases}$$

(c) Order the second components of the given locations: $a_2^1, \cdots, a_2^m$ into $\hat{a}_2^1 \leq \hat{a}_2^2 \leq \cdots \leq \hat{a}_2^m$.

(d) Choose (any or all)

$$x_2 \coloneqq \begin{cases} \hat{a}_2^{(n+1)/2} & \text{if } n \text{ is odd,} \\ \in [\hat{a}_1^{n/2}, \hat{a}_1^{n/2+1}] & \text{if } n \text{ is even.} \end{cases}$$

(e) The solution is $x = (x_1, x_2)$ (or the set of all combinations).

Total: 3 Points

7. Consider the Median problem with squared Euclidean distance:

$$\min_{x \in \mathbb{R}^2} \sum_{a \in A} (d_2(a, x))^2.$$

Show that the objective function is convex, and analytically compute the uniquely determined minimiser of this problem.

Solution:

The objective function $f_{d_2^2}^M(x) = \sum_{a \in A} (d_2(a, x))^2$ is strictly convex. Therefore, there exists at most one minimiser. We write $d_2^2(a, x) = (d_2(a, x))^2$.

- **Variant A:** From task 4 we already know that $d_2(a,\cdot)$ is convex and that the sum of convex functions is convex. So, we just have to prove that also the squared distance $d_2^2(a,\cdot)$ is convex. We can prove a slightly more powerful result:

  Proposition 9. *Let $g : \mathbb{R}^2 \to \mathbb{R}$ be convex and $h : \mathbb{R} \to \mathbb{R}$ be convex and monotone non-decreasing on the image of g, i.e. on $g[\mathbb{R}^2] \coloneqq \{g(x) \mid x \in \mathbb{R}\}$. Then the composition $h \circ g : \mathbb{R}^2 \to \mathbb{R}$, $(h \circ g)(x) = h(g(x))$, is convex.*

  Proof. For any $x, y \in \mathbb{R}^2$ and $\lambda \in (0,1)$ it is by convexity of g already $g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda)g(y)$ and since h is monotone non-decreasing on the image of g we get

  $$\begin{aligned}(h \circ g)(\lambda x + (1-\lambda)y) &= h(g(\lambda x + (1-\lambda)y)) \\ &\leq h(\lambda g(x) + (1-\lambda)g(y)) \\ &\leq \lambda h(g(x)) + (1-\lambda)h(g(y)) = \lambda(h \circ g)(x) + (1-\lambda)(h \circ g)(y).\end{aligned}$$

  Thus, $h \circ g$ is convex. □

  Since the image of $d(a,\cdot)$ are the non-negative real numbers $\mathbb{R}_{\geq}$ and $h(x) = x^2$ is convex and non-decreasing there, we now get that $d_2^2(a,\cdot)$ is a convex function and thus $f_{d_2^2}^M$ is convex.

  Remark: $d_2(a,\cdot)$ is even strictly convex. The proposition can be extended by the following result:
  If g is strictly convex and h is monotone increasing and strictly convex, then $h \circ g$ is strictly convex.
  The proof of all these parts remains almost the same. As result we get that even $f_{d_2^2}^M$ is strictly convex.

- **Variant B:** It is

  $$f_{d_2^2}^M(x) = \sum_{a \in A} d_2^2(a,x) = \sum_{i=1}^{m} \big((a_1^i - x_1)^2 + (a_2^i - x_2)^2\big).$$

  We compute the gradient

  $$\nabla f_{d_2^2}^M(x) = \left(\sum_{i=1}^{m} 2(x_1 - a_1^i)\,,\, \sum_{i=1}^{m} 2(x_2 - a_2^i)\right) = \left(2mx_1 - 2\sum_{i=1}^{m} a_1^i\,,\, 2mx_2 - 2\sum_{i=1}^{m} a_2^i\right)$$

  and the Hessian matrix

  $$H_{f_{d_2^2}^M}(x) = \operatorname{diag}(2m, 2m) = \begin{pmatrix} 2m & 0 \\ 0 & 2m \end{pmatrix}.$$

  $H_{f_{d_2^2}^M}(x)$ is positive definite (as diagonal matrix with positive entries), thus the function $f_{d_2^2}^M$ is strictly convex. *(3 P.)*

We use the necessary optimality condition $\nabla f_{d_2^2}^M(x) = (0,0)^T$ and from the computation in Variant B get the condition

$$\forall j \in \{1,2\} : \; x_j = \frac{1}{m}\sum_{i=1}^{m} a_j^i \quad \Longrightarrow \quad x = \frac{1}{m}\sum_{a \in A} a$$

The solution is the **centroid** of the given points. Since $f_{d_2^2}^M$ is convex, this condition is also sufficient. *(2 P.)*

Remark 10. *Notice that this problem is also separable*

$$\min_{x\in\mathbb{R}^2}\sum_{i=1}^m d_2^2(x,a)=\min_{x\in\mathbb{R}^2}\left(\sum_{i=1}^m(x_1-a_1^i)^2+\sum_{i=1}^m(x_2-a_2^i)^2\right)=\min_{x_1\in\mathbb{R}^2}\sum_{i=1}^m(x_1-a_1^i)^2+\min_{x_2\in\mathbb{R}^2}\sum_{i=1}^m(x_2-a_2^i)^2$$

Remark 11. *We can also derive the minimiser directly by expanding the square. Since*

$$\begin{aligned}f_{d_2^2}^M(x)&=\sum_{i=1}^m\|x-a^i\|^2\\&=\sum_{i=1}^m\left(\|x\|^2-2\langle x,a^i\rangle+\|a^i\|^2\right)\\&=m\|x\|^2-2\left\langle x,\sum_{i=1}^m a^i\right\rangle+\sum_{i=1}^m\|a^i\|^2,\end{aligned}$$

completing the square yields

$$f_{d_2^2}^M(x)=m\left\|x-\frac{1}{m}\sum_{i=1}^m a^i\right\|^2+\sum_{i=1}^m\|a^i\|^2-\frac{1}{m}\left\|\sum_{i=1}^m a^i\right\|^2.$$

The second and third terms are constant with respect to x, so $f_{d_2^2}^M$ is minimised exactly when

$$x=\frac{1}{m}\sum_{i=1}^m a^i.$$

This also shows uniqueness of the minimiser, because the first term vanishes only at this point.

Total: 5 Points

8. Consider the Median problem with Euclidean distance function

$$\min_{x\in\mathbb{R}^2} f_{d_2}^M(x)=\min_{x\in\mathbb{R}^2}\sum_{a\in A} d_2(a,x).$$

Derive necessary optimality conditions for minimisers of this problem (Hint: Note that the partial derivatives are not defined globally.).

Solution:

Since the partial derivatives are not defined everywhere (namely at $x=a^i$), we have to consider different cases:

(a) **Differentiable points:** Here $\nabla f_{d_2}^M(x)=0$ must hold for the minimiser x. We calculate

$$\nabla f_{d_2}^M(x)=\nabla\sum_{i=1}^m\sqrt{(a_1^i-x_1)^2+(a_2^i-x_2)^2}=\begin{pmatrix}\sum_{i=1}^m\frac{2(x_1-a_1^i)}{2\sqrt{(a_1^i-x_1)^2+(a_2^i-x_2)^2}}\\\sum_{i=1}^m\frac{2(x_2-a_2^i)}{2\sqrt{(a_1^i-x_1)^2+(a_2^i-x_2)^2}}\end{pmatrix}=\begin{pmatrix}\sum_{i=1}^m\frac{(x_1-a_1^i)}{\|a^i-x\|_2}\\\sum_{i=1}^m\frac{(x_2-a_2^i)}{\|a^i-x\|_2}\end{pmatrix}$$

leading to *(1 P.)*

$$\sum_{a\in A}\frac{x-a}{d_2(a,x)}=(0,0)^T.$$

Since $f_{d_2}^M$ is convex, this condition is even sufficient. *(1 P.)*

(b) **Non-differentiable points ($x = a^k$):**

- **Variant A: directional derivatives**
  The directional derivative of a function f at x in the direction d (we can take $\|d\|_2 = 1$) is defined as:
  $$f'(x;d) \coloneqq \lim_{t \downarrow 0} \frac{f(x+td) - f(x)}{t}$$

  Proposition 12. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and let $x^* \in \mathbb{R}^n$. Then x^* is a global minimizer of f if and only if*
  $$f'(x^*;d) \geq 0 \qquad \text{for all directions } d \in \mathbb{R}^n.$$

  Proof. If x^* is a global minimizer, then for every d and every $t > 0$ we have
  $$f(x^* + td) - f(x^*) \geq 0.$$
  Dividing by t and passing to the limit $t \downarrow 0$ gives $f'(x^*;d) \geq 0$.
  Conversely, assume $f'(x^*;d) \geq 0$ for all d. Let $x \in \mathbb{R}^n$ be arbitrary and set $d = x - x^*$. The function
  $$\varphi(t) \coloneqq f(x^* + td), \qquad t \in [0,1],$$
  is convex. Hence
  $$\varphi(1) \geq \varphi(0) + \varphi'_+(0).$$
  Since $\varphi'_+(0) = f'(x^*;d) \geq 0$, we obtain
  $$f(x) = \varphi(1) \geq \varphi(0) = f(x^*).$$
  Therefore x^* is a global minimizer. □

  A point x^* is therefore a local (and due to convexity, global) minimizer if and only if $f'(x^*;d) \geq 0$ for all (unit) directions $d \in \mathbb{R}^2$. *(1 P.)*
  At $x = a^k$, the term $f_k(x) = \|x - a^k\|_2$ is not differentiable, but its directional derivative exists: *(1 P.)*
  $$f_k'(a^k;d) = \lim_{t \downarrow 0} \frac{\|a^k + td - a^k\|_2 - 0}{t} = \lim_{t \downarrow 0} \frac{t\|d\|_2}{t} = \|d\|_2 = 1$$

  Now we calculate the directional derivative of the full objective function $f_{d_2}^M$ at a^k using the fact that for the differentiable part $f_{\neq k}(x) = \sum_{i \neq k} \|a^i - x\|_2$ it is $f'_{\neq k}(x;d) = \nabla f_{\neq k}(x) \cdot d$:
  $$f'(a^k;d) = f'_{\neq k}(a^k;d) + f'_k(a^k;d) = \left\langle \sum_{i \neq k} \frac{a^k - a^i}{\|a^k - a^i\|_2}\,,\, d \right\rangle + 1$$

  By Cuchy-Schwarz it is $\langle u, v \rangle \leq \|u\|_2 \|v\|_2$ and thus $\langle u, v \rangle \geq -\|u\|_2 \|v\|_2$ for any $u, v \in \mathbb{R}^2$ giving us
  $$f'(a^k;d) \geq 1 - \left\| \sum_{i \neq k} \frac{a^k - a^i}{\|a^k - a^i\|_2} \right\|_2 \|d\|_2 = 1 - \left\| \sum_{i \neq k} \frac{a^k - a^i}{\|a^k - a^i\|_2} \right\|_2$$

and this bound gets achieved by choosing the normalised direction *(2 P.)*

$$\tilde{d} = -\sum_{i \neq k} \frac{a^k - a^i}{\|a^k - a^i\|_2}.$$

The condition $f'(a^k; d) \geq 0$ for all d is therefore equivalent to: *(1 P.)*

$$\left\| \sum_{i \neq k} \frac{a^k - a^i}{\|a^k - a^i\|_2} \right\|_2 \leq 1$$

- **Variant B: subdifferential**
  At the anchor points, f is not differentiable. We use the concept of the *subdifferential*

$$\partial f(x_0) := \{\xi \in \mathbb{R}^2 \mid \forall x \in \mathbb{R}^2 : \ f(x) \geq f(x_0) + \langle \xi, x - x_0 \rangle\}$$

of f in a point x_0. Here are some useful results:

Proposition 13. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. Then*

$$\partial \|0\| = \{\xi \in \mathbb{R}^n \mid \|\xi\|_* \leq 1\},$$

i.e. the subdifferential at 0 is the closed unit ball of the dual norm. For the Euclidean norm this is exactly the unit ball

$$\partial \|0\|_2 = B_2(0, 1).$$

Proof. By definition, $\xi \in \partial\|0\|$ if and only if

$$\|x\| \geq \langle \xi, x \rangle \qquad \text{for all } x \in \mathbb{R}^n.$$

This is equivalent to

$$\sup_{\|x\| \leq 1} \langle \xi, x \rangle \leq 1,$$

which means exactly $\|\xi\|_* \leq 1$. In the Euclidean case, the dual norm is again the Euclidean norm, so the set is $B_2(0, 1)$. □

Proposition 14. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex. Then x^* is a global minimizer of f if and only if*

$$0 \in \partial f(x^*).$$

Proof. If $0 \in \partial f(x^*)$, then by the definition of the subdifferential,

$$f(x) \geq f(x^*) + \langle 0, x - x^* \rangle = f(x^*) \qquad \text{for all } x \in \mathbb{R}^n,$$

so x^* is a global minimizer.
Conversely, if x^* is a global minimizer, then $f(x) \geq f(x^*)$ for all x, hence

$$f(x) \geq f(x^*) + \langle 0, x - x^* \rangle \qquad \text{for all } x \in \mathbb{R}^n.$$

Thus $0 \in \partial f(x^*)$. □

Proposition 15. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and differentiable at x_0. Then*

$$\partial f(x_0) = \{\nabla f(x_0)\}.$$

Proof. Let $\xi \in \partial f(x_0)$. Then by definition

$$f(x) \geq f(x_0) + \langle \xi, x - x_0 \rangle \qquad \text{for all } x \in \mathbb{R}^n.$$

Since f is differentiable at x_0, we also have

$$f(x) = f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle + o(\|x - x_0\|).$$

Combining both relations gives

$$\langle \nabla f(x_0) - \xi, x - x_0 \rangle + o(\|x - x_0\|) \geq 0.$$

Now choose $x = x_0 + tv$ with $t > 0$ and $v \in \mathbb{R}^n$. Dividing by t and letting $t \downarrow 0$, we obtain

$$\langle \nabla f(x_0) - \xi, v \rangle \geq 0 \qquad \text{for all } v \in \mathbb{R}^n.$$

Replacing v by $-v$ yields the reverse inequality, hence

$$\langle \nabla f(x_0) - \xi, v \rangle = 0 \qquad \text{for all } v \in \mathbb{R}^n.$$

Therefore $\xi = \nabla f(x_0)$, so $\partial f(x_0) \subseteq \{\nabla f(x_0)\}$.
Conversely, for convex differentiable functions the first-order inequality holds:

$$f(x) \geq f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle \qquad \text{for all } x \in \mathbb{R}^n.$$

Thus $\nabla f(x_0) \in \partial f(x_0)$, and we conclude

$$\partial f(x_0) = \{\nabla f(x_0)\}. \qquad \square$$

We use the optimality condition $0 \in \partial f(x)$. The subdifferential of the Euclidean norm $f_i(x) = \|x - a^i\|_2$ at $x_0 = a^i$ is the unit ball $B_2(0,1) = \{\xi \in \mathbb{R}^2 : \|\xi\|_2 \leq 1\}$. For all other $i \neq k$, the function is differentiable at a^k. Thus, for $x = a^k$, the optimality condition becomes:

$$0 \in \sum_{i \neq k} \frac{a^k - a^i}{\|a^k - a^i\|_2} + B_2(0,1)$$

This can be rearranged to

$$\sum_{i \neq k} \frac{a^k - a^i}{\|a^k - a^i\|_2} \in -B_2(0,1) = B_2(0,1)$$

giving us the result

$$\left\| \sum_{i \in \mathcal{M} \setminus \{k\}} \frac{a^k - a^i}{\|a^k - a^i\|_2} \right\|_2 \leq 1.$$

Summary of Conditions The point x^* is the unique global minimizer if:

(a) Either $x^* = a^k$ and $\left\| \sum_{a \neq a^j} \frac{a^j - a}{\|a^j - a\|_2} \right\|_2 \leq 1$ for some $k \in \mathcal{M}$.

(b) Or $x^* \notin \{a^1, \ldots, a^m\}$ and $\sum_{i=1}^m \frac{x^* - a^i}{\|x^* - a^i\|_2} = 0$.

Remark 16. *From the condition for the case that $x = a^k$ we can easily see the result in Theorem 20. When adding some weights the calculations do not change much and we get the result that $x = a^k$ is a minimum if and only if*

$$Test_k = \left\| \sum_{i \neq k} v_i \frac{a^k - a^i}{\|a^k - a^i\|_2} \right\|_2 \leq v_k$$

In particular, by the triangle inequality,

$$Test_k \leq \sum_{i \neq k} v_i.$$

Hence a simple sufficient condition for optimality of $x = a^k$ is

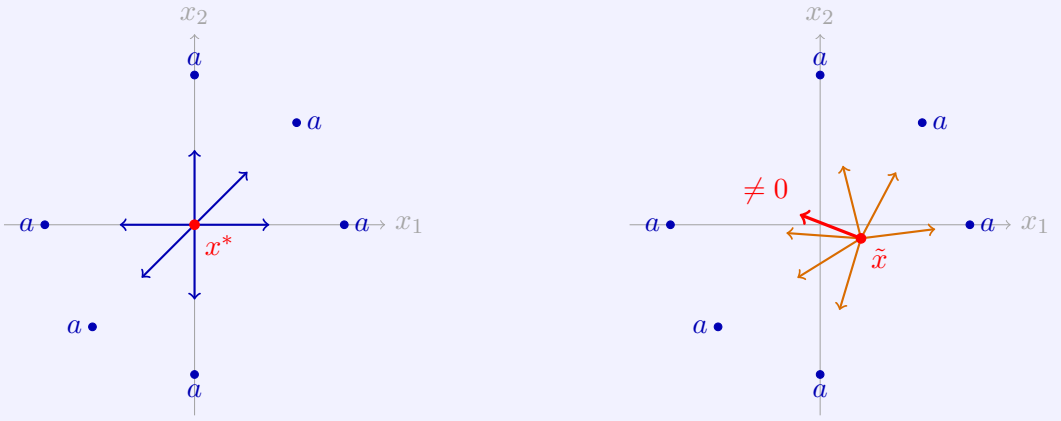
$$v_k \geq \sum_{i \neq k} v_i.$$

Therefore, if the weight of one given location is at least as large as the sum of all other weights, then this location is optimal for the weighted Euclidean median problem.

Remark 17 (Interpretation: Balance of directions)**.** *The condition*

$$\sum_{a \in A} \frac{x^* - a}{\|a - x^*\|_2} = 0 \qquad \Longleftrightarrow \qquad \sum_{a \in A} \frac{a - x^*}{\|a - x^*\|_2} = 0$$

has a direct geometric meaning: the normalised directions from x^* to the data points balance each other. In the picture on the left, these directions cancel out; in the picture on the right, they do not. Note that this interpretation only applies directly in the differentiable case $x^* \notin A$. If $x^* = a^k$ for one of the given locations, one term would involve division by $\|a^k - x^*\|_2 = 0$, so the formula is not defined. The correct replacement in that case is given by directional derivatives or, equivalently, by the subdifferential condition,resulting in the sum not having to be zero, but having a norm smaller or equal to 1.



Solution: directions cancel each other out Non-solution: the directions do not balance

Figure 3: Visualising the vanishing sum of normalised directions in the differentiable case

Remark 18 (Second-order optimality conditions)**.** *In a general nonlinear optimisation problem, one often studies second-order conditions in addition to first-order ones. Typically, if f is twice continuously differentiable, then a necessary condition for a local minimiser x^* is that*

$$\nabla f(x^*) = 0 \qquad \text{and} \qquad H_f(x^*) \text{ positive semi-definite},$$

while a standard sufficient condition is

$$\nabla f(x^*) = 0 \qquad \textit{and} \qquad H_f(x^*) \textit{ positive definite},$$

which yields a strict local minimiser. In our setting, however, these criteria are not really needed to identify global minimisers: the objective function of the Euclidean median problem is convex, so every first-order optimality condition that we derived is already sufficient for global optimality. Moreover, the function is not twice differentiable everywhere anyway, since it fails to be differentiable at the anchor points $a^i \in A$. Hence second-order conditions are of limited use here and would only apply on the differentiable region $\mathbb{R}^2 \setminus A$. There they can still be used as an additional consistency check, but conceptually the convexity-based first-order conditions are the more natural and more powerful tool in this problem.

Total: 7 Points

9. Give an example of a set of existing locations, where the set of global minimisers for the problems $\min_{x\in\mathbb{R}^2} f^M_{d_2}(x)$ and $\min_{x\in\mathbb{R}^2} f^M_{d_2^2}(x)$ do not coincide. Explain the solution.

Solution:

We can start from the optimality conditions from tasks 7 and 8 or we can think about the problems. Choosing the second approach we see that the biggest difference is that the squares get very big for big values of $d(a, x)$. Thus, the median problem with squared distances punishes big distances much more than when using non-squared distances, where the objective function is more robust towards big distances. Thus, the idea is to include one location far away from the others.
We choose $a^1 = (0, 0)^T$, $a^2 = (0, 2)^T$ and the far away point $a^3 = (9, 1)^T$. By symmetry, both minimisers must lie on the line $x_2 = 1$. Indeed, for fixed x_1, both objective functions are convex as functions of x_2 and the two points $a^1 = (0, 0)^T$ and $a^2 = (0, 2)^T$ are mirror images with respect to the line $x_2 = 1$. Hence the objective value is symmetric with respect to $x_2 = 1$, so the minimum in the x_2-direction is attained at $x_2 = 1$. Therefore it is enough to minimise both objective functions along the line $x_2 = 1$, which explains why both solution points have second component equal to 1. *(1 P.)*

- **squared distances:** When considering squared distances we in task 7 found the solution to be

$$x = \frac{1}{m}\sum_{i=1}^{m} a^i = \frac{1}{3}\left(\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0\\2\end{pmatrix} + \begin{pmatrix}10\\1\end{pmatrix}\right) = \begin{pmatrix}3\\1\end{pmatrix}.$$

- d_2 **distances:** The solution in this case is

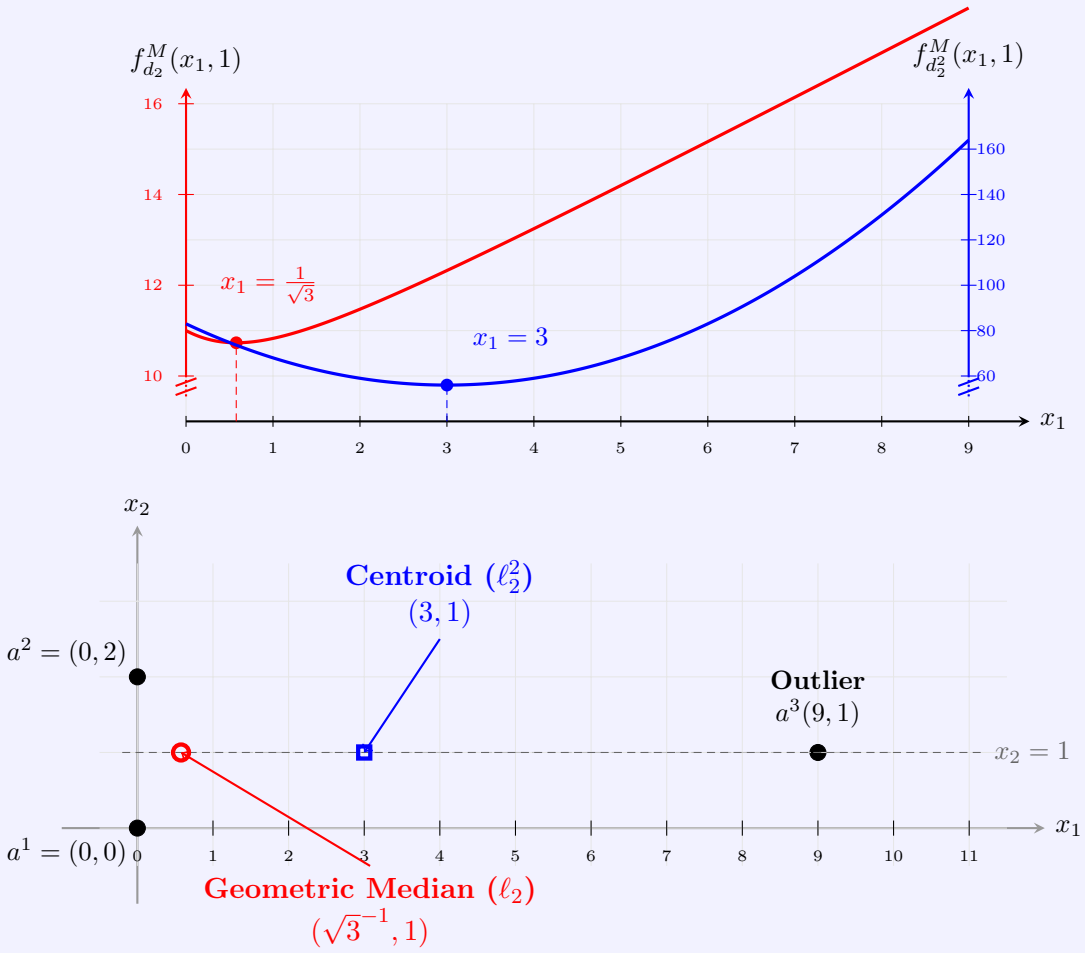
$$x = \left(\frac{1}{\sqrt{3}}, 1\right).$$

  Since this is not equal to any of the given locations, we can check with the condition from the last task:

$$\sum_{i=1}^{3} \frac{x - a^i}{\|a^i - x\|_2} = \frac{\left(\frac{1}{\sqrt{3}}, 1\right)^T}{2/\sqrt{3}} + \frac{\left(\frac{1}{\sqrt{3}}, -1\right)}{2/\sqrt{3}} + \frac{\left(\frac{1}{\sqrt{3}} - 9, 0\right)^T}{9 - \frac{1}{\sqrt{3}}} = (0, 0)^T.$$

  If you prefer checking the other condition: What would happen if we would add $a^4 = (\frac{1}{\sqrt{3}}, 1)^T$? (Maybe with a weight $v_4 = 0$. Does the problem change? Can we use an optimality condition now?) *(3 P.)*

The following picture shows both objective functions restricted to the symmetry line $x_2 = 1$. This makes it visible that the minimiser of the Euclidean-distance objective is at $x_1 = \frac{1}{\sqrt{3}}$, while the minimiser of the squared-distance objective is at $x_1 = 3$.



Remark 19 (Points on a line)**.** *In this example one can simplify the Euclidean-median problem further by placing all points on the x-axis. If the points are* $a^i = (a_1^i, 0)^\top$ *and* $x = (x_1, x_2)^\top$*, then*

$$f^M_{d_2}(x) = \sum_{i=1}^{m} \sqrt{(x_1 - a_1^i)^2 + x_2^2}.$$

For fixed x_1*, every summand is minimal for* $x_2 = 0$*, hence the whole sum is minimal for* $x_2 = 0$*. Therefore the problem reduces to*

$$\min_{x_1 \in \mathbb{R}} \sum_{i=1}^{m} |x_1 - a_1^i|,$$

which is exactly the one-dimensional median problem from task 5. Thus a minimiser is obtained by taking x_1 *to be a median of the numbers* $a_1^1, \ldots, a_1^m$ *and setting* $x_2 = 0$*. If the points are ordered and* m *is odd, this is just the middle point.*
More generally, assume that all given points lie on a line

$$L = \{p + td \mid t \in \mathbb{R}\}$$

with some fixed $p \in \mathbb{R}^2$ *and direction* $d \neq 0$*. Then we can write* $a^i = p + t_i d$*. Every point* $x \in \mathbb{R}^2$ *can be decomposed as*

$$x = p + td + z, \qquad z \perp d.$$

Hence

$$\|x - a^i\|_2^2 = \|(t - t_i)d + z\|_2^2 = |t - t_i|^2 \|d\|_2^2 + \|z\|_2^2,$$

because the two summands are orthogonal. Therefore

$$\|x - a^i\|_2 = \sqrt{|t - t_i|^2 \|d\|_2^2 + \|z\|_2^2} \geq |t - t_i| \|d\|_2,$$

with equality if and only if $z = 0$, i.e. if and only if $x \in L$. Summing over i shows that every minimiser must lie on L. Restricted to L, the objective becomes

$$f_{d_2}^M(p + td) = \sum_{i=1}^m \|(t - t_i)d\|_2 = \|d\|_2 \sum_{i=1}^m |t - t_i|.$$

Since the factor $\|d\|_2$ is constant, minimising this is again equivalent to the one-dimensional median problem from task 5 for the values t_i. Hence every median of the parameters t_i yields a minimiser. In particular, for an odd number of ordered points on a line, the middle point is the unique minimiser.

Total: 4 Points

3 Weiszfeld Algorithm

The above problem in task 7. was easy to solve, as the objective function is differentiable and one is able to derive the stationary points by rearranging the formula appearing in the optimality condition[cite: 89]. If we consider the Median problem with Euclidean distance function that is not squared, an according procedure is not as simple anymore, as one cannot solve for a minimizer directly. Instead, one uses an iterative procedure to compute approximations of a minimizer.

Here, we consider the Median problem with weighted Euclidean distances as follows:

$$\min_{x \in \mathbb{R}^2} \sum_{i \in \mathcal{M}} v_i d_2(a^i, x) \tag{3}$$

where the weights $v_i, i \in \mathcal{M} = \{1, \ldots, m\}$, are real-valued nonnegative (for $v_i < 0$ the problem is no longer convex and for $v_i = 0$ we can simply ignore location a^i, so we can also assume for the weights to be positive) numbers. The problem (3) is one of the most fundamental location problems[cite: 95]. It consists of finding a point that minimizes the sum of its weighted distances to a given finite set of anchor points.

The problem is credited to the well-known French mathematician Pierre de Fermat, who at the beginning of the seventeenth century posed the following question:

Given three points in a plane, find a fourth point such that the sum of its distances to the three given points is as small as possible.

The Italian physicist and mathematician Evangelista Torricelli found a construction method of this point by ruler and compass, and it is therefore also called "the Torricelli point". At the beginning of the twentieth century, the German economist Alfred Weber incorporated weights, and was able to treat facility location problems with more than 3 facilities; the problem was consequently called "the Fermat-Weber problem".

Before presenting an algorithm for solving problem (3), we give the following result:

Theorem 20 (see: Love, Morris Wesolowsky, 1988, Property 2.2)**.** *If for $k \in \mathcal{M}$ and*

$$Test_k := \left[\left(\sum_{i \in \mathcal{M} \setminus \{k\}} v_i \frac{a_1^k - a_1^i}{d_2(a^k, a^i)} \right)^2 + \left(\sum_{i \in \mathcal{M} \setminus \{k\}} v_i \frac{a_2^k - a_2^i}{d_2(a^k, a^i)} \right)^2 \right]^{\frac{1}{2}},$$

it holds that

$$Test_k \leq v_k,$$

for one $k \in \mathcal{M}$, then $x^* = a^k$ is a global minimizer.

The so-called Weiszfeld algorithm is an iterative method based on the first-order necessary conditions for a stationary point of the objective function.

Weiszfeld algorithm for solving (3)

Input. Existing locations $a^i = (a_1^i, a_2^i)^T \in \mathbb{R}^2$ (pairwise different), weights $v_i > 0, i \in \{1, \ldots, m\}$, error bound $\epsilon > 0$

(a) Check if the minimum is attained at an existing location a^i, $i \in \{1, \ldots, m\}$. If for

$$\text{Test}_k = \left[\left(\sum_{i \in \mathcal{M} \setminus \{k\}} v_i \frac{a_1^k - a_1^i}{d_2(a^k, a^i)} \right)^2 + \left(\sum_{i \in \mathcal{M} \setminus \{k\}} v_i \frac{a_2^k - a_2^i}{d_2(a^k, a^i)} \right)^2 \right]^{\frac{1}{2}}$$

it holds that $\text{Test}_k \leq v_k$ for one $k \in \mathcal{M} \Rightarrow$ Set $x^* = a^k$ and END. Otherwise go to Step (b).

(b) Choose a starting point $x = (x_1, x_2)$. This point can be found, for example, by solving the median problem using the squared Euclidean norm.

(c) Set for $j = 1, 2$:

$$x_j^{new} = \frac{\sum_{i=1}^m v_i \frac{a_j^i}{d_2(a^i, x)}}{\sum_{i=1}^m v_i \frac{1}{d_2(a^i, x)}}$$

(d) If $x^{new} = (x_1^{new}, x_2^{new})$ satisfies a stopping criterion w.r.t. the given ϵ (see §3.1 below) then set $x^* = x^{new}$. Otherwise: Set $x = x^{new}$ and go to Step (c).

Output. Approximation of the minimizer $x^* = (x_1^*, x_2^*)$.
The main iteration scheme in the algorithm is

$$x_1^{new} := \frac{\sum_{i=1}^m v_i \frac{a_1^i}{d_2(a^i, x)}}{\sum_{i=1}^m v_i \frac{1}{d_2(a^i, x)}}, \quad x_2^{new} := \frac{\sum_{i=1}^m v_i \frac{a_2^i}{d_2(a^i, x)}}{\sum_{i=1}^m v_i \frac{1}{d_2(a^i, x)}} \tag{4}$$

3.1 Termination Criterion

Let us define $f_{d_2}(x) := \sum_{i \in \mathcal{M}} v_i d_2(a^i, x)$. Let $x \in \mathbb{R}^2$ be an iteration point that was obtained during the Weiszfeld algorithm. We define

$$\sigma(x) := \max\{d_2(x, y) : y \in \text{conv}\{a^1, \ldots, a^m\}\}$$

with $\text{conv}\{a^1, \ldots, a^m\} := \{x \in \mathbb{R}^2 : \exists \lambda_1 \geq 0, \ldots, \lambda_m \geq 0, \sum_{i=1}^m \lambda_i = 1, x = \sum_{i=1}^m \lambda_i a^i\}$ being the convex hull.

Theorem 21. *Let $x \in \mathbb{R}^2$ be an iteration point that was obtained during the Weiszfeld algorithm. Then*

$$UB(x) := \|\nabla f_{d_2}(x)\|_2 \cdot \sigma(x)$$

is an upper bound for a possible improvement of the objective function $f_{d_2}(x)$, where $\|\cdot\|_2$ is the Euclidean norm in $\mathbb{R}^2$.

Proof. Because f_{d_2} is convex, we have for an iteration point x obtained during the Weiszfeld algorithm

$$\begin{aligned} f_{d_2}(x) - f_{d_2}(x^*) &\leq \langle \nabla f_{d_2}(x), x - x^* \rangle \\ &\leq \|\nabla f_{d_2}(x)\|_2 \cdot \|x - x^*\|_2 \qquad \text{(by Cauchy-Schwarz)} \\ &\leq \|\nabla f_{d_2}(x)\|_2 \cdot \sigma(x) \end{aligned}$$

□

Theorem 22. *Let x^* denote the global minimizer of (3). A lower bound for $f_{d_2}(x^*)$ is given by*

$$LB(x) := f_{d_2}(x) - \|\nabla f_{d_2}(x)\| \cdot \sigma(x).$$

Proof. We have from Theorem 21

$$f_{d_2}(x) - f_{d_2}(x^*) \leq \|\nabla f_{d_2}(x)\| \cdot \sigma(x),$$

and therefore

$$f_{d_2}(x) - \|\nabla f_{d_2}(x)\| \cdot \sigma(x) \leq f_{d_2}(x^*).$$

□

Based on these two results, show that the following consequence holds:

Theorem 23. *If for the lower bound, we have $LB(x) > 0$ and for a given $\epsilon > 0$*

$$\frac{UB(x)}{LB(x)} = \frac{\|\nabla f_{d_2}(x)\| \cdot \sigma(x)}{f_{d_2}(x) - \|\nabla f_{d_2}(x)\| \cdot \sigma(x)} < \epsilon$$

holds true, then the current iterate has at most a relative approximation error of ϵ. [1]

3.1 Project Tasks - Weiszfeld Algorithm

10. Explain why the main iteration scheme is chosen according to (4).

Solution:

Variant A: Fixed-point iteration based on optimality condition
The iteration scheme for the weighted Median problem is derived from the first-order necessary condition since we are not in the case $x = a^k$ (This is already checked with the test criterion. We assume we do not hit any a^k by accident during the iteration). The calculations are similar to the ones in task 8: We calculate *(1 P.)*

$$\nabla f_{d_2}(x) = \nabla \sum_{i=1}^{m} v_i \sqrt{(a_1^i - x_1)^2 + (a_2^i - x_2)^2} = \begin{pmatrix} \sum_{i=1}^{m} \frac{2v_i(x_1 - a_1^i)}{2\sqrt{(a_1^i - x_1)^2 + (a_2^i - x_2)^2}} \\ \sum_{i=1}^{m} \frac{2v_i(x_2 - a_2^i)}{2\sqrt{(a_1^i - x_1)^2 + (a_2^i - x_2)^2}} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} v_i \frac{(x_1 - a_1^i)}{\|a^i - x\|_2} \\ \sum_{i=1}^{m} v_i \frac{(x_2 - a_2^i)}{\|a^i - x\|_2} \end{pmatrix}$$

[1] The relative error of the current iterate x is defined as $\left| \frac{f_{d_2}(x) - f_{d_2}(x^*)}{f_{d_2}(x^*)} \right|$.

The condition $\nabla f_{d_2}(x) = 0$ gives us

$$(0,0)^T = \sum_{i=1}^m v_i \frac{(x-a^i)}{\|a^i - x\|_2} \quad \Longrightarrow \quad \sum_{i=1}^m v_i \frac{a^i}{\|a^i - x\|_2} = \sum_{i=1}^m v_i \frac{x}{\|a^i - x\|_2} = x \sum_{i=1}^m v_i \frac{1}{\|a^i - x\|_2}.$$

Dividing by the factor of x on the right side now gives the iteration scheme *(2 P.)*

$$x_j = \frac{\sum_{i=1}^m v_i \frac{a^i_j}{\|a^i - x\|_2}}{\sum_{i=1}^m v_i \frac{1}{\|a^i - x\|_2}} = \frac{\sum_{i=1}^m v_i \frac{a^i_j}{d_2(a^i,x)}}{\sum_{i=1}^m v_i \frac{1}{d_2(a^i,x)}} \quad \text{for } j = 1,2 \qquad \text{so} \quad x = \frac{\sum_{i=1}^m v_i \frac{a^i}{\|a^i - x\|_2}}{\sum_{i=1}^m v_i \frac{1}{\|a^i - x\|_2}}$$

We notice that x appears on both sides of the equation, so this can only be used as a fixed point iteration, which is exactly (4). Since f_{d_2} is convex, this condition is even sufficient, so every fixed point of the mapping is already a minimizer. However, the convergence of the iterates does not follow from convexity alone; it has to be shown separately for the Weiszfeld mapping. Under the usual assumption that no iterate coincides with one of the data points a^i, the classical convergence result states that the generated sequence converges to a minimizer of f_{d_2}. A longer proof that this indeed defines a convergent fixed point iteration can be found in the work by Weiszfeld and see the next remark. *(1 P.)*

Variant B: Gradient descent with given step length

The Weiszfeld step can also be written as a gradient descent step with a specific, iteration-dependent step size. For $x \notin A$ we have

$$\nabla f_{d_2}(x) = \sum_{i=1}^m v_i \frac{x - a^i}{\|a^i - x\|_2} = x \sum_{i=1}^m v_i \frac{1}{\|a^i - x\|_2} - \sum_{i=1}^m v_i \frac{a^i}{\|a^i - x\|_2}.$$

Define

$$D(x) := \sum_{i=1}^m v_i \frac{1}{\|a^i - x\|_2}.$$

Then

$$\nabla f_{d_2}(x) = xD(x) - \sum_{i=1}^m v_i \frac{a^i}{\|a^i - x\|_2}.$$

Hence

$$\sum_{i=1}^m v_i \frac{a^i}{\|a^i - x\|_2} = xD(x) - \nabla f_{d_2}(x).$$

Substituting this into the Weiszfeld formula gives

$$x^{new} = \frac{\sum_{i=1}^m v_i \frac{a^i}{\|a^i - x\|_2}}{D(x)} = \frac{xD(x) - \nabla f_{d_2}(x)}{D(x)} = x - \frac{1}{D(x)} \nabla f_{d_2}(x).$$

Therefore the iteration has exactly the form of a gradient method,

$$x^{new} = x - \alpha(x) \nabla f_{d_2}(x) \qquad \text{with} \qquad \alpha(x) = \frac{1}{\sum_{i=1}^m v_i \frac{1}{\|a^i - x\|_2}}.$$

So Weiszfeld's algorithm moves in the negative gradient direction, but with a special step size that is determined by the geometry of the current iterate relative to the anchor points.

Remark 24 (Local contraction of the Weiszfeld map)**.** *To make the fixed point idea more explicit, define the Weiszfeld map*

$$T(x) \coloneqq \frac{\sum_{i=1}^m v_i \frac{a^i}{\|a^i - x\|_2}}{\sum_{i=1}^m v_i \frac{1}{\|a^i - x\|_2}}, \qquad x \notin A.$$

Then the iteration is simply $x^{n+1} = T(x^n)$. We now compute the derivative of T and show that, at a non-degenerate fixed point $x^* = T(x^*) \notin A$, its operator norm is at most 1, and in the generic non-collinear case even strictly smaller than 1.
Set

$$s_i(x) \coloneqq \frac{v_i}{\|a^i - x\|_2}, \qquad W(x) \coloneqq \sum_{i=1}^m s_i(x), \qquad N(x) \coloneqq \sum_{i=1}^m s_i(x) a^i.$$

Then $T(x) = \frac{N(x)}{W(x)}$. Since

$$\nabla \|a^i - x\|_2 = \frac{x - a^i}{\|a^i - x\|_2},$$

we obtain

$$\nabla s_i(x) = v_i \nabla \big(\|a^i - x\|_2^{-1}\big) = v_i \frac{a^i - x}{\|a^i - x\|_2^3}.$$

Hence for a direction $h \in \mathbb{R}^2$,

$$Ds_i(x)[h] = \langle \nabla s_i(x), h \rangle = v_i \frac{\langle a^i - x, h \rangle}{\|a^i - x\|_2^3}.$$

Using the quotient rule for vector-valued maps gives

$$DT(x)[h] = \frac{DN(x)[h] W(x) - N(x) DW(x)[h]}{W(x)^2} = \frac{1}{W(x)} \sum_{i=1}^m Ds_i(x)[h] \, (a^i - T(x)).$$

Substituting the expression for $Ds_i(x)[h]$ yields the explicit formula

$$DT(x)[h] = \frac{1}{W(x)} \sum_{i=1}^m v_i \frac{\langle a^i - x, h \rangle}{\|a^i - x\|_2^3} \, (a^i - T(x)).$$

In particular, if $x^* = T(x^*)$ is a fixed point, then

$$DT(x^*)[h] = \frac{1}{W(x^*)} \sum_{i=1}^m v_i \frac{\langle a^i - x^*, h \rangle}{\|a^i - x^*\|_2^3} \, (a^i - x^*).$$

Therefore the Jacobian matrix at x^* is

$$DT(x^*) = \frac{1}{W(x^*)} \sum_{i=1}^m \frac{v_i}{\|a^i - x^*\|_2^3} (a^i - x^*)(a^i - x^*)^\top.$$

This matrix is symmetric and positive semidefinite. Its operator norm with respect to the Euclidean norm is therefore its largest eigenvalue. Moreover,

$$\operatorname{tr}(DT(x^*)) = \frac{1}{W(x^*)} \sum_{i=1}^m \frac{v_i}{\|a^i - x^*\|_2^3} \operatorname{tr}\big((a^i - x^*)(a^i - x^*)^\top\big).$$

Since $\operatorname{tr}(uu^\top) = \|u\|_2^2$, *we get*

$$\operatorname{tr}(DT(x^*)) = \frac{1}{W(x^*)} \sum_{i=1}^{m} \frac{v_i}{\|a^i - x^*\|_2^3} \|a^i - x^*\|_2^2 = \frac{1}{W(x^*)} \sum_{i=1}^{m} \frac{v_i}{\|a^i - x^*\|_2} = 1.$$

For a symmetric positive semidefinite 2×2 *matrix, the largest eigenvalue is bounded by the trace. Hence*

$$\|DT(x^*)\|_2 = \lambda_{\max}(DT(x^*)) \leq \operatorname{tr}(DT(x^*)) = 1.$$

We even have strict inequality, $\|DT(x^*)\|_2 < 1$, *unless all vectors* $a^i - x^*$ *are parallel, i.e. unless all data points lie on one common line through* x^* *(see Remark 19 for that case). Indeed, equality* $\lambda_{\max} = \operatorname{tr}$ *can only occur if the second eigenvalue is zero, which means that* $DT(x^*)$ *has rank one. This is only possible when all rank-one matrices* $(a^i - x^*)(a^i - x^*)^\top$ *point in the same direction.*
Consequently, in the non-collinear case the fixed point is locally attractive: by continuity of DT, *there is a neighbourhood* U *of* x^* *and a constant* $q < 1$ *such that* $\|DT(x)\|_2 \leq q$ *for all* $x \in U$. *The mean value estimate then gives*

$$\|T(x) - T(y)\|_2 \leq q\|x - y\|_2 \qquad \text{for all } x, y \in U,$$

so T *is a contraction on* U. *By Banach's fixed point theorem, every iteration started sufficiently close to* x^* *converges to* x^*.
This argument is only a local *convergence proof. The full global convergence theorem for the Weiszfeld algorithm is more subtle, in particular because the map* T *is not defined at points of* A *and because the case of a solution located at some* a^k *has to be handled separately.*

Remark 25. *The iteration scheme* (4) *is a convex combination of the given locations* $a \in A$, *where the weights depend on the previous iterate. To see this we define the weights*

$$w_i := \frac{v_i \frac{1}{\|a^i - x\|_2}}{\sum_{i=1}^{m} v_i \frac{1}{\|a^i - x\|_2}}.$$

It is $w_i \geq 0$ *and*

$$\sum_{i=1}^{m} w_i = \sum_{i=1}^{m} \frac{v_i \frac{1}{\|a^i - x\|_2}}{\sum_{i=1}^{m} v_i \frac{1}{\|a^i - x\|_2}} = 1.$$

Thus, the solution x^* *and every iterate* x^n *(*$n \geq 1$*) satisfy* $x^*, x^n \in \operatorname{conv}(A)$. *For* x^n *this is obvious, for* x^* *we use that it satisfies the fix point equation. This result was implicitly used in the proof of Theorem 21.*

Total: 4 Points

11. Prove Theorem 23 above using Theorem 21 and Theorem 22.

Solution:

The relative approximation error / accuracy is defined as

$$\left| \frac{f_{d_2}(x) - f_{d_2}(x^*)}{f_{d_2}(x^*)} \right|,$$

where x^* is the global minimiser, so that $f_{d_2}(x) - f_{d_2}(x^*) \geq 0$. From Theorem 21, we know

that

$$f_{d_2}(x) - f_{d_2}(x^*) \leq \|\nabla f_{d_2}(x)\|_2 \cdot \sigma(x) = UB(x)$$

and from Theorem 22, we know that

$$LB(x) = f_{d_2}(x) - \|\nabla f_{d_2}(x)\| \cdot \sigma(x) \leq f_{d_2}(x^*).$$

Combining these two inequalities (notice the assumption $LB(x) > 0$ or just in general that $f_{d_2}(x) \geq 0$ for all $x \in \mathbb{R}^2$), we get: *(2 P.)*

$$\frac{f_{d_2}(x) - f_{d_2}(x^*)}{f_{d_2}(x^*)} \leq \frac{UB(x)}{LB(x)}$$

If the right side is less than ϵ, then:

$$\text{Relative Accuracy} \leq \frac{UB(x)}{LB(x)} < \epsilon$$

This confirms that the iterate x has at least the desired relative accuracy. *(1 P.)*

Remark 26. *The condition $LB(x) > 0$ is automatically satisfied for all x sufficiently close to a differentiable minimiser $x^* \notin A$. Indeed, in this case*

$$\nabla f_{d_2}(x^*) = 0 \qquad \textit{and} \qquad f_{d_2}(x^*) > 0,$$

where the strict positivity follows unless one is in the trivial case $A = \{a\}$.
Now observe that all functions involved in

$$LB(x) = f_{d_2}(x) - \|\nabla f_{d_2}(x)\|\, \sigma(x)$$

are continuous in a neighbourhood of x^*. The function f_{d_2} is continuously differentiable on $\mathbb{R}^2 \setminus A$, hence both f_{d_2} and $x \mapsto \|\nabla f_{d_2}(x)\|$ are continuous near x^*. Moreover,

$$\sigma(x) = \max_{a \in A} \|a - x\|_2$$

is continuous as the maximum of finitely many continuous functions. Therefore LB itself is continuous at x^*, and

$$LB(x^*) = f_{d_2}(x^*) - \|\nabla f_{d_2}(x^*)\|\, \sigma(x^*) = f_{d_2}(x^*) > 0.$$

By continuity, there exists $\delta > 0$ such that for all x with $\|x - x^*\|_2 < \delta$ we have

$$|LB(x) - LB(x^*)| < \frac{1}{2} f_{d_2}(x^*).$$

Hence for all such x,

$$LB(x) > LB(x^*) - \frac{1}{2} f_{d_2}(x^*) = \frac{1}{2} f_{d_2}(x^*) > 0.$$

So at least locally around a differentiable solution x^*, the denominator in the relative error bound is strictly positive, and the stopping criterion is well defined.

Total: 3 Points

12. Derive a termination criterion for the Weiszfeld algorithm based on Theorem 23.

Solution:

Based on Theorem 23, the algorithm should terminate when the ratio of upper and lower bounds is sufficiently small. The criterion is: Stop if $f_{d_2}(x) - \|\nabla f_{d_2}(x)\|\sigma(x) > 0$ AND *(1 P.)*

$$\frac{UB(x)}{LB(x)} = \frac{\|\nabla f_{d_2}(x)\|\sigma(x)}{f_{d_2}(x) - \|\nabla f_{d_2}(x)\|\sigma(x)} < \epsilon$$

Additionally, a safety check should be included to handle cases where x is very close to an anchor point a^i, as the gradient is not defined there. *(1 P.)*

Total: 2 Points

13. Implement the Weiszfeld algorithm using Python. Test your implementation on the following cases:

    (a) A unit square $A = \{(-1,-1)^T, (-1,1)^T, (1,1)^T, (1,-1)^T\}$ with equal weights $v_i = 1$ for $i = 1, \ldots, 4$. Verify that the algorithm converges to the origin $x^* = (0,0)^T$.

    (b) Create an original test case. Discuss how the set of given locations $A = \{a^1, \ldots, a^m\}$ and weights $v_i, i \in \mathcal{M} = \{1, \ldots, m\}$ affects the location of the minimizer.

Your code must be runnable.

Solution:

First, we give one possible implementation. Notice that here we tried to use many individual functions instead of writing just one minimisation function, because this way we can reuse the individual parts. Also notice that some lines of code had to be broken in order to not go over the page margins of this pdf. *(5 P.)*

```
import matplotlib.pyplot as plt
import numpy as np
import timeit       #for later timing of methods
import warnings
from numpy.linalg import norm

tol = 1e-12             #another tolerance
timeiterations = 100    #for later timing of methods

# Defining helper functions for Weiszfeld and Gradient Descent

def Test_k(A,v):
    for k,ak in enumerate(A):
        Testk=norm(sum(v[i]*(ak-ai)/norm(ak-ai) for i,ai in enumerate(A) if i
     != k))
        if Testk <= v[k]:
            return ak
    return None

def Weiszfeld_iterate(x,A,v):
    return sum(v[i]*ai/norm(ai-x) for i, ai in enumerate(A)) / sum(v[i]/norm(
    ai-x) for i, ai in enumerate(A))

def f_d2(x,A,v):
    return sum(v[i]*norm(ai-x) for i, ai in enumerate(A))

def sigma(x,A):
    return max(norm(ai-x) for ai in A)

def gradf_d2(x,A,v):
    return sum(v[i] * (x-ai)/norm(x-ai) for i, ai in enumerate(A))
```

```
30
31 def UB(x,A,v): #upper bound
32     return norm(gradf_d2(x,A,v))*sigma(x,A)
33
34 def LB(x,A,v): # lower bound
35     return f_d2(x,A,v) - norm(gradf_d2(x,A,v)) * sigma(x,A)
36
37 def squared_problem_minimiser(A, v):
38     return np.average(A, axis=0, weights=v)
39
40 # General minimiser function for this problem where gradient descent or
    Weiszfeld can be used
41 # uses Weiszfeld and squared problem minimiser if no x0 or iteration scheme
    is given
42 def Location_d2_minimiser(A,v,epsilon=1e-8,x0=None,abs_tol=1e-12,
    iterationScheme = Weiszfeld_iterate):
43     if x0 in A:
44         warnings.warn("x0 in A, choose different starting point or use None
    for automatic x0.")
45         x0 = None
46
47     Testk = Test_k(A,v)
48     if  Testk is not None: #
49         return Testk, [Testk], [f_d2(Testk,A,v)]
50     else:
51         x0 = squared_problem_minimiser(A,v)
52
53     X, Z = [x0], [f_d2(x0,A,v)]
54     xnew=x0+np.ones_like(x0)    # make a new vector that is not x0
55     while ((LB(x0,A,v)>0 and (UB(x0,A,v)/LB(x0,A,v)>epsilon)) and (norm(x0-
    xnew) > abs_tol)):
56         x0 = xnew
57         xnew = iterationScheme(x0,A,v)
58         X.append(xnew)
59         Z.append(f_d2(xnew,A,v))
60
61     return x0,X,Z
```

Below are a conceptual description of the implementation for your report and some remarks. First we want a result for the computation of $\sigma(x)$, which can be seen by the fact that $\text{conv}(A)$ is convex polytope and σ is a convex function, but we will prove it in more detail.

Proposition 27. *The value for $\sigma(x)$ is achieved in a given location $a^k \in A$, that means it exists $a^k \in A$ with*

$$\max_{a\in A} d_2(a,x) = d_2(x,a^k) = \sigma(x) = \max_{y\in \text{conv}(A)} d_2(x,y).$$

Proof. Since $\text{conv}(A)$ is closed and bounded, there exists $y \in \text{conv}(A)$ and thus $\lambda \in R^m$ with $\lambda_i \geq 0$ and $\sum_{i=1}^m \lambda_i = 1$ for all $i \in \{1,\dots,m\}$ with $y = \sum_{i=1}^m \lambda_i a^i$ such that $d_2(x,y) = \sigma(x)$.

$$\sigma(x) = \|x-y\|_2 = \left\|x - \sum_{i=1}^m \lambda_i a^i\right\|_2 \leq \sum_{i=1}^m \lambda_i \|x-a^i\|_2 \leq \sum_{i=1}^m \lambda_i \max_{a\in A}\|x-a\|_2 = d_2(x,a^k)$$

On the other hand, from $A \subseteq \text{conv}(A)$ we immediately get $\max_{a\in A} d_2(a,x) \leq \sigma(x)$ and thus proved the claim. □

- In the given implementation most values are saved for later plotting or general inspection of the solution and the method. You do not have to save these.

- Since the termination criterion from task 12 requires $LB(x) > 0$, it is required to have this in the code and it would be good to add another criterion in case this is not satisfied. We added the condition that we stop if the distance between successive iterates does not change more than a given tolerance. Another typical condition is to check whether the objective function values of successive iterates change enough, but since our objective function basically is a distance we do not need to do both checks. $\nabla f(x) = 0$ can also be checked.

- It is not expected, but we should pay attention to one of the iterates becoming a given location a^i, because then the iteration would break. A possible fix is to check whether that happens and possibly changing the result into $x + (\epsilon_1, \epsilon_2)^T$ with maybe randomly chosen small ϵ_1, ϵ. Luckily, we have Remark 25, telling us that we never exactly hit one of the given locations on the boundary $\partial \operatorname{conv}(A)$, because for this the weight w_i would have to be zero. For locations in the inside int(conv(A) there exist multiple possible convex combinations to achieve these points, so we might still land on these. Choosing to set that part of the gradient compotutation to zero is a good idea, because the result will be a feasible subgradient. The step in the algorithm will still turn out to be unfavourable in many situations.

- **Case (i): Unit Square.** Due to the symmetry of the four points $\{(-1,-1),(-1,1),(1,1),(1,-1)\}$, the gradient at $(0,0)$ is the sum of four vectors of equal length pointing in opposite directions, resulting in $\nabla f(0,0) = 0$. The algorithm correctly identifies the origin as the geometric median. If the starting point is chosen by minimising the squared Euclidean distances, we thus find the solution without going into the iteration process. If the starting point is of any of the lines connecting the origin with an anchor, the iteration will converge on that line straight to the origin. You might hit the anchor, which can lead to different behaviour. *(1 P.)*

- **Case (ii): Weighted Case.** If one weight v_i is significantly larger than the others (specifically if $v_k \geq \text{Test}_k$), the minimiser x^* will coincide with a^k. Otherwise, the minimiser is pulled towards areas with higher density of weights and anchor points. *(2 P.)*

Total: 8 Points

14. Implement the gradient descent method with backtracking and replace steps (c)-(d) in the above algorithm with it. Test your implementation on the cases from above. Compare with the Weiszfeld algorithm and discuss the difference between the two algorithms and their performance. Which algorithm would you suggest, and why?

Solution:

We only need to implement the iteration scheme for the gradient descent method with backtracking and then give this as optional argument to our solver. Theoretically, we now really would have to check whether an iteration lands on a given location, $x^n = a^k \in A$, and somehow deal with that, but we still do not have a better strategy than before. *(1 P.)*

```
def gradientDescent_iterate(x,A,v):
    alpha = 1
    c = 0.1
    gradient = gradf_d2(x,A,v)
    while (f_d2(x-alpha*gradient,A,v) > f_d2(x,A,v)- c * alpha * norm(
    gradient)): #Armijo condition
        alpha *= 0.5
    return x - alpha*gradient
```

- **test case:** Both algorithms should give the same solution(s). *(1 P.)*
- **Comparison/Performance:** Weiszfeld usually converges faster for the Fermat-Weber problem because it utilizes the specific structure of the objective function (pseudo-convexity = convexity). Gradient descent can struggle with the non-differentiability at anchor points unless smoothed. You might see that with fewer points or good geometries gradient descent outperforms Weiszfeld. It usually has a somewhat chaotic iteration path and thus can make hugh improvements. For this to happen the initial step size (in the code alpha) needs to be big enough. It however does nt scale that nicely with the number of points. For large point sets the number of non-regular points is quite high and the gradient descent method slows down or may not even converge (due to numerical instability). In contrast, the Weiszfeld algorithm is relatively stable. The number of used iterations to achieve a given tolerance does not grow that quickly with the number of anchors as it does for gradient descent and the iteration paths remain comparatively smooth. *(1 P.)*
- **Recommendation:** Gradient descent with backtracking line search is a general-purpose optimizer. Weiszfeld's method is a specialized fixed-point iteration.. Weiszfeld is suggested due to its simplicity (no step-size parameter needed) and robust convergence properties for this specific class of problems. For specific settings, gradient descent might be much better though. Stay true to your finding! *(1 P.)*

Total: 4 Points

4 Suggested Reading Material

- Love, Morris and Wesolowsky (1988): Facility Location.
- Beck and Sabach (2015): Weiszfeld's Method.

Note that the solution proposal is much more detailed than it was expected from you!

Grading Guide Summary

Task / Criteria	Points
Task 1	**12**
• d_1 definiteness	1
• d_1 symmetry	1
• d_1 triangle inequality - It is not allowed to simply use that $\|\cdot\|_1$ is a norm and thus satisfies the triangle inequality.	1
• d_2 definiteness	1
• d_2 symmetry	1
• d_2 triangle inequality. Simply using that $\|\cdot\|_2$ satisfies the inequality and thus d_2 does, only gives 1P (if shown correctly).	3
• d_∞ definiteness	1
• d_∞ symmetry	1
• d_∞ triangle inequality. It is not enough to simply use that $\|\cdot\|$ satisfies the inequality - this gives 1P (if correct)	2
Task 2	**3**
• 1 pt for each unit ball (axes, units, not just the sphere)	3
Task 3	**2**
• Prove that norms are convex	2
Task 4	**6**
• Show that $d_i(a, \cdot)$ is convex (For variant B the remark has to be somewhere (for task 7).)	1
• Prove that the maximum (of convex functions) is convex	2
• Prove that the sum (of convex functions) is convex	2
• Conclude convexity of our objective functions	1
Task 5	**2**
• Any valid interpretation giving an explanation and reason	2
Task 6	**3**
• correct separation	1
• Explain how to get sub-problems	1
• Give any solution idea of sub-problems	1
Task 7	**5**
• Show convexity of objective function (A: right conditions + proof + conclusion, B: gradient, Hessian, conclusion	3
• Found minimiser (condition + computation)	2
Task 8	**7**
• differentiable case: correct gradient	1
• differentiable case: $\nabla f = 0$	1
• non-differentiable case: general optimality condition	1
• non-differentiable case: Calculate directional derivative / subgradient	1
• non-differentiable case: Calculations for finding condition (A: CS or cos, B:Minkowski sum)	2
• non-differentiable case: end result	1
Task 9	**4**
• give example	1
• Find the minimum in both cases (2P) and give the calculations (1P for d_2). It is also ok to say that the minimum for the d_2^2 problem does not satisfy the optimality conditions for the d_2 problem as long as it is explained.	3
Task 10	**4**

• compute gradient (refering to task 8 here is ok)	1
• A: Use the optimality condition and reformulate. B: Reformulate to gradient descent.	2
• A: Notice that this is only a fixed point iteration scheme. B: Write down the complete gradient descent scheme.	1
Task 11	**3**
• Use the two inequalities to prove that the relative accuracy is smaller than UB/LB.	2
• relative accuracy $< \epsilon$	1
Task 12	**2**
• main criterion	1
• Mention that $LB > 0$ and that we might hit an a^k. Also see the code for this, giving the point if implemented.	1
Task 13	**8**
• Correct implementation/use of f_{d_2}, ∇f_{d_2}, $\sigma(x)$, UB(x), LB($x,$) (1P), Correct implementation of Test$_k$ and check whether a^k is solution (1P), Correct implementation to find x_0, maybe with $f_{d_2^2}$ (1P), Correct Weiszfeld implementation testing Test$_k \leq v_k$ and using the stopping criterion (1P) and with the correct iteration and rest (1P)	5
• First test case - correct result and implementation	1
• Second test case - Choose any set of points and weights and try the algorithm (1P), discuss the influence of the data and interpret your own test case (1P)	2
Task 14	**4**
• Implementation of gradient descent	1
• Test your implementation on both test cases from before.	1
• Comparison of both methods. The point is given as long as the discussion matches the behaviour in the test cases or a good explanation is given.	1
• Suggestion of method. The point is given as long as it matches the observations from the tst cases.	1
writing poper math	**3**
logic	**1**
extra (1P) if proven Proposition 27 ($\sigma(x) = \max_{a \in A} d_2(x,a)$) or Remark 25 ($x^n \in \operatorname{conv}(A)$)	**+1**
extremely good numerical experiments	**+1**
Project total	**69**