NTNU
Kunnskap for en bedre verden

Department of Mathematics

TMA4180 - Optimization 1

Project 1

Authors:
Haakon Muggerud, Åsmund Engmark & Tinius Mellbye

September, 2020

Part 1 – Theoretical Work

In this part we are going to prove and establish some of the theoretical aspects used in Part 2.

Problem 2.2.1

The aim of problem is to show that the Manhattan distance, Euclidean distance and the maximum distance is indeed metrics. This is done by showing that they satisfy the conditions of definiteness, symmetry, and the triangle inequality.

1. Manhattan distance:

$$d_1(x, y) = |x_1 - y_1| + |x_2 - y_2|$$

- Definiteness:

$$\begin{aligned} d_1(x, y) &= |x_1 - y_1| + |x_2 - y_2| = 0 \\ |x_1 - y_1| &= 0 \Rightarrow x_1 = y_1 \\ |x_2 - y_2| &= 0 \Rightarrow x_2 = y_2 \\ \Rightarrow x &= y \end{aligned}$$

- Symmetry:

$$\begin{aligned} d_1(x, y) &= |x_1 - y_1| + |x_2 - y_2| \\ &= |-(y_1 - x_1)| + |-(y_2 - x_2)| \\ &= |(y_1 - x_1)| + |(y_2 - x_2)| = d_1(y, x) \end{aligned}$$

- Triangle inquality:

$$\begin{aligned} d_1(x, y) &= |x_1 - y_1| + |x_2 - y_2| \\ &= |x_1 - z_1 + z_1 - y_1| + |x_2 - z_2 + z_2 - y_2| \\ &\leq |x_1 - z_1| + |z_1 - y_1| + |x_2 - z_2| + |z_2 - y_2| = d_1(x, z) + d_1(z, y) \end{aligned}$$

2. Euclidean distance

$$d_2(x, y) = \sqrt{(x_1 - y_1)^2 + (x_1 - y_1)^2}$$

- Definiteness:

$$\begin{aligned} d_2(x, y) &= \sqrt{(x_1 - y_1)^2 + (x_1 - y_1)^2} \\ (x_1 - y_1)^2 &= 0 \rightarrow x_1 = y_1 \\ (x_2 - y_2)^2 &= 0 \rightarrow x_2 = y_2 \\ x &= y \end{aligned}$$

- Symmetry:

$$\begin{aligned} d_2(x, y) &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \\ &= \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2} \\ &= d_2(y, x) \end{aligned}$$

- Triangle inequality:

$$d_2(x, z) \leq d_2(x, y) + d_2(y, z)$$

To make the algebra more manageable, we move $x, y, \text{and} z$ in the plane so that y is at the origin. Then the inequality is

$$\begin{aligned}
&\sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2} \leq \sqrt{x_1^2 + x_2^2} + \sqrt{z_1^2 + z_2^2} \\
\iff &(x_1 - z_1)^2 + (x_2 - z_2)^2 \leq (\sqrt{x_1^2 + x_2^2} + \sqrt{z_1^2 + z_2^2})^2 \\
&\qquad = (x_1^2 + x_2^2) + 2\sqrt{x_1^2 + x_2^2}\sqrt{z_1^2 + z_2^2} + (z_1^2 + z_2^2) \\
\iff &-2(x_1 z_1 + x_2 z_2) \leq 2\sqrt{x_1^2 + x_2^2}\sqrt{z_1^2 + z_2^2}.
\end{aligned}$$

We may assume the left hand side is positive, since when it is negative the inequality is trivial. Then we can square both sides and get

$$\begin{aligned}
(x_1 z_1)^2 + 2x_1 x_2 z_1 z_2 + (x_2 z_2)^2 &\leq (x_1^2 + x_2^2)(z_1^2 + z_2^2) \\
&= (x_1 z_1)^2 + (x_2 z_2)^2 + (x_1 z_2)^2 + (x_2 z_1)^2 \\
\iff (x_1 z_2)^2 - 2(x_1 z_2)(x_2 z_1) + (x_2 z_1)^2 &= (x_1 z_2 - x_2 z_1)^2 \\
&\geq 0
\end{aligned}$$

this proves the inequality

3. Maximum distance

$$d_\infty(x, y) = \max(|x_1 - y_1|, |x_2 - y_2|)$$

- Definiteness:

$$\begin{aligned}
\max(|x_1 - y_1|, |x_2 - y_2|) &= 0 \\
|x_1 - y_1| = 0 &\Rightarrow x_1 = y_1 \\
|x_2 - y_2| = 0 &\Rightarrow x_2 = y_2 \\
x &= y
\end{aligned}$$

- Symmetry. Using the fact that $|x - y| = |-(y - x)| = |y - x|$:

$$\max(|x_1 - y_1|, |x_2 - y_2|) = \max(|y_1 - x_1|, |y_2 - x_2|)$$

- Triangle inequality:

$$\begin{aligned}
\max(|x_1 - y_1|, |x_2 - y_2|) &= \max(|x_1 - z_1 + z_1 + y_1|, |x_2 - z_2 + z_2 + -y_2|) \\
&\leq \max(|x_1 - z_1| + |z_1 + y_1|, |x_2 - z_2| + |z_2 + -y_2|) \\
&\leq \max(|x_1 - z_1|, |x_2 - z_2|) + \max(|z_1 - y_1|, |z_2 - y_2|) = d_\infty(x, z) + d_\infty(z, y)
\end{aligned}$$

Problem 2.2.2

Here we plot the plot the unit balls $B_i(0,1) := \{x \in R^2 | ||x||_i \leq 1\} (i \in 1, 2, \infty)$ of the so-called Manhattan norm, the Euclidean norm and of the maximum norm.

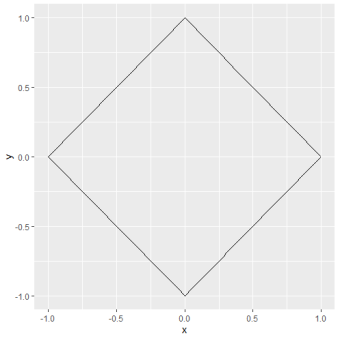


Figure 1: Unit ball of the Manhattan norm

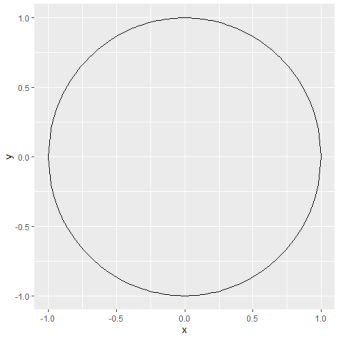


Figure 2: Unit ball of the Euclidean Norm

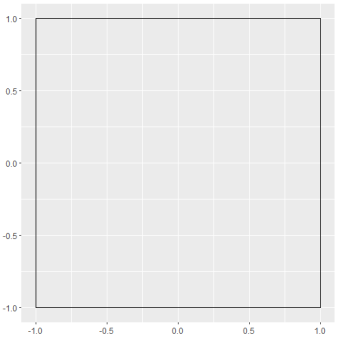


Figure 3: Unit ball of the maximum norm

Problem 2.2.3

Using the properties in the definition of a norm given, we can show that every norm is convex:

$$\begin{aligned} ||\lambda v + (1-\lambda)w|| &\leq ||\lambda v|| + ||(1-\lambda)w|| \\ &= \lambda||v|| + (1-\lambda)||w|| \end{aligned}$$

Where the first inequality follows from the triangle-inequality and the second follows from positive homogeneity property.

Problem 2.2.4

Here we will show that the objective functions in Problem (2.1) and Problem (2.2) from Köbis (2021) are convex.

We have already shown that all the norms are convex. Therefor for Problem (2.1) we need to show that the max of convex functions is convex: Let $g = \max(g_1, ..., g_m)$, where g_j is the metric for point j.

$$\begin{aligned} g(\lambda x + (1-\lambda)y) &= \max(g_1(\lambda x + (1-\lambda)y), ..., g_m(\lambda x + (1-\lambda)y)) \\ &\leq \max(\lambda g_1(x) + (1-\lambda)g_1(y), ..., \lambda g_m(x) + (1-\lambda)g_m(y)) \\ &\leq \lambda \max(g_1(x), ..., g_m(y)) + (1-\lambda)\max(g_1(y), ..., g_m(y)) \\ &= \lambda g(x) + (1-\lambda)g(y) \end{aligned}$$

For problem (2.2) we need to show that the sum of convex functions are convex. Let $f = f_1 + f_2 + ... + f_m$, where m is the number if points in A, and f_j is the metric for point j:

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &= f_1(\lambda x + (1-\lambda)y) + ... + f_m(\lambda x + (1-\lambda)y) \\ &\leq \lambda(f_1(x) + ... + f_m(x)) + (1-\lambda)(f_1(y) + ... + f_m(y)) \\ &= \lambda f(x) + (1-\lambda)f(y) \end{aligned}$$

Problem 2.2.5

The geometrical interpretation of solving the center problem with euclidean distance

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

(2.1) in Köbis (2021) is to find the center, x, of the circle with the smallest radius that makes sure that every point, a_i, is contained within the circle. x then becomes the center and $\max d(a_i, x)$ becomes the radius.

Problem 2.2.6

The Manhattan distance moves with the axis, meaning in the 2D-plane it moves horizontal and vertical not across in any way. This way we can separate the sum of Manhattan distances by looking at the horizontal distance and the vertical distance. Which in turn means that we only need to optimize a single variable per sum.

$$\sum_{a \in A} d_1(a, x) = |a_{11} - x_1| + |a_{12} - x_2| + ... + |a_{N1} - x_1| + |a_{N2} - x_2|$$

We can separate the sum into x_1 and x_2

$$\begin{aligned} &|a_{11} - x_1| + ... + |a_{N1} - x_1| + |a_{12} - x_2| + ... + |a_{N2} - x_2| \\ &\sum_{i=1}^{N} |a_{i1} - x_1| + \sum_{i=1}^{N} |a_{i2} - x_2| \end{aligned}$$

These sums are minimized by the median of the points in the sum.

Problem 2.2.7

We already know that the Euclidean norm is convex (Problem 2.2.3), so it's sufficient to show that the square of a positive convex function is convex. We start out by squaring both sides of the convex condition

$$\begin{aligned} f^2(\lambda x + (1-\lambda)y) &\leq \lambda^2 f^2(x) + (1-\lambda)^2 f^2(y) + 2\lambda(1-\lambda)f(x)f(y) \\ &= \lambda^2 f^2(x) + (1-\lambda)^2 f^2(y) + 2\lambda(1-\lambda)f(x)f(y) - \lambda f^2(x) - (1-\lambda)f^2(y) + \lambda f^2(x) + (1-\lambda)f^2(y) \\ &= -\lambda(1-\lambda)(f(x)-f(y))^2 + \lambda f^2(x) + (1-\lambda)f^2(y) \leq \lambda f^2(x) + (1-\lambda)f^2(y) \end{aligned}$$

This shows that the square of a positive convex function is convex. In Problem 2.2.4 we showed that the sum of convex function is also convex, so the objective function is convex.
When finding the uniquely determined minimizer we start by rewriting the function.

$$\begin{aligned} \sum_{i=1}^{N}(a_i - x)^2 &= \sum_{i=1}^{N}(a_i - m + m - x)^2 \\ &= \sum_{i=1}^{N}((a_i - m)^2 + 2(a_i - m)(m - x) + (m - x)^2) \end{aligned}$$

We can now split the sum, because only two is dependent of a, and look at the middle expression.

$$\sum_{i=1}^{N} 2(a_i - m)(m - x) = 2(m - x)\sum_{i=1}^{N}(a_i - m)$$

Here $\sum_{i=1}^{N}(a_i - m)$ is 0 only when $m = \bar{a}$. Further,

$$\sum_{i=1}^{N}(m - x)^2 = k(m - x)^2$$

So after switching m and splitting the equation we end up with.

$$\sum_{i=1}^{N}(a_i - x)^2 = k(\bar{a} - x)^2 + \sum_{i=1}^{N}(a_i - \bar{a_i})^2$$

So the x that minimizes the problem is $x = \bar{a} = (a_1 + ... + a_N)/N$.

Problem 2.2.8

Lets start out by calculating the partial derivatives of the function euclidean distance.

$$\begin{aligned} \frac{\partial||a-x||}{\partial x_1} &= \frac{x_1 - a_1}{\sqrt{(a_1 - x_1)^2 + (a_2 - x_2)^2}} \\ \frac{\partial||a-x||}{\partial x_2} &= \frac{x_2 - a_2}{\sqrt{(a_1 - x_1)^2 + (a_2 - x_2)^2}} \end{aligned}$$

By just looking at the partial derivatives of the euclidean distance we can see that optimal solution can not be a part of the sets of points, as the partial derivative then becomes undefined. So the necessary optimality condition for minimizing the problem is that the x that minimizes can not be a point in A.

Problem 2.2.9

Consider the points $(0,0)$ and $(1,0)$. It is clear that y must be 0, so we only need to consider x to find the minimizer. Then the point that minimizes the Euclidean squared distance $h(x) = ((x-0)^2 + (0-0)^2) + ((x-1)^2 + (0-0)^2) = 2x^2 - 2x + 1$ is the median $(0.5, 0)$, which can be seen by taking the derivative and setting it to zero

$$h'(x) = 4x - 2 = 0 \Rightarrow x = \frac{1}{2}\,.$$

Furthermore, from the second derivative we know that we have attained a minimum;

$$h''(x) = 4 > 0 \;\forall x\,.$$

Though, if we want to minimize the unsquared euclidean distance function

$$\sqrt{(x-0)^2 + (0-0)^2} + \sqrt{(x-1)^2 + (0-0)^2} = x + (1-x) = 1\,,$$

we see that any point on the line $l : \lambda(0,0) + (1-\lambda)(1,0), \lambda \in (0,1)$, will minimize the distance, so the minimization sets are different. The reason for this is that squared functions "punish" large distances from a point, while for linear distance functions there are no extra punishment for large distances, i.e. for linear distance functions an increased distance to some point can be retributed excactly by an equal diminishment of distance to another point.

Part 2 – Weiszfeld algorithm

In this part we will consider the median problem with weighted Euclidean distance

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

where $\mathcal{M} = \{1, \ldots, m\}$ and $0 \leq v^i \in \mathbb{R}$, $\forall i \in \mathcal{M}$.

We will here look into the Weiszfeld algorithm for solving the Problem P_{d_2}.

Problem 1

The objective function is

$$f_{d_2}(x) = \sum_{i\in\mathcal{M}} v^i d_2(a^i, x) = \sum_{i=1}^{m} v^i \sqrt{\sum_{j=1}^{J} (a^i_j - x_j)^2}\,. \tag{1}$$

As stated in the project description "the Weiszfeld algorithm is based on the first-order necessary condition for the stationary point of the objective function" Köbis, 2021. The first order necessary condition can be found in Nocedal and Wright (2006, page 14). To find a stationary point we need to solve

$$\nabla f_{d_2}(x) = \begin{pmatrix} \frac{\partial f_{d_2}}{\partial x_1} \\ \vdots \\ \frac{\partial f_{d_2}}{\partial x_J} \end{pmatrix} = \mathbf{0} \tag{2}$$

We aim to solve Equation 2 for an arbitrary component $\frac{\partial f_{d_2}}{\partial x_k}$ where $k \in \{1, \ldots, J\}$ (note: x_k is the k^{th} element of x not the k^{th} iterate.) which then gives the solution for the entire system.

$$\begin{aligned}
\frac{\partial f_{d_2}}{\partial x_k} &= \frac{\partial}{\partial x_k} \sum_{i=1}^{m} v^i \sqrt{\sum_{j=i}^{J} (a_j^i - x_j)^2} \\
&= \sum_{i=1}^{m} v^i \frac{\partial}{\partial x_k} \sqrt{\sum_{j=i}^{J} (a_j^i - x_j)^2} \\
&= \sum_{i=1}^{m} v^i \frac{1}{2} \frac{1}{\sqrt{\sum_{j=i}^{J} (a_j^i - x_j)^2}} \frac{\partial}{\partial x_k} \sum_{j=i}^{J} (a_j^i - x_j)^2 \\
&= \sum_{i=1}^{m} v^i \frac{1}{2} \frac{1}{\sqrt{\sum_{j=i}^{J} (a_j^i - x_j)^2}} 2(a_k^i - x_k)(-1) \\
&= \sum_{i=1}^{m} v^i \frac{x_k - a_k^i}{d_2(a^i, x)} \\
&= x_k \sum_{i=1}^{m} \frac{v^i}{d_2(a^i, x)} - \sum_{i=1}^{m} v^i \frac{a_k^i}{d_2(a^i, x)}
\end{aligned}$$

Setting the last expression to zero yields

$$x_k^{new} = \frac{\sum_{i=1}^{m} \frac{v^i a_k^i}{d_2(a^i, x)}}{\sum_{i=1}^{m} \frac{v^i}{d_2(a^i, x)}} \qquad \forall k \in \{1, \ldots, J\} \tag{3}$$

Moreover, because f_{d_2} is convex, as it is a sum of convex functions (Problem 2.2.4), and clearly continuous, we have from theorem 2.5 in Nocedal and Wright (2006, page 16) that for an x^* such that $\nabla f_{d_2}(x^*) = 0$, f_{d_2} obtains a global minimum.

Problem 2

The definition of *relative accuracy* can be found on page 3 in Kreinovich (1999), named "Definition 1".

Proof.

$$\epsilon > \frac{||\nabla f_{d_2}||\sigma(x)}{f_{d_2}(x) - ||\nabla f_{d_2}(x)||\sigma(x)} \overset{\text{Theorem 3}}{\geq} \frac{||\nabla f_{d_2}||\sigma(x)}{f_{d_2}(x^*)} \tag{4}$$

$$\overset{\text{Theorem 2}}{\geq} \frac{f_{d_2}(x) - f_{d_2}(x^*)}{f_{d_2}(x^*)} \tag{5}$$

$$= \frac{|f_{d_2}(x) - f_{d_2}(x^*)|}{|f_{d_2}(x^*)|} \tag{6}$$

The first expression is what is stated in Theorem 4. The last equality follows from the fact that $f_{d_2}(x) \geq f_{d_2}(x^*)\ \forall x$ which in turn leads to

$$\epsilon > \frac{|f_{d_2}(x) - f_{d_2}(x^*)|}{|f_{d_2}(x)|} \tag{7}$$

Hence, the relation is strongly relatively ϵ-close and consequently, relatively ϵ-close. □

Problem 3

We use the termination criterion suggested in Theorem 4 directly. So we use

$$\frac{||\nabla f_{d_2}||\sigma(x)}{f_{d_2}(x) - ||\nabla f_{d_2}(x)||\sigma(x)} < \epsilon \tag{$\mathcal{S}$}$$

for some small ϵ as stopping criterion. f and ∇f is easily computed, but we need a way to compute $\sigma(x)$. To do this, first notice that $C = \text{conv}\{a_1, ..., a_m\}$ is a convex polygon where the vertices are points in A. This is clear since finding the convex hull of finitely many discrete points in $\mathbb{R}^2$ can intuitively be thought of as stretching a tight rubber band around the entirety of the points. Thus

$$\mathcal{C} = \{b_i, i \in \{1, \cdots, p\}\}$$

where $b_i \in A$ are the vertices of C. Now we want to show that $\sigma(x)$ (the maximal distance from $x \in \mathbb{R}^2$ to $\mathcal{C}$) is found as the line from x to one of the vertices b_k of $\mathcal{C}$, i.e. not on any of the edges $\overrightarrow{b_i b_{i+1}}$ of $\mathcal{C}$ or in the interior. The proof of this fact is as follows:

Proof. Firstly, assume for contradiction that the maximum is found at an interior point, y' so that $\sigma(x) = |\overrightarrow{xy'}|$. However, by moving from y' away from x of a distance of δ to y'', we have that $\sigma(x) = |\overrightarrow{xy''}| = |\overrightarrow{xy'}| + |\delta| > |\overrightarrow{xy'}|$. This contradicts our assumption, hence the maximum is not attained for an interior point.

Secondly, assume for contradiction that the maximum is found at one of the edges of C, say $\sigma(x) = |\overrightarrow{xq}|$ where $q = \lambda b_j + (1-\lambda)b_{j+1}, \lambda \in (0,1)$. The line $\overrightarrow{xq}$ makes an angle θ with the normal of the edge h. Now we may simply increase θ marginally to $\theta + \delta$, getting a point $p \in \overrightarrow{b_i b_{i+1}}$ on the edge which is so that $|\overrightarrow{xp}| = \frac{|h|}{\cos(\theta+\delta)} > \frac{|h|}{\cos(\theta)} = |\overrightarrow{xq}|$, contradicting that q generates the maximum. Thus we have shown that $\sigma(x)$ will definitely be on one of the vertices of C, so that we have the computation formula $\sigma(x) = \max_{i \in \{1,\cdots,p\}} d_2(x, b_i)$ with b_i as defined above. □

We use this to implement the stopping criterion in the code.

Problem 4

Here we implement the Weiszfeld algorithm to solve Problem P_{d_2}.

Pseudo code:

Algorithm 1 Our Weiszfeld algorithm

Input:
$A = [a^1, ..., a^i, ..., a^m]^T$ where $a^i = [a_1^i, a_2^i]^T$
$v = [v^1, ..., v^i, ..., v^m]^T$ where $0 < v^i \in \mathbb{R}, \forall i \in \mathcal{M}$
$\epsilon > 0$

if theorem 1 is fullfilled for a k (Köbis, 2021) **then**
  $x^* = a^k$
  return x^*
initialize $x = [x_1, x_2]^T$ ▷ We have used the element-wise mean of the a^i-s (see Problem 5)
while stopping criterion not met **do** ▷ We used the stopping criterion as described above.
  $x_1^{new} = \frac{\sum_{i=1}^m v^i a_1^i / d_2(a^i, x)}{\sum_{i=1}^m v^i / d_2(a^i, x)}$
  $x_2^{new} = \frac{\sum_{i=1}^m v^i a_2^i / d_2(a^i, x)}{\sum_{i=1}^m v^i / d_2(a^i, x)}$
  $x = x^{new}$
return x

As a test example we will use

$$A = \begin{bmatrix} 0 & 4 & 8 & 6 & 10 \\ 0 & 0 & 10 & 6 & 4 \end{bmatrix}$$

where column i of A corresponds to a_i for $i \in \{1, \ldots, m\}$; $m = 5$ in this example. Moreover, for our result to correspond to the software used to make Figure 4, we set all the weights $v_i = 1$.

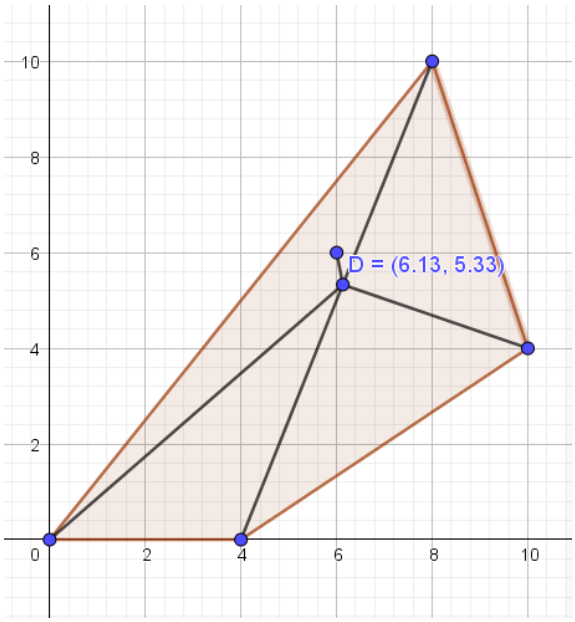


Figure 4: Visual representation of the points (D is the minimizer, the other points are the locations).

```
In [586]:
1  A_test = np.array([[0.,4.,8.,6.,10.],[0.,0.,10.,6.,4.]])
2
3  v_test = np.array([1.,1.,1.,1.,1.])
4
5  ourWeiszfeld(A_test, v_test)[0]
6

Out[586]: array([6.13064863, 5.33042671])
```

Figure 5: The code example with the minimum point as an output.

This example is illustrated by Figure 4 and the result of our algorithm when $\epsilon = 10^{-6}$ is shown in Figure 5.

Problem 5

In the comparison of the Steepest Descent algorithm and the Weiszfeld algorithm both perform the test described in step 1 of the algorithm(Köbis, 2021, page 3). For step 2 of the algorithm we have set the initial value of $x = [x_1, x_2]^T = [\bar{a_1}, \bar{a_2}]$ where $\bar{a_1} = \frac{1}{m}\sum_{i=1}^{m} a_1^i$ and $\bar{a_2} = \frac{1}{m}\sum_{j=1}^{m} a_2^j$, in other words the element-wise means. Here $a_k^l \in [0, 100]$ $\forall k \in \{1, 2\}$ and $l \in \{1, \ldots, m\}$.

Speed analysis

In this section both of the algorithms use the same stopping criterion, with $\epsilon = 10^{-6}$, as described above.

One could set ϵ smaller than 10^{-6}, but we decided not to do so in the analysis below to save time.

		Weiszfeld	Steepest descent
Time	Mean	0.0397934	0.935247
	Var	0.0278188	0.0217527
Iterations	Mean	65.35	1154.18
	Var	79366.2	23497.1

(a) $m = 5$

		Weiszfeld	Steepest descent
Time	Mean	0.0410592	0.988823
	Var	0.00140061	0.494118
Iterations	Mean	42.97	991.32
	Var	1471.69	328215

(b) $m = 10$

		Weiszfeld	Steepest descent
Time	Mean	0.0446668	0.924519
	Var	0.0018262	0.0189662
Iterations	Mean	44.69	1089.48
	Var	1086.37	11703.3

(c) $m = 20$

		Weiszfeld	Steepest descent
Time	Mean	0.0951673	0.881625
	Var	0.00096526	0.0208993
Iterations	Mean	70.19	1061.91
	Var	280.274	10325.7

(d) $m = 50$

		Weiszfeld	Steepest descent
Time	Mean	0.720073	0.8363
	Var	0.00937541	0.0248575
Iterations	Mean	210.92	1033.02
	Var	1387.69	12650.7

(e) $m = 200$

		Weiszfeld	Steepest descent
Time	Mean	15.9446	1.11879
	Var	1.24391	0.0409634
Iterations	Mean	1014.99	1332.83
	Var	4.4699	10.4411

(f) $m = 1000$

Figure 6: Tables of performance measures for $m \in \{5, 10, 20, 50, 200, 1000\}$ for 100 runs per m. The measure can be described as:
Time $\rightarrow$ Mean: is calculated as the average time spent on the 100 runs.
Time $\rightarrow$ Var: is the sample variance of the time for the 100 runs.
Iter $\rightarrow$ Mean: is the average of the number of iterations the algorithms need on each of the 100 runs.
Iter $\rightarrow$ Var: is the sample variance of the number of iterations for the 100 runs.

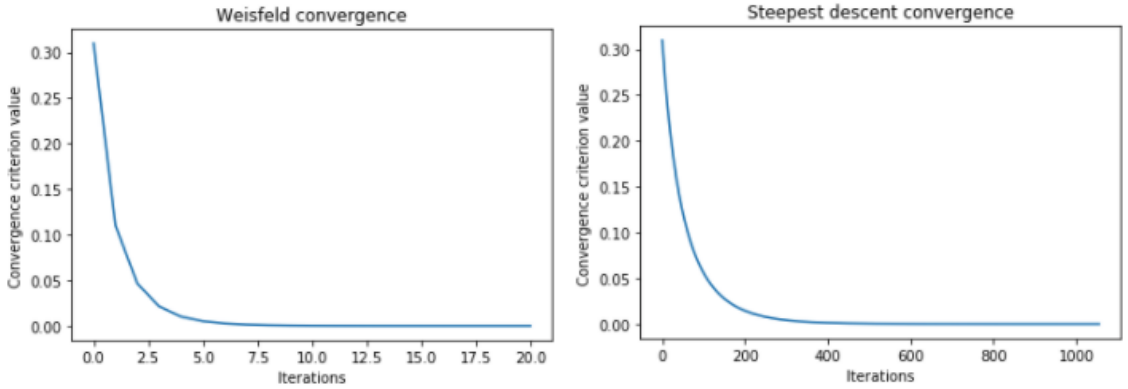
From the tables of Figure 6 one can observe some interesting characteristics of the algorithms. Firstly, the mean time of the steepest descent algorithm appear to decrease for larger m. The differences are small so we can not be curtain that the mean time is negatively correlated with m, though we can be fairly confident that it do not increase. In contrast, the Weiszfeld algorithm has an increase in the mean time from $m = 10$ (Figure 6b) to $m = 1000$ (Figure 6f) with a surge from 0.72, for $m = 200$ (Figure 6e), to 15.9446, for $m = 1000$. The variance of the time spent by the two algorithms are mostly stable, though, for the Weiszfeld algorithm the variance increase from about 0.009375, for $m = 200$, to 1.2439, for $m = 1000$ an increase of a factor of about 133.

In terms of number of iterations we can observe some interesting traits. Firstly, the mean number of iterations needed by the Steepest Descent algorithm is stably, although their are some variations. Secondly, the variance of the iterations for the Steepest Descent algorithm is also relatively stable for $m = 5$ (Figure 6a) to $m = 200$, but drops significantly for $m = 1000$ with a sample variance of 10.44. On the other hand, the Weiszfeld algorithm have a highly varying mean and variance in terms of number of iterations. The mean number of iterations needed are generally fewer than that of the Steepest Descent, but appear to increase significantly as m goes into the hundreds and thousands. The variance drops also for the Weiszfeld algorithm when $m = 1000$, to 4.4699. This decrease in variance can come as a result of it being many points in a relatively small region from which the a^i-s can be generated. The weights and points can be thought of as "averaging" each other out and a minimum is more often located close to the center of the region, which in turn decrease the variance. Thus the drop in variance is likely to be problem-specific rather than a consequence of the algorithms used.

Based on what we find in Figure 6 we would recommend the Weiszfeld for $m \leq 200$, because the Weiszfeld algorithm is faster. For larger m we would recommend the Steepest Descent algorithm, as the Weiszfeld show indication of a large increase in number of iterations.

Effect of ϵ on the convergence

Different problems demand different precision. Therefore, we will briefly look into the effect changing ϵ has on the convergence. In this part we only consider specific examples with $m = 10$ and all the plots have the value of the left side of inequality $\mathcal{S}$ as y-axis and iterations as x-axis.

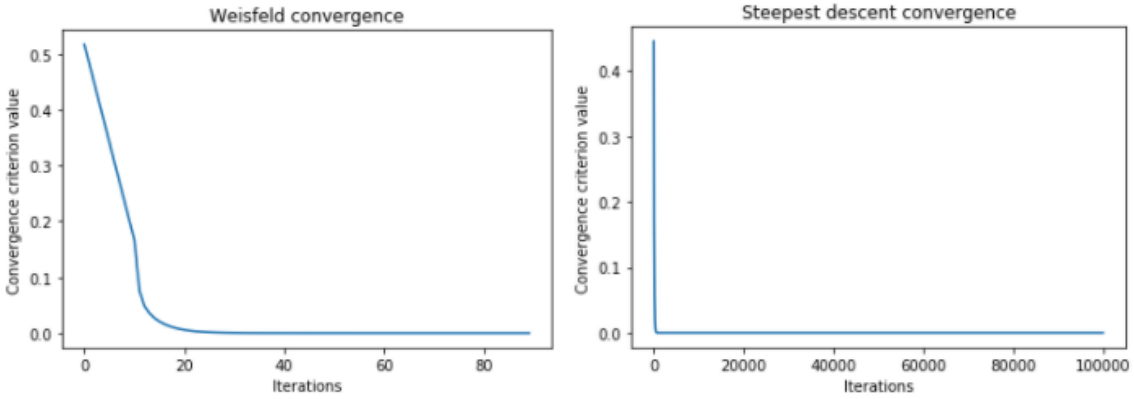


(a) Convergence plot of the Weiszfeld algorithm. (b) Convergence plot of the Steepest Descent.

Figure 7: Plots of the convergence of the two algorithms for one realisation where $m = 10$ and $\epsilon = 10^{-6}$.

Figure 7 show the convergence of the two algorithms when $\epsilon = 10^{-6}$. From Figure 7a we see that in this specific case the Weiszfeld algorithm converged in 20 iterations. From Figure 7b we see that the Steepest Descent algorithm converged after approximately 1100 iterations.

Figure 8 show the differences in convergence of the two algorithms when $\epsilon = 10^{-10}$. Figure 8a show that the Weiszfeld algorithm convergence in around 90 iterations. In contrast, we stopped the Steepest Descent algorithm after 100′000 iterations and we see from Figure 8b that not much happens after the first drop in the value of the stopping criterion.



(a) Convergence plot of the Weiszfeld algorithm. (b) Convergence plot of the Steepest Descent.

Figure 8: Plots of the convergence of the two algorithms for one realisation where $m = 10$ and $\epsilon = 10^{-10}$.

In Figure 9 the last 98′000 of the total 100′000 iterations are plotted and we see that nothing happens. This show that for this specific example we cannot attain a precision better than $1.36 \cdot 10^{-7}$. We consider this to be an additional drawback of the steepest descent algorithm, at least for the case when m is small and strengthens our advice on using the Weiszfeld algorithm in those cases.

To conclude, we advice using the Weiszfeld for $m \leq 200$. For larger values of m where a precise solution ($\epsilon < 10^{-6}$) is not important one can use the Steepest Descent algorithm, but we advice to

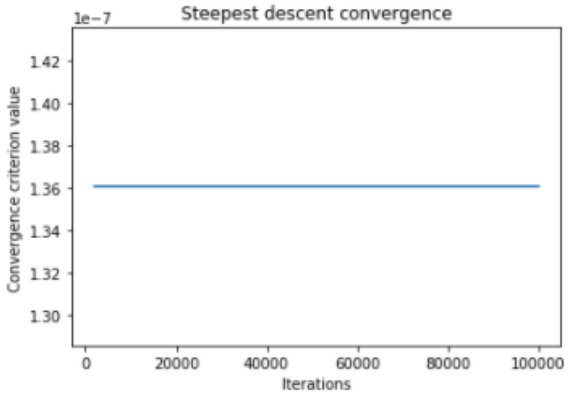


Figure 9: The last 98′000 iterations of the 100′000 iterations in the Steepest Descent algorithm when $\epsilon = 10^{-10}$.

investigate the performance of other algorithms as we believe that better results are attainable.

References

Köbis, Elisabeth (2021). *Project description*. URL: https://wiki.math.ntnu.no/_media/tma4180/2021v/projectdescriptionoptimization1.pdf.

Kreinovich, Vladik (1999). 'For Interval Computations, If Absolute-Accuracy Optimization is NP-Hard, Then So Is Relative-Accuracy Optimization'. In: URL: http://www.cs.utep.edu/vladik/1999/tr99-10.pdf.

Nocedal, Jorge and Stephen J. Wright (2006). *Numerical Optimization*. Springer Series in Operations Research. Springer Science+Business Media. ISBN: 9780387303031.