TMA4180 Optimization 1, project 1

August Arnstad, Aisha Halane and Ellen Skrimstad

March 2022

2.2

1)

A function is a metric if it fulfills the criteria:

1.
$$d(x,y)=0 \Rightarrow x=y$$

2.
$$d(x,y)=d(y,x)$$

3.
$$d(x,y)\leq d(x,z)+d(z,y)$$

Showing for d_1

1.
$$\begin{aligned} d_1(x,y)=x_1-y_1|+|x_2-y_2|=0 \\ \Rightarrow |x_1-y_1|=0, |x_2-y_2|=0 \\ \Rightarrow x=y \end{aligned}$$

2.
$$d_1(x,y)=|x_1-y_1|+|x_2-y_2|=|y_1-x_1|+|y_2-x_2|=d_1(y,x)$$

3.
$$\begin{aligned} d_1(x,y) &= \sum_i |x_i-y_i| \\ &= \sum_i |x_i-z_i+z_i-y_i| \\ &\leq \sum_i |x_i-z_i| + \sum_i |y_i-z_i| \\ &= d_1(x,z)+d_1(z,y) \end{aligned}$$

Showing for d_2

1.
$$\begin{gathered} d_2(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2}=0 \\ \Rightarrow (x_1-y_1)^2=0, (x_2-y_2)^2=0 \\ \Rightarrow x=y \end{gathered}$$

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

3.

$$\begin{aligned}
d_2(x,y)^2 &= \sum_i |x_i - y_i|^2 \\
&= \sum_i |x_i - z_i - y_i + z_i|^2 \\
&= \sum_i |x_i - z_i|^2 - 2\sum_i |x_i - z_i||x_i - y_i| + \sum_i |z_i - y_i|^2 \\
&\leq \sum_i |x_i - z_i|^2 + 2\sum_i |x_i - z_i||x_i - y_i| + \sum_i |z_i - y_i|^2 \\
&= d_2(x,z)^2 + 2d_2(x,z)d_2(x,y) + d_2(z,y)^2 \\
&= (d_2(x,z) + d_2(z,y))^2
\end{aligned}$$

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

Showing for d_∞

1.

$$d_\infty(x,y) = \max_i |x_i - y_i| = 0$$

$$\Rightarrow x - y = 0 \Rightarrow x = y$$

2.

$$d_\infty(x,y) = \max_i |x_i - y_i| = \max_i |y_i - x_i| = d_\infty(y,x)$$

3.

$$\begin{aligned}
d_\infty(x,y) &= \max_i |x_i - y_i| \\
&= \max_i |x_i - z_i + z_i - y_i| \\
&\leq \max_i |x_i - z_i| + |z_i - y_i| \\
&\leq \max_i |x_i - z_i| + \max_i |z_i - y_i| \\
&= d_\infty(x,z) + d_\infty(z,y)
\end{aligned}$$

2)

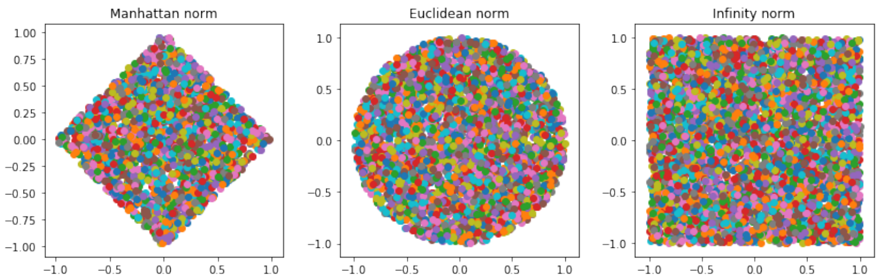

Figure 1: The unit ball in the Manhattan, Euclidean and maximum norm

3)

We recall the definition of a convex function:
A function $f : S \to \mathbb{R}$ is convex if its domain $S \subseteq \mathbb{R}^\Bbbk$ is a convex set and if for any two points $x, y \in S$ and $\lambda \in [0,1]$ the following inequality holds:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

We assume it to be known that $\mathbb{R}^\Bbbk$ is a convex set. We further proceed by checking the inequality. Let V be a vectorfield, then a norm is a function $f : V \to \mathbb{R}$.

It is known that every norm satisfies the triangle inequality

$$\|x + y\| \leq \|x\| + \|y\|$$

and the positive homogenity criteria, so we apply this to the inequality above:

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

which satisfies the inequality from the definition of a convex function. Hence every norm is a convex function. ■

4)

As we have established, every norm is convex. Further, if a norm $\|\cdot\| : \mathbb{R}^2 \to \mathbb{R}$ is a norm, then we can define a metric on $\mathbb{R}^2$ that is induced by the norm $\|\cdot\|$ as

$$d(x, y) := \|x - y\|$$

We shall take advantage of this, by proving that the objective functions can be represented as norms, making the convexity trivial.
Objective function 2.1): We make the assumption that $A \subseteq \mathbb{R}^2$. Further we have

$$f(x) = \max_{a \in A} d(a, x)$$

Recall the definition of p-norms on an n-dimensional (in our case $n = 2$) vector space:

$$\|x\|_p = \left(\sum_{i=1}^{n} x_i^{\frac{1}{p}}\right)^p = \left(\sum_{i=1}^{2} x_i^{\frac{1}{p}}\right)^p$$

for $1 \leq p < \infty$, and

$$\|x\|_\infty := \max_i |x_i|$$

We now notice that

$$f(x) = \max_{a \in A} d(a, x) = \max_{a \in A} \|a - x\|$$

Hence $f(x)$ must be convex, as the the convexity holds for all points $a \in A$ and $x \in \mathbb{R}^2$, including the points with the maximal distance, which by definition is our $f(x)$.
Objective function 2.2): With the same assumptions, we rewrite the objective function

$$f(x) = \sum_{a \in A} d(a, x) = \sum_{a \in A} \|a - x\|$$

and we procede to prove that the sum of norms satisfy the properties of a norm;

$$\|x\|_a + \|y\|_b \geq 0$$

$$|\alpha| \left(\|x\|_a + \|y\|_b\right) = |\alpha|\|x\|_a + |\alpha|\|y\|_b = \|\alpha x\|_a + \|\alpha y\|_b$$

$$\|x + z\|_a + \|y + z\|_b \leq \|x\|_a + \|z\|_a + \|y\|_b + \|z\|_b$$

where the triangle inequality holds immediately as both $\|\cdot\|_a, \|\cdot\|_b$ are norms by assumption.
We conclude that also this objective function can be written as a norm, and so it must be convex. ■

5

We are dealing with the Euclidean norm in a vectorspace $\mathbb{R}^2$. The vectorspace can be viewed as a surface or a screen. The objective is to choose the $x \in \mathbb{R}^2$ that minimizes the greatest distance between this x and the point $a^i \in A$ with the greatest Euclidean distance to x.
Geometrically speaking, this is the equivalent to enclosing all points $a^i \in A$ in a circle, with a radius $d_2(x, a)$ which is as small as possible, centered at x which is the minimizer.

6

A solution approach for $\min_{x\in\mathbb{R}^2}\sum_{a\in A} d(a,x)$ with Manhattan distance function is to separate the Manhattan distance. The Manhattan distance is written as $\sum_{a\in A}|a-x_1|+\sum_{a\in A}|a-x_2|$. If we find the minimum for each of these parts separable, the sum of these minimums will finally be the totally minimum as well. So if we first find $\min_{x\in\mathbb{R}^2}\sum_{a\in A}|a-x_1|$ and then $\min_{x\in\mathbb{R}^2}\sum_{a\in A}|a-x_2|$. Then we would have found $\min_{x\in\mathbb{R}^2}\sum_{a\in A} d(a,x)$ as well.

7)

We begin by recalling that the Euclidean norm is convex. Going forward, we can find from the definition of a convex function

$$f(\lambda x+(1-\lambda)y) \leq \lambda f(x)+(1-\lambda)f(y)f^2(\lambda x+(1-\lambda)y) \leq \lambda f^2(x)+(1-\lambda)f^2(y)+2\lambda(1-\lambda)f(x)f(y)$$

Now we manipulate this expression to obtain

$$f^2(\lambda x+(1-\lambda)y) \leq \lambda^2 f^2(x)+(1-\lambda)^2 f^2(y)+2\lambda(1-\lambda)f(x)f(y)+\big(-\lambda f^2(x)-(1-\lambda)f^2(y)+\lambda f^2(x)+(1-\lambda)f^2(y)$$

Combining the five first terms on the right hand side of this inequality yields

$$f^2(\lambda x+(1-\lambda)y) \leq -\lambda(1-\lambda)\left(f(x)-f(y)\right)^2+\lambda f^2(x)+(1-\lambda)f^2(y)$$

and we immediately see that the first term on the right hand side is strictly negative as $0\leq\lambda\leq 1$ This gives

$$f^2(\lambda x+(1-\lambda)y) \leq \lambda f^2(x)+(1-\lambda)f^2(y)$$

proving that the squared euclidean norm, is convex. To establish convexity of the objective function, we recall the property that if functions $f, g:\mathbb{R}^n\to\mathbb{R}$ are convex, so is the function $f+g$. Therefore the objective function, a sum of squared euclidean norms, is convex. ■
Note that the convexity could have been shown by proving that the Hessian of $f(x)$ is positive semi definite.
Since $f(x)$ is convex and differentiable w.r.t. $x\in\mathbb{R}^2$, it is know that an x^* satisfying $\nabla f(x^*)=0$ is a global minimizer of $f(x)$
Derive the gradient of $f(x)$:

$$\nabla f(x)=\left(\frac{\partial(\sum_{a\in A}(x_1-a_1)^2+(x_2-a_2)^2)}{\partial x_1},\frac{\partial(\sum_{a\in A}(x_1-a_1)^2+(x_2-a_2)^2)}{\partial x_2}\right)^T=\left(\sum_{a\in A}2(x_1-a_1),\sum_{a\in A}2(x_2-a_2)\right)^T$$

equate to 0:

$$\left(\sum_{a\in A}2(x_1-a_1),\sum_{a\in A}2(x_2-a_2)\right)^T=(0,0)^T \iff \sum_{a\in A}x_1-a_1=0,\sum_{a\in A}x_2-a_2=0$$

With a total of m points in the set A, we arrive at the global minimizer

$$(x_1,x_2)^T=\left(\frac{\sum_{a\in A}a_1}{m},\frac{\sum_{a\in A}a_2}{m}\right)^T \blacksquare$$

8)

The first order necessary condition states that if $x^*\in\mathbb{R}^2$ is a minimizer of the objective function f which is assumed to be continously differentiable in an open neighbourhood of x^*, then we must have $\nabla f(x^*)=0$-
To derive what this means for our case, obtain the gradient:

$$\nabla f(x)=\left(\frac{\partial(\sum_{a\in A}\sqrt{(x_1-a_1)^2+(x_2-a_2)^2})}{\partial x_1},\frac{\partial(\sum_{a\in A}\sqrt{(x_1-a_1)^2+(x_2-a_2)^2})}{\partial x_2}\right)^T$$

$$=\left(\sum_{a\in A}\frac{(x_1-a_1)}{d_2(x,a)},\sum_{a\in A}\frac{(x_2-a_2)}{d_2(x,a)}\right)^T$$

At this point, notice that if $d_2(x^*, a) = 0 \iff x^* = a$ then the gradient is not defined. If this is the case, our problem reduces to finding the point $x^* = a^i \in A$ which lies closest to all other points, i.e. we must minimize the euclidean norm from $x^* = a^i$ to all other points $a^j \in A/\{a^i\}$.
If $x^* \neq a \ \forall a \in A$, then our gradient is defined. The necessary condition is then

$$\nabla f(x) = \left(\sum_{a \in A} \frac{(x_1 - a_1)}{d_2(x, a)}, \sum_{a \in A} \frac{(x_2 - a_2)}{d_2(x, a)} \right)^T = 0$$

The second order necessary condition states that if $x^* \in \mathbb{R}^2$ is a minimizer of the objective function f, then if $\nabla^2 f(x^*)$ exists and is continouosly differentiable in an open neighbourhood of x^*, then $\nabla^2 f(x^*)$ is positive semi definite. We have shown that the Median problem with euclidean metric is convex. Thus we state without proof that a twice differentiable function f is convex if and only if $\nabla^2 f$ is positive definite. Hence, we have obtained that the necessary optimality conditions when the gradient and Hessian exist, and stated how to handle the problem when they do not. ■

9)

An example of a set where the global minimizer of $\min_{x \in \mathbb{R}^2} \sum_{a \in A} d(a, x)$ and $\min_{x \in \mathbb{R}^2} \sum_{a \in A} (d(a, x))^2$ do not coincide is an asymmetric set. The global minimizer of $\min_{x \in \mathbb{R}^2} \sum_{a \in A} d(a, x)$ will give the median point of the set, while the global minimizer of $\min_{x \in \mathbb{R}^2} \sum_{a \in A} (d(a, x))^2$ will give the average point of the set. A set that is asymmetric will not have the same median and average point. An example of such a set is $\{(1, 1), (1, 3), (1, 4)\}$, where the median will be $(1, 3)$, and the average will be $(1, 2.67)$.

3.2

1)

We know that the first orer necessary optimality condition is $\nabla f(x^*) = 0$, so a minimizer x^* must satisfy this. Further, we move forward under the assumption $x \notin \mathcal{M}$. We compute the gradient of our objective function and equate to zero:

$$\nabla f(x) = \left(\frac{v_1 2(x_1 - a_1^1)}{2d_2(x, a^1)} + ... + \frac{v_m 2(x_1 - a_1^m)}{2d_2(x, a^m)}, \frac{v_1 2(x_2 - a_2^1)}{2d_2(x, a^1)} + ... + \frac{v_m 2(x_2 - a_2^m)}{2d_2(x, a^m)} \right) = 0$$

We simply solve for (x_1, x_2) and obtain

$$\nabla f(x) = 0 \iff \sum_{i \in \mathcal{M}} v_i \frac{x_j}{d_2(x, a^i)} = \sum_{i \in \mathcal{M}} v_i \frac{a_j^i}{d_2(x, a^i)} \quad , \qquad j = 1, 2$$

which in yields the main iteration ofthe Weiszfeld scheme

$$x_j = \frac{\sum_{i \in \mathcal{M}} v_i \frac{a_j^i}{d_2(x, a^i)}}{\sum_{i \in \mathcal{M}} v_i \frac{1}{d_2(x, a^i)}} \quad , \qquad j = 1, 2$$

Now why is the iteration chosen like this? Firstly, it satisfies the first order necessary optimality condition. This combined with the convexity of the function, leads to a global minima x^* of f if and only if the gradient at x^* is 0.
We also note that this algorithm is very much like a gradient descent method, with a step size inversely proportional to the sum of weights divided by the euclidean distance between the current iterate x^k and the point a^i. By manipulating the iteration scheme we see it clearly:

$$\begin{aligned} x_j^{k+1} &= \frac{\sum_{i \in \mathcal{M}} v_i \frac{a_j^i + x_j^k - x_j^k}{d_2(x, a^i)}}{\sum_{i \in \mathcal{M}} v_i \frac{1}{d_2(x, a^i)}} = \frac{\sum_{i \in \mathcal{M}} v_i \frac{x_j^k}{d_2(x, a^i)}}{\sum_{i \in \mathcal{M}} v_i \frac{1}{d_2(x, a^i)}} + \frac{\sum_{i \in \mathcal{M}} v_i \frac{(a_j^i - x_j^k)}{d_2(x, a^i)}}{\sum_{i \in \mathcal{M}} v_i \frac{1}{d_2(x, a^i)}} \\ &= x_j^k - \underbrace{\frac{1}{\sum_{i \in \mathcal{M}} v_i \frac{1}{d_2(x, a^i)}}}_{\alpha_k} \frac{\partial f}{\partial x_j} \quad , \qquad j = 1, 2 \end{aligned}$$

Since a gradient descent method on a differentiable and convex function converges to a global minima, the scheme converges to a global minima, and that is why the iteration is chosen in such a way.

2)

Suppose $f(x)$ is the value of the objective function at the current iterate x and $f(x^*)$ is the objective function at the minimizer x^*. We assume $f(x)$ moves closer to $f(x^*)$ for each iteration. Furthermore, theorem 4 assumes that the expression $LB > 0$ and $\frac{UB}{LB} < \epsilon$. We wish to prove that if these assumptions hold, we have a relative error of atleast ϵ. Apply theorem 2 and 3 to the definition of the relative error

$$\left| \frac{f(x) - f(x^*)}{f(x^*)} \right| \underbrace{\leq}_{2\&3} \frac{\|\nabla f(x)\|\sigma(x)}{f(x) - \|\nabla f(x)\|\sigma(x)} = \frac{UB}{LB} < \epsilon$$

which is what we wanted to prove. ■

3)

To find a termination criterion we will look at the upper and the lower bound defined in theorem 2 and theorem 3. We will use our iteration values for x to find the termination criteria. We know that $f(x_{old}) - f(x_{new}) \leq$ upper bound. And that $f(x_{new}) \geq$ lower bound. We then know that $\frac{f(x_{old}) - f(x_{new})}{f(x_{new})} \leq \frac{UB}{LB}$. From theorem 4 we then know that if this is smaller the ϵ, then the current iterate at least has a relative accuracy1 of ϵ. We therefor set the termination criteria to be

$$\text{termination criteria} = \frac{f(x_{old}) - f(x_{new})}{f(x_{new})}.$$

4)

The implementation follows the following pseudocode

Algorithm 1 Weiszfeld algorithm

Check if minimum is attained in existing points of A:
if $\left[\left(\sum_{i \in \mathcal{M} \setminus k} v_i \frac{a_1^k - a_1^i}{d_2(x, a^i)} \right)^2 + \left(\sum_{i \in \mathcal{M} \setminus k} v_i \frac{a_2^k - a_2^i}{d_2(x, a^i)} \right)^2 \right]^{\frac{1}{2}} \leq v_k$ **then**
    $x^* = a^k$
    return x^*
Choose starting point x^0
x^0 is minimizer of Median problem with squared euclidean norm
while stop criteria$> \epsilon$ **do**
    Main iteration:
    for all $a \in A$ **do**

$$x_j = \frac{\sum_{i \in \mathcal{M}} v_i \frac{a_j^i}{d_2(x, a^i)}}{\sum_{i \in \mathcal{M}} v_i \frac{1}{d_2(x, a^i)}} \quad , \qquad j = 1, 2$$

return x^*

For testing an arbitrary set of points and weights can be used, the following has been used to produce our plots

i	Weight	a^i
1	4	(1,2)
2	10	(-1,3)
3	6	(4,4)
4	8	(3,1)
5	21	(-2,-3)
6	17	(5,-4)
7	9	(1,1)
8	4	(0,5)
9	11	(-2,5)
10	1	(-3,0)

The test result are shown in the plot below.

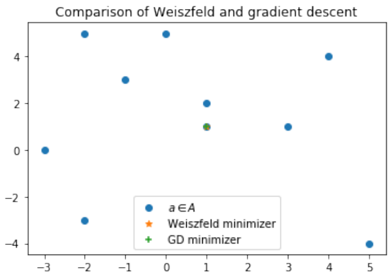

Figure 2: The points in A are plotted in blue, while the minimizer found with the Weiszfeld algortihm and gradient descent method coincide and are shown in orange and green.

5)

A comparison is done between the Weiszfeld algorithm and the Gradient Decent Algorithm using the same stopping criterion, $\epsilon = 10^{-6}$. The gradient descent method was implemented by replacing steps 2-4. A line search(backtracking) function was defined and instead of the main iteration, a step length value α in each iteration was found and updated the iterate by the scheme $x^{k+1} = x^k + \alpha_k p^k$ where $p^k = -\nabla f(x^k)$.
From the table it's seen that the gradient descent used less steps, and the time difference between the two methods is given in the table below.

Table 1: Comparison of methods

Measurement	Weiszfeld	Gradient descent
Iterations	12	8
Δt	0.027	0.13

As mentioned above, Weiszfeld is essentially a gradient descent method. The key difference is the step length, which in the Weiszfeld is chosen more specifically for the problem at hand. This choice of step is smart, as it is inversely proportional to the sum of weights divided by the distance between the iterate and each point. A possible problem with the Weiszfeld algorithm might appear if an iterate has the same location as an existing location, but we do not go further in on this as it has not been discussed in the project description.
The gradient descent method chooses its step by a much more general approach. This is sufficient for convergence but might lead to unnecessary steps that increase the runtime, as shown in the table above the runtime for this method was approximately 5 times longer (4.8) runtime than the Weiszfield algorithm or might not always choose the smartest step. Further the gradient descent method might depend heavily on the starting point, in this case the solution to the problem in squared euclidean norm, but it seems that this is a good initial guess for both our methods.
When testing the performance of the algorithms, we noticed that the Weiszfeld algorithm generally performs better as the number of locations increases. However, with a relatively small number of locations, both algorithms will eventually converge. We might thus conclude that our preferred method for this specific problem is the Weiszfeld algorithm, but both are efficient methods for solving the Fermat Weber problem.