Mathematics of Learning – Worksheet 1

Basics. [Solving linear equation systems.][1]
Solve the linear equation system $Ax = b$. A and b are given as

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 4 & 9 \\ 1 & 8 & 27 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 2 \\ 7 \end{pmatrix}.$$

Control yourself, if your solution is right. If you need some practice, generate some random linear equation systems and solve them.

Solution. We apply the gaussian elimination algorithm.
1. Subtract line 1 from lines 2 and 3, 2. Subtract 3 times line 2 from line 3, 3. divide through diagonal elements:

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 3 \\ 1 & 4 & 9 & 2 \\ 1 & 8 & 27 & 7 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 2 & 3 & 3 \\ 0 & 2 & 6 & -1 \\ 0 & 6 & 24 & 4 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 2 & 3 & 3 \\ 0 & 2 & 6 & -1 \\ 0 & 0 & 6 & 7 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 2 & 3 & 3 \\ 0 & 1 & 3 & -0.5 \\ 0 & 0 & 1 & \frac{7}{6} \end{array}\right)$$

4. Insert backwards.

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 3 \\ 0 & 1 & 3 & -0.5 \\ 0 & 0 & 1 & \frac{7}{6} \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 2 & 3 & 3 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & \frac{7}{6} \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 0 & 0 & 7.5 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & \frac{7}{6} \end{array}\right).$$

Basics. [Norms.]
A mapping $\|\cdot\|$ from any (real) vector space V to the non-negative real numbers $\mathbb{R}$ (bonus exercise: find a mapping on the real numbers, which has the three properties and takes some negative values, or show that it is impossible) is called a norm, whenever

$$\|v+w\| \leq \|v\| + \|w\|, \quad \|v\| = 0 \implies v = 0_V, \quad \|\lambda v\| = |\lambda|\|v\| \text{ for all } \lambda \in \mathbb{R}, v, w \in V.$$

Proof for the following statements if they are true or false.

1. Let $V = \mathbb{R}^n$ for some $n \in \mathbb{N}$. The euclidean norm

   $$\|v\|_2 := \sqrt{\sum_{i=1}^{n} v_i^2}$$

   is a norm.
   Solution. This is a norm, since we can check that the three properties hold.
   Property 1. Let $v, w \in \mathbb{R}^n$ be arbitrary vectors, and $\lambda \in \mathbb{R}$. Since both sides of property 1 are positive, it suffices to check if

   $$\|v+w\|^2 \leq (\|v\| + \|w\|)^2 = \|v\|^2 + \|w\|^2 + 2 \cdot \|v\|\|w\|$$

[1] There are lots of nice tutorial books for linear algebra and analysis available in our library. For a less formal introduction, you can, e.g., also consult wikipedia ;)

The left hand side is just

$$\sum_{i=1}^{n} v_i^2 + \sum_{i=1}^{n} w_i^2 + 2\sum_{i=1}^{n} v_i w_i = \|v\|^2 + \|w\|^2 + 2\sum_{i=1}^{n} v_i w_i,$$

the latter term being the scalar product of v and w (which induces the euclidean norm). Hence the inequality is equivalent to

$$\langle v, w\rangle \leq \|v\|\|w\|,$$

which is the well-known Cauchy-Schwarz Inequality, and thus holds true.
Property 2. We assume that $\|v\| = 0$. Then it holds, that

$$\|v\| = 0 \implies \sqrt{\sum_{i=1}^{n} v_i^2} = 0 \implies \sum_{i=1}^{n} v_i^2 = 0 \implies v_i^2 = 0 \quad \forall i \in [n] \implies v = 0.$$

Property 3. It holds that

$$\|\lambda v\| = \sqrt{\sum_{i=1}^{n} (\lambda v_i)^2} = \sqrt{\sum_{i=1}^{n} \lambda^2 v_i^2} = \sqrt{\lambda^2 \sum_{i=1}^{n} v_i^2} = |\lambda|\|v\|.$$

2. Let $V = \mathbb{R}^n$ for some $n \in \mathbb{N}$. The mapping

$$\|v\|_{\frac{1}{2}} := (\sum_{i=1}^{n} \sqrt{|v_i|})^2$$

is a norm.
Solution. This is not a norm, since property 1 does not hold. Consider $\mathbb{R}^2$ and unit vectors $e_1 = (1,0)^T$ and $e_2 = (0,1)^T$. It holds that $\|e_1\|_{\frac{1}{2}} = \|e_2\|_{\frac{1}{2}} = 1$ but on the other hand

$$\|e_1 + e_2\|_{\frac{1}{2}} = (\sqrt{1} + \sqrt{1})^2 = 4 > 2 = \|e_1\|_{\frac{1}{2}} + \|e_2\|_{\frac{1}{2}}.$$

3. Let V be the space of convergent sequences. The mapping

$$\|v\|_{lim} := \lim_{n\to\infty} v_n$$

is a norm.
Solution. This is not a norm, since (e.g.) the sequence $1, 0, 0, \ldots$ violates property 2: The sequence is not the 0-Sequence, but the limit is 0.

Exercise 1 [Python, Pandas, K-Means].
Install Python 3 on your computer and make sure you are able to import the following packages: `NumPy`, `Matplotlib`, `Pandas`. If you are new to Python you should first watch any Python introduction you find on your favorite video platform - or you look for written tutorials using your favorite search engine.

a) Download the dataset `faithful.csv` [2] from StudOn and load it into Python using the Pandas package.[3] Explore the dataset and visualize it as a two-dimensional plot using `Matplotlib`. Save the plot to a png file.

b) From plotting the data you should see two distinct clusters. Implement the K-means algorithm in Python (by completing the code `K-means_incomplete.py`) and test it (by running `python3 -i K-means.py` in a terminal). Apply K-means to `faithful.csv`.

Exercise 2 [Implementing EM for Clustering].
Implement the EM clustering algorithm for Gaussian mixtures as described on the slides. You can use the code `EM_incomplete.py`. Apply EM to `faithful.csv`.

Bonus [Experiments with K-Means and EM].
Generate own data sets. For example, take a few pictures of different objects (10 apples, 10 classrooms, 10 desks) with your smartphone camera (I propose to choose relatively low resolution), transform them to gray-scale matrices and apply the K-Means/EM Algorithm to the data set. Describe, visualize, and interpret your results.

Exercise 3 [Theory of K-means].
Letting $X \subset \mathbb{R}^M$ denote a finite set of N points, the i-th iteration of the K-means algorithm can be compactly written as ($\|\cdot\|$ is the euclidean norm)

$$\begin{cases} k_n^{(i)} \in \operatorname{argmin}_{k=1}^K \|x_n - {\color{red} m_k^{(i-1)}}\|, & \forall n = 1,\dots,N, \\ C_k^{(i)} := \{n \in \{1,\dots,N\} \,:\, k_n^{(i)} = k\}, & \forall k = 1,\dots,K, \\ m_k^{(i)} := \frac{1}{|C_k^{(i)}|} \sum_{x \in C_k^{(i)}} x, & \forall k = 1,\dots,K, \end{cases}$$

where the first line means that *exactly one* element in the argmin is selected.

- Show that the iterates of the algorithm satisfy

$$\frac{1}{2} \sum_{k=1}^K \sum_{x \in C_k^{(i)}} \|x - m_k^{(i)}\|^2 \le \frac{1}{2} \sum_{k=1}^K \sum_{x \in C_k^{(i-1)}} \|x - m_k^{(i-1)}\|^2.$$

Solution. We give the proof in two steps; first we show that

$$\sum_{k=1}^K \sum_{x \in C_k^{(i)}} \|x - m_k^{(i-1)}\|^2 \le \sum_{k=1}^K \sum_{x \in C_k^{(i-1)}} \|x - m_k^{(i-1)}\|^2.$$

[2]See `https://www.stat.cmu.edu/~larry/all-of-statistics/=data/faithful.dat`

[3]You can learn how to use Pandas here: `https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html`.

We see this easily by rearranging the sum a little bit - which works since we have at both sides exactly N terms:

$$\sum_{k=1}^{K} \sum_{x \in C_k^{(i)}} \|x - m_k^{(i-1)}\|^2 = \sum_{n=1}^{N} \|x_n - m_{k_n^{(i)}}^{(i-1)}\|^2 \leq$$

$$\leq \sum_{n=1}^{N} \|x_n - m_{k_n^{(i-1)}}^{(i-1)}\|^2 = \sum_{k=1}^{K} \sum_{x \in C_k^{(i-1)}} \|x - m_k^{(i-1)}\|^2,$$

this works because $k_n^{(i)}$ has been set to minimize the terms $\|x_n - m_k^{(i-1)}\|$.
Next we show that

$$\sum_{x \in C_k^{(i)}} \|x - m_k^{(i)}\|^2 \leq \sum_{x \in C_k^{(i)}} \|x - m_k^{(i-1)}\|^2$$

for all $k \in [K]$. To see this, we consider the function

$$f_k^{(i)}(m) := \sum_{x \in C_k^{(i)}} \|x - m\|^2$$

A necessary optimality condition of this function would be the gradient being 0, hence, $\forall d = 1, \ldots, D$,

$$\frac{\partial}{\partial m_d} \sum_{x \in C_k^{(i)}} \sum_{j=1}^{D} (x_j - m_j)^2 = \sum_{x \in C_k^{(i)}} \sum_{j=1}^{D} \frac{\partial}{\partial m_d} (x_j - m_j)^2 = \sum_{x \in C_k^{(i)}} 2 \cdot (x_d - m_d) \cdot (-1) = 0.$$

This is equivalent to $|C_k^{(i)}| m = \sum_{x \in C_k^{(i)}} x$. As the hessian matrix $H^{(i)_k}$ of the function is defined as

$$H_{k,dj}^{(i)} := \frac{\partial}{\partial m_d \partial m_j} f_k^{(i)}(m) = \frac{\partial}{\partial m_j} \sum_{x \in C_k^{(i)}} -2(x_d - m_d) = \begin{cases} 0 \text{ for } d \neq j \\ 2 \cdot |C_k^{(i)}| \text{ otherwise} \end{cases}$$

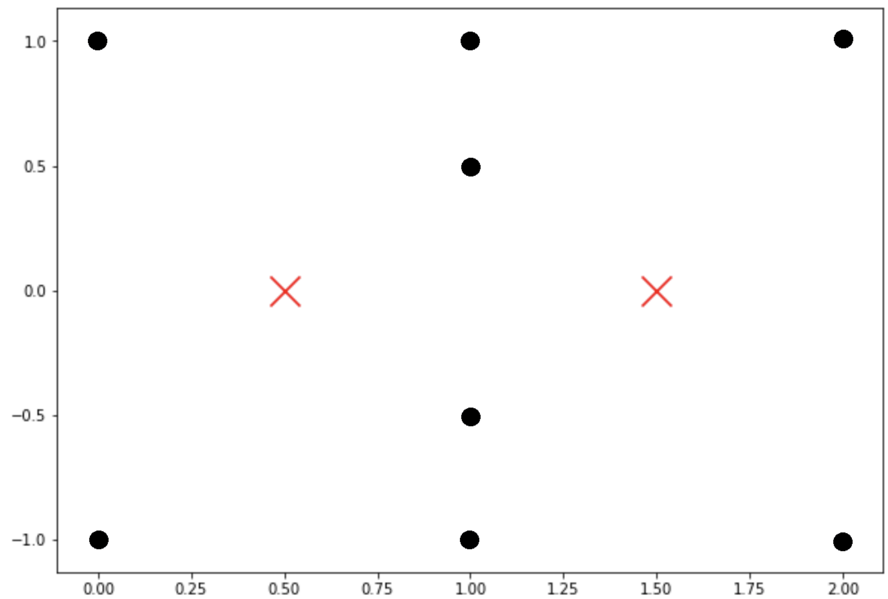
which is a positive matrix, the critical point is a minimum. Hence, $m_k^{(i)}$ is a minimizer of $f_k^{(i)}$ and the second inequality follows. Taking all together, we get the desired result.

- Why is it important for this that every data point x_n is assigned to precisely one class?
  Solution. Basically the answer here is a discussion. One possible reasoning would be that otherwise you would double-count some distances.

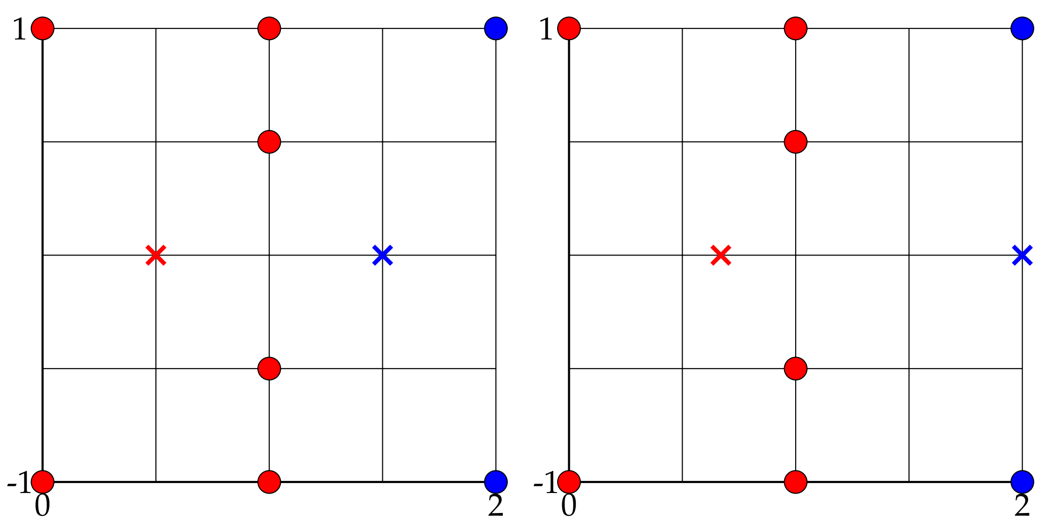
- Try to extend the result to an arbitrary norm $\| \cdot \|$.
  Solution. It does not work, mostly because the arithmetic mean and other norms do not fit together, roughly speaking. Consider the maximum norm (which assigns a vector to its largest component in terms of absolute values), $K = 1$ and $N = 3$, $D = 1$, points $x_1 = 0, x_2 = 1, x_3 = 1000, m_0 = 500$, which is obviously the optimal cluster center. The algorithm just skips step 1 (since we only have

one cluster, so re-assigning points to clusters is non-sense), and recalculates the cluster center as $\frac{1}{3}(0+1+1000)$ which is worse than 500, measured in maximum distance from all cluster points. Nevertheless, it is possible to adapt the update rule in step 3 that it fits for the maximum norm (bonus exercise) or for some arbitrary norms you would try here (also bonus exercise).

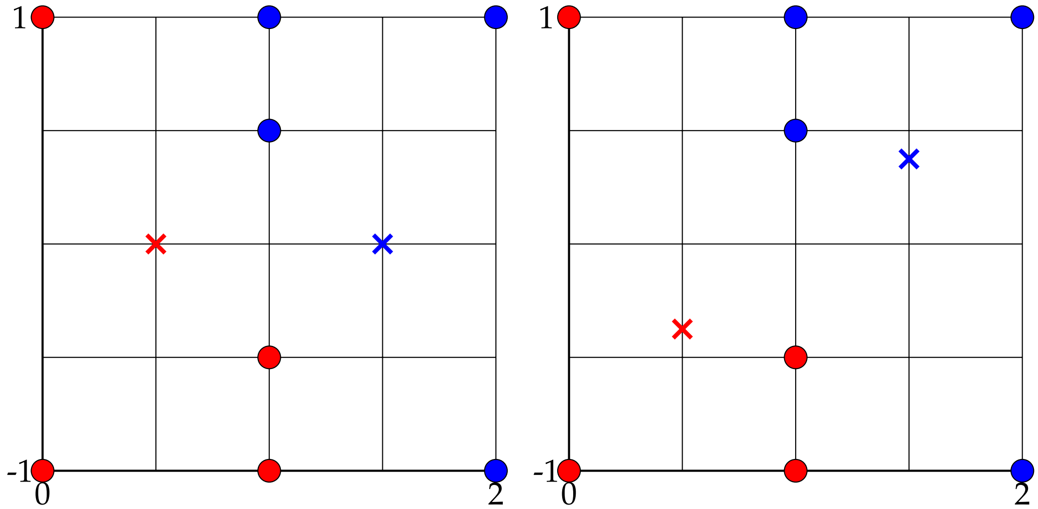
- Construct explicit solutions of K-means in the following situation, where the two red crosses correspond to the initialization $m_k^{(0)}$ of the means. How does this depend on the choice of assignment in the first line of K-means?



Solution. A possible way of the algorithm...



A 2nd trajectory...



Acknowledgement, for generating a previous version of this sheet, to Leon Bungert and Daniel Tenbrinck.