LAB 1 : Entropic regularization for optimal transport

- EXERCISE 1: CODING SINKHORN -

Let $\alpha = \sum_{i=1}^n a_i \delta_{\mathbf{x}_i}, \beta = \sum_{j=1}^m b_j \delta_{\mathbf{y}_j}$ be two discrete probability measures. Let $\mathbf{C} = (c(\mathbf{x}_i, \mathbf{y}_j))_{ij}$ be the cost matrix between the samples. Consider the entropic regularized OT problem

$$\mathrm{OT}_\varepsilon(\alpha, \beta) = \min_{\mathbf{P} \in U(\mathbf{a},\mathbf{b})} \langle \mathbf{P}, \mathbf{C} \rangle \ - \varepsilon H(\mathbf{P}) \,, \tag{1}$$

where $H(\mathbf{P}) = -\sum_{ij} P_{ij}(\log(P_{ij}) - 1)$ is the negative entropy.

Q1. Code the Sinkhorn algorithm for solving (1). The function should take as input $\mathbf{a}, \mathbf{b}, \mathbf{C}$ a regularization parameter ε, a number of maximum iterations and a convergence criterion (below which the algorithm stops). It should output the optimal transport plan.

Q2. Generate random Gaussian samples in dimension $d = 2$: $\mathbf{x}_1, \cdots, \mathbf{x}_n \sim \mathcal{N}(0, \sigma^2 I)$ and $\mathbf{y}_1, \cdots, \mathbf{y}_m \sim \mathcal{N}(0.5, \sigma^2 I)$ with $\sigma = 0.05$, consider uniform weights $\mathbf{a} = \frac{1}{n}\mathbf{1}_n, \mathbf{b} = \frac{1}{m}\mathbf{1}_m$ and $c(\mathbf{x}_i, \mathbf{y}_j) = \|\mathbf{x}_i - \mathbf{y}\|_2^2$. Compute the OT plan using the previous algorithm with varying $\varepsilon \in \{0.001, 0.01, 0.1, 1\}$ and the dual loss during the iteration (for only one ε). Vizualize the plan (for different ε) using the code below and compare with the unregularized problem.

```python
def plot_plan(X, Y, P, ax, thresh=1e-6, scale=1.0):
    # P is the OT plan
    n, m = P.shape[0], P.shape[1]
    Pmax = P.max()
    for i in range(n):
        for j in range(m):
            if P[i, j] / Pmax > thresh:
                ax.plot([X[i, 0], Y[j, 0]],
                        [X[i, 1], Y[j, 1]],
                        color='black',
                        alpha=(P[i, j] / Pmax)*scale)
```

Q3. Now compute the OT plan with $\varepsilon = 1e - 5$. What happens ? What can you suggest to overcome this problem ?

We define the *logsumexp* of a vector $\mathbf{z}$ as $\mathrm{LSE}(\mathbf{z}) = \log(\sum_i \exp(z_i))$ and the *softmin* at level $\varepsilon > 0$ as $\mathrm{softmin}_\varepsilon \mathbf{z} = -\varepsilon \,\mathrm{LSE}(-\mathbf{z}/\varepsilon)$.

Q4. Show that for any $\alpha \in \mathbb{R}$ we have $\mathrm{LSE}(\mathbf{z} + \alpha \mathbf{1}) = \alpha + \mathrm{LSE}(\mathbf{z})$. Deduce that $\mathrm{softmin}_\varepsilon \mathbf{z} = \alpha + \mathrm{softmin}_\varepsilon \{\mathbf{z} - \alpha \mathbf{1}\}$.

Q5. What is the value of $\mathrm{softmin}_\varepsilon \mathbf{z}$ as ε goes to 0? Prove it. What is the problem when computing $\mathrm{softmin}_\varepsilon \mathbf{z}$ when ε is small ?

Q6. Based on the previous answers code a stable implementation of softmin.

Q7. Show that the iterations of Sinkhorn's algorithm are equivalent to choosing $\mathbf{g}^{(0)} \in \mathbb{R}^m$ and then iterating for $k \geq 1$

$$\begin{aligned}
&\forall i \in [\![n]\!],\ f_i^{(k)} \leftarrow \varepsilon \log(a_i) + \mathrm{softmin}_\varepsilon \left(C_{ij} - g_j^{(k-1)}\right)_j \,, \\
&\forall j \in [\![m]\!],\ g_j^{(k)} \leftarrow \varepsilon \log(b_j) + \mathrm{softmin}_\varepsilon \left(C_{ij} - f_i^{(k)}\right)_i \\
&k \leftarrow k + 1 \,.
\end{aligned} \tag{2}$$

These iterations are called the log-domain Sinkhorn. Implement these iterations and show on the previous example that it is more stable. What is the main drawback ?

Q8. Compare the previous implementations with the result of the `POT` library (you can use `ot.sinkhorn`).

Q9. Generate data according to the following code and plot the corresponding points.

```
from sklearn.datasets import make_blobs
import numpy as np
centers = np.array([[0, 0],
                    [1, 1],
                    [1, 0]])
n_samples = np.array([50, 100, 50])
X, y = make_blobs(n_samples=n_samples, centers=centers, shuffle=False,
cluster_std=0.1)
```

Q10. The previous dataset corresponds to one distribution α. Compute and plot the optimal transport plan corresponding to $\mathrm{OT}_\varepsilon(\alpha, \alpha)$ for varying ε. How do you interpret the result ?