ENS Lyon
2024/2025
Mathurin Massias, Titouan Vayer, Quentin Bertrand

Lab 2 : Domain adaptation with optimal transport

In this practical session we will apply on digit classification the OT based domain adaptation method described in the course. This practical session is based on the code provided by Rémi Flamary and Nicolas Courty. You need to import the following modules

```
# standard, frequent imports:
import ot
import numpy as np
import matplotlib.pyplot as plt # do the plots
from sklearn.svm import SVC
from sklearn.preprocessing import OneHotEncoder

# import for pretty visualization
from matplotlib.colors import LogNorm, Normalize
from mpl_toolkits.axes_grid1 import make_axes_locatable
```

- Exercise 1: OTDA on digits -

We will consider a digits classification task and two datasets: the MNIST dataset and the USPS dataset. First you need to download the data. Go here https://github.com/rflamary/OTML_DS3_2018/blob/master/data/mnist_usps.npz and download the file.

```
# Load the data
data = np.load(path_to_the_data+'/mnist_usps.npz')
xs, ys = data['xs'], data['ys']
xt, yt = data['xt'], data['yt']
# normalization
xs = xs/xs.sum(1, keepdims=True)
xt = xt/xt.sum(1, keepdims=True)
```

Q1. Do a quick inspection of the data: how many samples in each distribution, how many classes, the proportions of each class in each distribution.

We have a source distribution μ_s that corresponds to the MNIST dataset. Our goal is to train a classifier on MNIST and to apply this classifier on the USPS dataset, that corresponds to a target distribution μ_t. We are in an artificial setting where we have access to the labels in the target domain, **which is not the case in real life**. We will first inspect if we need to do domain adaptation between the two distributions.

Q2. Display the images in each distribution using the code

```
# function for plotting images
def plot_image(x):
    plt.imshow(x.reshape((28, 28)), cmap='gray')
    plt.xticks(())
    plt.yticks(())

nb = 10
for x, y, name in zip([xs, xt], [ys, yt], ["MNIST", "USPS"]):
    plt.figure(figsize=(nb, nb))
    for i in range(nb*nb):
        plt.subplot(nb, nb, 1+i)
```

```
        c = i % nb
        plot_image(x[np.where(y == c)[0][i//nb], :])
    plt.suptitle(name, fontsize=20)
    plt.subplots_adjust(top=0.95)
```

Q3. What can we say about the different distributions?

Q4. Using `sklearn.model_selection.train_test_split`, do a train-test split of the MNIST data. Then, train a SVM classifier with regularization parameter $C = 1$, Gaussian kernel parameter $\gamma = 100$ on the MNIST training data. What is the classification accuracy of this model on the MNIST test data? Compare to its accuracy on the USPS data. What do you observe?

Q5. Compute the cost matrix $\mathbf{C} = (c(\mathbf{x}_i^s, \mathbf{x}_j^t))_{ij}$ and display it using the following snippet. What can we say ?

```
fs = 12
imshow_kwargs = {'cmap': 'Blues', 'norm': Normalize(vmin=vmin, vmax=vmax)}
fig, ax = plt.subplots(1, 1)
im = plt.imshow(C, **imshow_kwargs)
ax.set_title('Cost matrix', fontsize=fs)
divider = make_axes_locatable(ax)
cax = divider.append_axes("right", size="3%", pad=0.05)
cb = fig.colorbar(im, cax=cax, orientation='vertical')
cb.ax.tick_params(labelsize=fs-2)
```

Q6. Code a function that implements the barycentric mapping from the source distribution to the target distribution: it should take as input the target data, and the optimal transport plan and return the mapped samples.

Q7. Compute the standard OT between the source and the target distributions (using uniform weights and `ot.emd`) and compute the barycentric projection $\hat{\mathbf{X}}_s$ of the source data.

Q8. Re-train the same classifier but using $\hat{\mathbf{X}}_s$. What are the different performances (on USPS and MNIST) in this case ?

Q9. We will compare this procedure with a simple label propagation procedure. Code a function that implements label propagation: it should take as input the source labels, the optimal transport plan, and return the estimated target labels (you might use `OneHotEncoder`).

Q10. Apply label propagation using the OT plan found before. Plot the confusion matrix between the estimated target labels and the true target labels. Which of the two methods is the best ?

Q11. Use now entropic optimal transport and compare the results with the case of no regularization. You should obtain something as in Figure 1.

Q12. How should we properly tune the hyperparameters ?

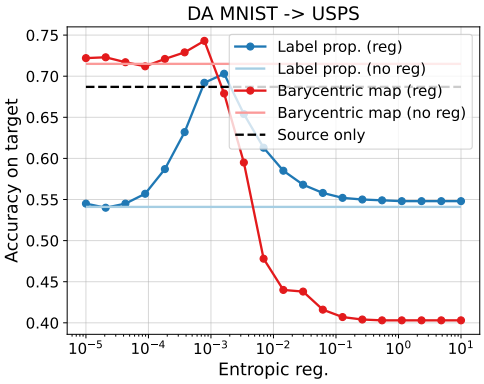

Figure 1: Domain adaptation results w.r.t. regularization parameter.