Assignment 4

Convex Optimization SS 2018

Patrick Knöbelreiter, knoebelreiter@icg.tugraz.at

19.06.2018

Submission: Upload your solution as a PDF and your implementation as PY files to the TeachCenter. You can also use your scanned hand-written manuscript.

Deadline: July 9$^{\text{th}}$, 2018 at 23:55h.

1 Theoretical Part

1.1 Optimal Transport

Consider the problem of finding an optimal transport plan $\mathbf{\Pi} \in \mathbb{R}_+^{N\times N}$ from a source set $\mathcal{X} = \{X_i\}_{i=1}^N$ to a target set $\mathcal{Y} = \{Y_i\}_{i=1}^N$. For example the i-th row of the transport plan, Π_i, contains for all j the probability that X_i maps to Y_j. To find the best assignment we want to minimize

$$\min_{\mathbf{\Pi}\in\mathbb{R}_+^{N\times N}} \quad \langle \mathbf{C}, \mathbf{\Pi}\rangle_F \quad \text{s.t.} \textstyle\sum_i \Pi_{ij} = p_j, \quad \sum_j \Pi_{ij} = q_i, \tag{1}$$

which is the Kantorovich relaxation of the optimal transport problem. In (1) $\mathbf{C} \in \mathbb{R}_+^{N\times N}$ is a cost matrix and $\mathbf{p}, \mathbf{q} \in \mathbb{R}^N$ are the marginal probability distributions of $\mathbf{\Pi}$, where $\sum_i p_i = 1$ and $\sum_i q_i = 1$. Notice that $\langle ., .\rangle_F$ is the Frobenius dot product which is defined as

$$\langle \mathbf{C}, \mathbf{\Pi}\rangle_F = \sum_{i=1}^N \sum_{j=1}^N C_{ij}\Pi_{ij}.$$

The solution of (1) yields the bijective assignment

$$\mathbf{\Pi}: \qquad \mathcal{X} \mapsto \mathcal{Y} \tag{2}$$

$$X_i \to \sum_{j=1}^N \Pi_{ij} Y_j, \tag{3}$$

i.e. each source point maps to exactly one target point and vice versa.

1.2 Primal-Dual Algorithm

Using the primal-dual algorithm we can solve problems of the form

$$\min_{\mathbf{x}} f(\mathbf{K}\mathbf{x}) + g(\mathbf{x}), \tag{4}$$

where f, g are convex l.s.c. and 'simple' functions, and $\mathbf{K} \in \mathbb{R}^{m\times n}$ is a bounded linear operator. We can rewrite the primal problem (4) to the saddle-point problem

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \langle \mathbf{K}\mathbf{x}, \mathbf{y}\rangle - f^*(\mathbf{y}) + g(\mathbf{x}). \tag{5}$$

Problems of the form (5) can be solved with the PDHG algorithm

$$\begin{cases} x^{k+1} = \text{prox}_{\tau g}(x^k - \tau K^T y^k) \\ y^{k+1} = \text{prox}_{\sigma f^*}(y^k + \sigma K(2x^{k+1} - x^k)) \end{cases} \tag{6}$$

for an initial pair of primal and dual points (x^0, y^0) and steps $\tau, \sigma > 0$.

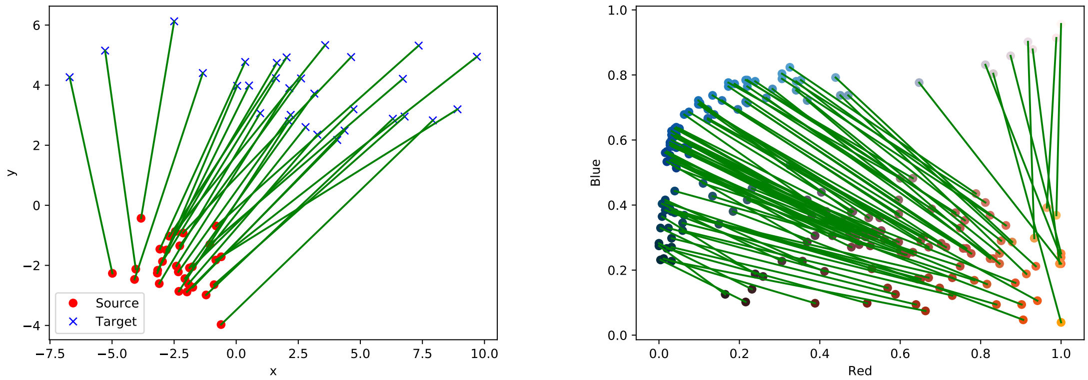

Figure 1: Matching datapoints with optimal transport. Left (Dataset matching): Each datapoint from the source dataset is 'transported' to exactly one datapoint of the target dataset. Right: Color transfer.

Tasks

1. Rewrite the optimization problem given in (1) to a saddle point problem of the form (5).
2. What are $\mathbf{K}, \mathbf{x}$ and $\mathbf{y}$ in the optimal transport setting?
3. Compute the Lipschitz constant of (5) in the optimal transport setting analytically.
4. Compute the proximal maps arising in (6) in the optimal transport setting analytically.

1.3 Sinkhorn-Knopp Algorithm

The Sinkhorn Knopp algorithm arises from an entropic regularized optimal transport problem:

$$\min_{\mathbf{\Pi} \in \mathbb{R}_+^{N \times N}} \langle \mathbf{C}, \mathbf{\Pi} \rangle_F + \varepsilon \sum_{ij} \Pi_{ij} \log \Pi_{ij} \quad \text{s.t.} \sum_i \Pi_{ij} = p_j, \quad \sum_j \Pi_{ij} = q_i. \tag{7}$$

The Sinkhorn-Knopp iterates are given by

$$\begin{cases} \mathbf{u}^{k+1} = \frac{\mathbf{p}}{\mathbf{M}\mathbf{v}^k} \\ \mathbf{v}^{k+1} = \frac{\mathbf{q}}{\mathbf{M}^T\mathbf{u}^{k+1}}, \end{cases} \tag{8}$$

with $\mathbf{v}^0 = \mathbf{1}$, i.e. the one vector with size N and $\mathbf{M} = \exp(\frac{-\mathbf{C}}{\varepsilon})$. Note that the division above is to be understood element-wise. The transport plan $\mathbf{\Pi}$ can be computed for each iteration with

$$\mathbf{\Pi}^k = \operatorname{diag}(\mathbf{u}^k)\, \mathbf{M}\, \operatorname{diag}(\mathbf{v}^k). \tag{9}$$

Having a solution of (7) we can still use (3) to compute the transport. However, due to the entropic regularization we do not have a bijective mapping anymore, but instead it is also possible that a new point is a combination of several source points. Here we additionally have the possibility to compute the winner-takes-it-all solution by using the mapping

$$\mathbf{\Pi}: \qquad \mathcal{X} \mapsto \mathcal{Y} \tag{10}$$

$$X_i \to Y_{j^*}, \quad j^* \in \arg\max_j \Pi_{ij}, \tag{11}$$

i.e. we use the match with the highest probability.

2 Practical Part

In the practical part we are going to use the theoretical results from section 1 and apply them to dataset matching and color transfer. For both examples we choose the distribution $\mathbf{p}$ and $\mathbf{q}$ to be a uniform distribution, i.e. $\forall i: \; p_i = \frac{1}{N}$ and $\forall i: \; q_i = \frac{1}{N}$. We set the cost between two samples X_i and Y_j to be the squared Euclidean cost, i.e. $C_{ij} = ||X_i - Y_j||^2$.

Hint For the implementation it is advantageous to make two matrices $\mathbf{X}$ and $\mathbf{Y}$ for the datasets $\mathcal{X}$ and $\mathcal{Y}$, where each sample X_i and Y_i corresponds to a row in $\mathbf{X}$ and $\mathbf{Y}$, respectively.

Figure 2: Color Transfer with OT. Top left: The image we want to recolor with the colors from the top right image. Bottom left: Result with the PDHG algorithm of problem (1). Bottom right: Result of the Sinkhorn-Knopp algorithm of problem (7).

2.1 Dataset Matching

Dataset matching fits perfectly the definition of an optimal transport problem: For each point in a source dataset we want to find the best datapoint in a target dataset while minimizing a cost. The left plot in fig. 1 shows an example of dataset matching.

Tasks

1. Generate a suitable source and target dataset with $N = 50$ 2D samples and plot them.
2. Compute the cost matrix $\mathbf{C}$.
3. Implement the PDHG algorithm (6) for this optimal transport problem. Use the Lipschitz constant computed in a previous task to choose σ and τ appropriately.
4. Implement the Sinkhorn-Knopp algorithm (8).
5. Perform the following steps for both algorithms:
   a) Run the algorithm until convergence. For the Sinkhorn-Knopp algorithm additionally vary ε.
   b) Plot the energy over the iterations.
   c) Plot the optimized optimal transport map $\mathbf{\Pi}$.
   d) Plot the datasets and the optimal assignments similar to Fig. 1. For the Sinkhorn-Knopp algorithm additionally compare (3) with (11).

2.2 Color Transfer

Color transfer is the task of transferring the colors of one image to another image as shown in Fig. 2. We can use exactly the same algorithms as for the dataset matching problem to achieve this. The only difference is that each point is now a 3D vector, because we are dealing with RGB color images.

Tasks

1. Select two suitable images of your choice.
2. Select $N = 1000$ colors pixels from the source image and 1000 color pixels from the target image.
3. Perform the tasks 2 and 5 from the dataset matching task using your source and target colors. Note: You can plot the assignments using a scatter plot similar to `plt.scatter(Xs[:,0], Xs[:,1], c=Xs)`.
4. Transfer the colors using your computed OT map. Hint: Use the KD-Tree from scipy to perform a nearest neighbor search (`scipy.spatial.KDTree`) of the remaining pixels.