Project: Optimal Transport and Entropy in Data Science

Convex Optimisation in Mathematical Finance

Andreas Søjmark, TA: Benedikt Petko

Friday 12th Feb, 2021 (v3 — 22nd Feb)

1 Introduction

Let $\mathcal{S}$ be the set of possible outcomes of some data collection process, $\mathcal{T} = \{1, \ldots, k\}$ the set of labels and π a probability distribution on $\mathcal{S} \times \mathcal{T}$. The prototypical problem in classification is to find a function

$$f : \mathcal{S} \to \mathcal{T}$$

such that, given a loss function $\mathcal{L} : \mathcal{T} \times \mathcal{T} \to [0, \infty)$, the expected loss

$$L(X, Y) := \mathbb{E}[\mathcal{L}(f(X), Y)]$$

is minimized for $(X, Y) \sim \pi$. In practice, to estimate the expected loss, a dataset

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{S} \times \mathcal{T}$$

is sampled from the distribution π. As a surrogate for the expected loss $L(X, Y)$, the aim is then to minimize the empirical loss

$$\hat{L}(\mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i)$$

Solving this problem usually requires giving the outcome space $\mathcal{S}$ and the data $\mathcal{D}$ some structure by embedding them in an ambient metric or Euclidean space. One can then use a model, such as a support vector machine (SVM) or artificial neural networks, and finally apply an optimization algorithm to find the minimum.

In this project, we are interested in the case when $\mathcal{S}$ is the space of one- or two-dimensional histograms of probability distributions. This corresponds to the problem of matching signals or images to their labels. For example, the MNIST dataset [LC10] contains images of handwritten digits and corresponding ground truth labels $\{0, \ldots, 9\}$ and has become the standard in the machine learning literature to benchmark the performance of newly proposed supervised learning algorithms.

We will focus on the problem of equipping the space of histograms with a suitable metric. In particular, we will first equip the space of histograms with a Wasserstein metric. Computing the Wasserstein metric turns out to be computationally infeasible; however, a type of regularization will yield the so-called (dual) Sinkhorn metric which is a solution to a convex optimization problem and is computable with linear complexity. Finally, we will compute the solution of the regularized problem by the Sinkhorn-Knopp algorithm.

The optimal transport approach has been a successful and innovative method in various other machine learning tasks, e.g. domain adaptation [Cou+17], natural language processing [Kus+15] and dimensionality reduction [Tol+18].

2 Optimal transport and Wasserstein distances

We will follow closely the exposition in [Cut13]. For simplicity, we only consider one-dimensional probability histograms of length d. For more details on computational optimal transport, see [PC19].

Definition 1 (Probability simplex and transport plans)**.** Denote the probability simplex in $\mathbb{R}^d$ as

$$\Sigma_d := \{x \in \mathbb{R}^d_+ : x^T 1_d = 1\}$$

and define the space of transportation plans between $r \in \Sigma_d$ and $c \in \Sigma_d$ as

$$U(r,c) := \{P \in \mathbb{R}^{d\times d}_+ : P1_d = r, P^T 1_d = c\}$$

for histograms $r, c \in \Sigma_d$.

Remark 2. Note that $U(r,c)$ is simply the set of stochastic matrices with marginals r and c. For any transportation plan $P \in U(r,c)$, each entry P_{ij} can be interpreted as the proportion of r_i to be transported to the location of c_j.

Definition 3 (Optimal transport problem)**.** Given probability distributions $r, c \in \Sigma_d$ and a matrix $M \in \mathbb{R}^{d\times d}_+$, the optimal transport problem is

$$d_M(r,c) := \min_{P\in U(r,c)} \langle P, M\rangle := \min_{P\in U(r,c)} \sum_{i,j=1}^{d} P_{ij} M_{ij} \tag{1}$$

Remark 4. Note that the expression to be minimized over is linear in P and we are minimizing under the implicit (linear) constraints $P1_d = r, P^T 1_d = c$. Such an optimization problem is known as a linear program.

Remark 5. M is commonly referred to as the *cost matrix.* Each entry M_{ij} should be interpreted as the cost per unit of transporting from site r_i to site c_j. A cost matrix may arise from pairwise distances of points in a metric space (e.g. a data set randomly sampled from a metric space) or it may be a fixed object, as in the case for image pixel intensities that we consider in Section 6, where the cost matrix consists of the fixed pairwise pixel distances.

Definition 6 (Metric matrices)**.** $M \in \mathbb{R}^{d \times d}$ is a metric matrix if it is symmetric, $\forall i \leqslant d : M_{ii} = 0$ and $\forall i, j, k \leqslant d : M_{ij} \leqslant M_{ik} + M_{kj}$.

Theorem 7. *If M is a metric matrix then the function $d_M : \Sigma_d \times \Sigma_d \to \mathbb{R}$ defined in (1) is a metric on Σ_d.*

Definition 8 (1-Wasserstein distance on Σ_d)**.** Let $M \in \mathbb{R}_{+}^{d \times d}$ be a metric cost matrix. The metric d_M on Σ_d as defined by (1) is called the **1-Wasserstein distance** and is also commonly referred to as the **earth mover's distance** (EMD).

Remark 9. In the case where Σ_d arises from uniformly spaced histograms (e. g. pixel intensities), the cost matrix will stay fixed with respect to sampling and will correspond to pixel distances in the context of image processing.

The worst case computational cost of computing the minimum in the linear program (1) is known to be $O(d^3 \log d)$ which is prohibitive for large scale applications. However, recent developments in computational optimal transport have reduced the computational complexity of optimal transport distances to $O(d)$ by regularization, thus bringing optimal transport back to relevance in machine learning. This is the content of the next section.

3 Basic information theory

Let us recall some definitions from information theory. For more details, see Chapter 2 of Cover and Thomas [CT06].

Definition 10 (Entropy)**.** For a histogram $r \in \Sigma_d$ define the entropy as

$$h(r) := -\sum_{i=1}^{d} r_i \log r_i$$

Similarly, for a transportation plan $P \in U(r, c)$ define the entropy as

$$h(P) := -\sum_{i,j=1}^{d} P_{ij} \log P_{ij}$$

Definition 11 (Divergence)**.** A function $D(\cdot \parallel \cdot) : \Sigma_d \times \Sigma_d \to \mathbb{R}^*$ is said to be a divergence if

$$\begin{aligned} &\forall p, q \in \Sigma_d : D(p \parallel q) \geqslant 0 \\ &\forall p, q \in \Sigma_d : D(p \parallel q) = 0 \iff p = q \end{aligned}$$

Definition 12 (Kullback-Leibler divergence)**.** For $p, q \in \Sigma_d$ define the Kullback-Leibler divergence as

$$\mathrm{KL}(p \parallel q) = \begin{cases} \sum_i p_i \log \frac{p_i}{q_i} & p, q \text{ equivalent} \\ +\infty & \text{otherwise} \end{cases}$$

Similarly, for $P, Q \in U(r,c)$ define

$$\mathrm{KL}(P \parallel Q) := \begin{cases} \sum_{i,j} P_{ij} \log \frac{P_{ij}}{Q_{ij}} & P, Q \text{ equivalent} \\ +\infty & \text{otherwise} \end{cases}$$

Remark 13. In general, a divergence is not necessarily a metric.

4 Entropic regularization and Sinkhorn distances

In the following, assume that M is a metric matrix. Cuturi [Cut13] suggested regularization of the linear program (1) by restricting the feasible set $U(r,c)$ by an upper bound on the KL divergence from the so-called independence table $rc^T \in U(r,c)$.

Definition 14 (Sinkhorn distance)**.** For any $\alpha > 0$ define

$$U_\alpha(r,c) := \{P \in U(r,c) : \mathrm{KL}(P \parallel rc^T) \leqslant \alpha\} \subset \Sigma_{d\times d}$$

The Sinkhorn distance is then defined as $d_{M,\alpha} : \Sigma_d \times \Sigma_d \to \mathbb{R}$,

$$d_{M,\alpha}(r,c) := \min_{P \in U_\alpha(r,c)} \langle P, M \rangle := \min_{P \in U_\alpha(r,c)} \sum_{i,j} P_{ij} M_{ij} \tag{2}$$

Remark 15. In the above form, the Sinkhorn distance does not strictly satisfy the positive definiteness axiom. This can be remedied by considering $\mathbb{1}_{i\neq j} d_{M,\alpha}$ instead. The fact that $\mathbb{1}_{i\neq j} d_{M,\alpha}$ is a metric is not difficult to show, but is omitted here for brevity and its proof can be found in [Cut13].

Note that the problem (2) contains a hard constraint in the form of the feasible domain $U_\alpha(r,c)$. As we have seen in the course, the dual problem is usually more amenable to analysis in such cases.

Definition 16 (Dual Sinkhorn divergence)**.** For any $\lambda > 0$, define

$$P^\lambda = \operatorname*{arg\,min}_{P \in U(r,c)} \sum_{i,j} P_{ij} M_{ij} - \frac{1}{\lambda} h(P) \tag{3}$$

and define the dual Sinkhorn divergence as

$$d_M^\lambda(r,c) := \langle P^\lambda, M \rangle$$

Remark 17. The goal of Exercise 17 is to show that P^λ realizes the value of the dual function $f^d(0, \frac{1}{\lambda})$ corresponding to problem (2) for $\lambda > 0$. Then by strong duality, for every $\alpha \geqslant 0$ there exists $\lambda > 0$ such that

$$d_{M,\alpha}(r,c) = d_M^\lambda(r,c)$$

Nonetheless, the aim in Cuturi [Cut13] is not to find the λ corresponding to a given α, but rather to use λ as the tuning parameter for the strictly convex problem (3) instead of the hard constraint α, and work with d_M^λ instead of $d_{M,\alpha}$ even though it can only be shown to be a divergence.

Remark 18. By strict convexity of $-h(P)$ (see Exercise 3), the problem (3) is convex. Moreover, the particular structure of the problem allows for efficient, parallelized algorithms such as that of Sinkhorn and Knopp [KS67], which was also exploited by Cuturi [Cut13]. As a consequence, one can compute all pairwise dual Sinkhorn distances between N histograms of length d in complexity $O(d^2N^2)$.

Remark 19. There exists now the easy-to-use Python Optimal Transport package [FC17] which contains functions for computation of both Wasserstein distances and Sinkhorn divergences.

5 Marked problems

5.1 Optimal transport

Exercise 1. Let (X, d) be a metric space, let $\mathcal{D} = \{x_i\}_{i=1}^N$ be a dataset in X and define the cost matrix $(M_{ij})_{i,j=1}^N$ as $M_{ij} = d(x_i, x_j)$. Show that M is a metric matrix.

Exercise 2. Let $r = (\frac{1}{2}\ \frac{1}{2}), c = (\frac{1}{3}\ \frac{2}{3})$ be two histograms and $M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Find the optimal transport plan P for the optimal transport problem as per Definition 3.

5.2 Information theory

Exercise 3. Show from the definition that the entropy $h : \Sigma_d \to \mathbb{R}$ is non-negative. Looking at the function h on the open set $\{x \in \mathbb{R}_+^d : x > 0\}$, where it is well-defined, compute the Hessian matrix and conclude that h is concave, by referring to a result from Extra Problem Sheet 1. What about strict concavity?

Exercise 4. Show that for any two histograms $r, c \in \Sigma_d$, we have $rc^T \in U(r, c)$ and

$$h(rc^T) = h(r) + h(c)$$

Exercise 5. State what it means for two histograms p and q to be equivalent. Using that $y \mapsto \log(y)$ is strictly concave, show that $\mathrm{KL}(p \parallel q)$ as defined in Definition 12 is a divergence as per Definition 11.
Hint: Work with $y \mapsto -\log(y)$, noting that $\log \frac{p_i}{q_i} = -\log \frac{q_i}{p_i}$

Exercise 6. Is the KL divergence a metric? Prove or disprove.

Exercise 7. Show that $\mathrm{KL}(p \parallel q)$ as defined in Definition 12 is convex in the pair (p, q), that is $\forall (p, q), (p', q') \in \Sigma_d \times \Sigma_d, \forall \lambda \in [0, 1]$:

$$\mathrm{KL}(\lambda p + (1-\lambda)p' \parallel \lambda q + (1-\lambda)q') \leqslant \lambda \,\mathrm{KL}(p \parallel q) + (1-\lambda)\,\mathrm{KL}(p' \parallel q')$$

Hint: Write for every $i \leqslant d$:

$$\frac{\lambda p_i + (1-\lambda)p_i'}{\lambda q_i + (1-\lambda)q_i'} = \tilde{\lambda}\frac{p_i}{q_i} + (1-\tilde{\lambda})\frac{p_i'}{q_i'}$$

with

$$\tilde{\lambda} := \frac{\lambda q_i}{\lambda q_i + (1-\lambda)q_i'}, \quad 1 - \tilde{\lambda} = \frac{(1-\lambda)q_i'}{\lambda q_i + (1-\lambda)q_i'}$$

Exercise 8. Let $u = \frac{1}{d}\mathbb{1}_d = \frac{1}{d}(1 \dots 1) \in \mathbb{R}^d$ be the histogram of a uniform distribution. Show that

$$\mathrm{KL}(p \parallel u) = \log d - h(p)$$

By convexity of $\mathrm{KL}(p \parallel q)$, conclude again that the entropy $h : \Sigma_d \to \mathbb{R}$ is concave.

For Exercises 9-13, we consider the following setting. Let $f : \mathbb{R}^d \to \mathbb{R}^*$ be a convex function that is strictly convex differentiable on its domain. For any $x \in \operatorname{dom}(f)$, let H_x be the supporting hyperplane of $\operatorname{epi}(f)$ supported at $(x, f(x))$, i.e.

$$H_x = \{(y, h_x(y)) : y \in \mathbb{R}^n\}$$

for an affine function $h_x : \mathbb{R}^d \to \mathbb{R}$ with $h_x(x) = f(x)$ on $\operatorname{dom}(f)$.

Given a reference point $x_0 \in \mathbb{R}^d$ and any other point $x \in \mathbb{R}^d$, define the f-divergence from x to x_0 as

$$D_f(x_0 \parallel x) := f(x_0) - h_x(x_0).$$

For Exercises 9-12, you may assume $\operatorname{dom}(f) = \mathbb{R}^d$.

Exercise 9. Illustrate an example of $\operatorname{epi}(f)$ and H_x in a 2D drawing. Explain why the function h_x is uniquely determined and express $h_x(x_0)$ in terms of $\nabla f(x)$.

Exercise 10. By comparing $f(x_0)$ and $h_x(x_0)$, argue that $D_f(x_0 \parallel x)$ is a divergence in the sense of Definition 11. Illustrate this in the previous drawing.

Exercise 11. Show that

$$D_f\left(x_0 \| x\right) = f\left(x_0\right) + f^*\left(x^*\right) - x^* \cdot x_0$$

where $x^* := \nabla f(x)$ and f^* is the Legendre transform of f.

Exercise 12. Using the result of the previous exercise, show that the f-divergence from x_0 to x can be written in the 'dual' form

$$D_f(x \parallel x_0) = D_{f^*}(x_0^* \parallel x^*)$$

for the 'dual' variables $x^* := \nabla f(x)$ and $x_0^* := \nabla f(x_0)$.

Exercise 13. First, compute the Legendre transform of the negative entropy $-h : \Sigma_d \to \mathbb{R}$. Then confirm that it satisfies the relation from Exercise 12, and show that

$$\forall p, q \in \Sigma_d : D_{-h}(p \parallel q) = \operatorname{KL}(p \parallel q)$$

5.3 Entropic regularization

Exercise 14. Using the definitions, prove that

$$\operatorname{KL}(P \parallel rc^T) = h(r) + h(c) - h(P)$$

Show that, as a consequence, we can equivalently write

$$U_\alpha(r, c) = \{P \in U(r, c) : h(P) \geqslant h(r) + h(c) - \alpha\}$$

and deduce that $U_\alpha(r, c)$ is convex as a subset of $\Sigma_{d \times d}$, where $U_\alpha(r, c)$ is defined in Definition 14.

Exercise 15. For arbitrary $r, c \in \Sigma_d$, what is the Sinkhorn distance $d_{M,\alpha}(r, c)$ when $\alpha = 0$? What is the Sinkhorn distance for α large?

Exercise 16. Argue by Slater's condition that the problem (2) is strictly feasible and hence strong duality holds.

Exercise 17. Define the primal function corresponding to the convex optimization problem (2) as

$$f(P, \beta) = \begin{cases} \langle P, M \rangle & \text{if } -h(P) \leqslant -h(r) - h(c) + \alpha - \beta \\ +\infty & \text{otherwise} \end{cases}$$

Show that the dual function (with $\mu = 0$) is

$$f^d(0, \lambda) = \begin{cases} \inf_{P \in U(r,c)} -\lambda h(P) + \lambda(h(r) + h(c) + \alpha) + f(P, 0) & \text{if } \lambda \geqslant 0 \\ -\infty & \text{otherwise} \end{cases}$$

Conclude that P^λ as defined in Definition 16 is the transfer plan that realizes the value $f^d(0, \frac{1}{\lambda})$.

6 Computational component

In this section, we will consider a toy dataset of 1797 handwritten digits of size 8×8. The goal is simply to play a bit around with the EMD and Sinkhorn distances, so it is by no means the intention that this particular example is better solved with one of these distances. Indeed, we would need to consider more complex problems for these distances to become relevant.

For exercises 18-25, you may wish to download template code from `https://github.com/benediktpetko/convex-optimization-project`.

The aforementioned dataset is loaded by the script via scikit-learn. For plotting, we recommend using the matplotlib.pyplot plotting library.

Exercise 18. Reshaping the data suitably, plot the images of the first three hand-written digits.

Exercise 19. The images are 8×8 matrices of pixel intensities, (where we assume that the pixels exist on a uniformly spaced grid in the plane). What should a sensible cost matrix look like in terms of size and zero entries? Plot the cost matrix provided in the template script, and explain how it is constructed.

Exercise 20. Using the Python Optimal Transport package, compute the earth mover's distance between the first two images (for the template cost matrix). Report also the computational time.

Exercise 21. Using Python Optimal Transport, compute the dual Sinkhorn divergence between the first two images (for the template cost matrix). Note that you will need to pre-process the data to avoid error due to division by zero (hint: substitute zeroes for, say, 1e-17). Using only 3 iterations in the Sinkhorn algorithm (or something similarly small), do some trial and error to find a regularisation parameter *reg* $\in (0, 1]$ such that the distance is essentially the same as for EMD but at half the computational time (or less).

Exercise 22. Allowing yourself a large number of iterations (say, 1000), so that the algorithm should be close to the true Sinkhorn distance, report on what happens as you make *reg* $\in (0, 1]$ smaller and smaller. Can you relate this to the definition of the Sinkhorn distance in Section 4. (It is enough to work with the distance between the first two images in the dataset).

Exercise 23. Using scikit-learn, we split the data into a training set and a test set, assigning 10% of the data to the test set (this is already done in the template). With help of scikit-learn, apply the k-nearest neighbour algorithm (k-NN) to classify the test set. Report accuracy of the predicted labels against true labels, using the Euclidean metric for pairwise distances in the k-NN algorithm. You may use the default value of $k = 5$.

We will use the resulting accuracy as a benchmark for the following parts.

Exercise 24. Repeat Exercise 23, this time with EMD as the input metric for the k-NN algorithm, again reporting accuracy. Computational time aside, can you obtain sufficient accuracy for the EMD distance to be sensible for this classification task?

Exercise 25. Repeat Exercise 23, this time with the dual Sinkhorn divergence as the input "metric" for the k-NN algorithm, again reporting accuracy. Why is "metric" now in quotation marks? Using only 3 iterations (or similar) for the algorithm, show by trial and error (i.e., by finding a suitable *reg* $\in (0, 1]$) that the algorithm can be at least two times faster than the EMD from Exercise 24 with at least the same level of accuracy.

References

[KS67] P. Knopp and R. Sinkhorn. “Concerning Nonnegative Matrices and Doubly Stochastic Matrices”. In: *Pacific J. Math.* 21.2 (1967), pp. 343–348.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. ISBN: 0471241954.

[LC10] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: `http://yann.lecun.com/exdb/mnist/`.

[Cut13] M. Cuturi. “Sinkhorn distances: lightspeed computation of optimal transportation distances”. In: *NIPS’13: Proceedings of the 26th International Conference on Neural Information Processing Systems* 2 (2013), pp. 2292–2300.

[Kus+15] Matt Kusner et al. “From word embeddings to document distances”. In: *International Conference on Machine Learning*. 2015, pp. 957–966.

[Cou+17] N. Courty et al. “Optimal Transport for Domain Adaptation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.9 (2017).

[FC17] Rémi Flamary and Nicolas Courty. *POT Python Optimal Transport library*. 2017. URL: `https://pythonot.github.io/`.

[Tol+18] I. Tolstikhin et al. “Wasserstein Auto-Encoders”. In: *6th International Conference on Learning Representations (ICLR)*. May 2018. URL: `https://openreview.net/forum?id=HkL7n1-0b`.

[PC19] G. Peyré and M. Cuturi. “Computational Optimal Transport”. In: *Foundations and Trends in Machine Learning* 11.5-6 (2019).