Optimal Decision Making: Group Project
Optimal Transport and Distributionally Robust Optimization

In this project you will see how linear transportation problems can be used in image processing and data-driven portfolio selection. Please observe the following guidelines:

- You are supposed to hand in a report (as a pdf file) that contains your answers and explains your reasoning. Please provide a short description of your implementations in MATLAB.
- You must hand in your MATLAB files in a standardized format described below so that we can automatically test them. All linear programs should be solved with the GUROBI solver using the YALMIP interface. For guidelines on how to install GUROBI and to obtain a free academic license, see http://www.gurobi.com/academia/for-universities.
- A skeleton of each MATLAB script you should implement is available on Moodle (Project.zip). Please use this skeleton as a basis for your implementation. The name as well as the structure of the script should not be changed. Make sure that each script, namely `p1.m`, `p2.m`, `p33.m`, `p34.m`, and `p35.m` runs without errors. If your code does not run without errors, we will not be able to give you any points for the corresponding implementation exercise.
- Each team should upload a single zip file containing the report and the MATLAB code to Moodle. Please do **not** send your files by email. The deadline is on May 12 at midnight.

Description

A transportation or shipping problem determines the amount of goods or items to be transported from a number of sources or supply locations to a number of destinations or demand locations. The objective is to minimize the total shipping cost.

Transportation Problem

The transportation problem can be formalized as follows. Denote by ξ_i, $i = 1, \ldots, S$, the supply locations, by ξ'_j, $j = 1, \ldots, S'$, the demand locations, by $\mathbb{P}_i$ the supply at source i, by $\mathbb{Q}_j$ the demand at destination j and by c_{ij} the unit transportation cost from source i to destination j. A visualization of the transportation problem is provided in Figure 1. If the decision variable π_{ij} represents the quantity shipped from source i to destination j, then the transportation problem can be formulated as in (1).

$$\begin{array}{lll} \min\limits_{\pi} & \sum_{i=1}^{S} \sum_{j=1}^{S'} c_{ij}\, \pi_{ij} & \\ \text{s.t.} & \sum_{i=1}^{S} \pi_{ij} = \mathbb{Q}_j & \forall j = 1, \ldots, S' \\ & \sum_{j=1}^{S'} \pi_{ij} = \mathbb{P}_i & \forall i = 1, \ldots, S \\ & \pi_{ij} \geq 0 & \forall i = 1, \ldots, S \quad \forall j = 1, \ldots, S' \end{array} \tag{1}$$

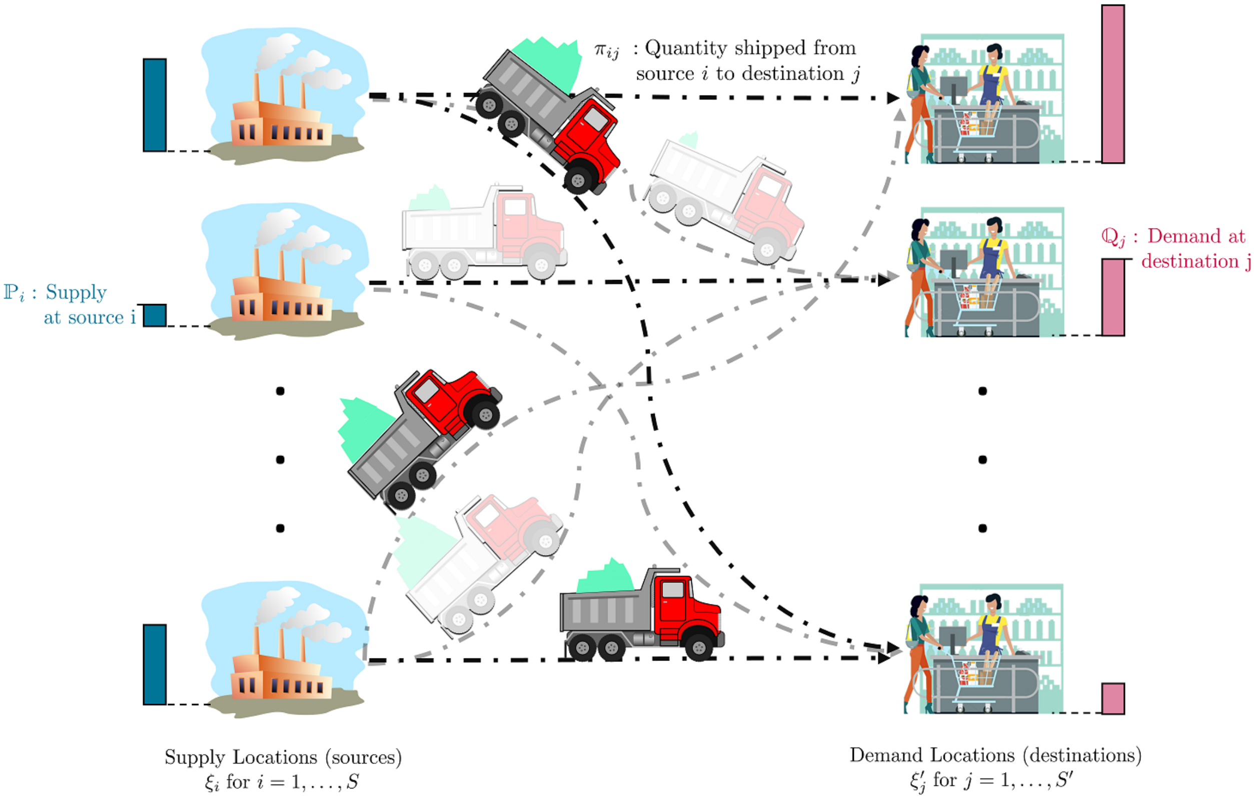


Figure 1: Illustration of the transportation problem

Wasserstein Distance

A popular distance measure between two probability distributions is the Wasserstein distance, which is inspired by the transportation problem. The Wasserstein distance between two discrete probability distributions $\mathbb{P}$ and $\mathbb{Q}$ denoted by $d(\mathbb{P}, \mathbb{Q})$, is defined as the square root of the optimal value of the transportation problem (1) with shipping costs $c_{ij} = \|\xi_i - \xi'_j\|^2$ (i.e., the squared Euclidean distance between ξ_i and ξ'_j).

Questions

1. Computing Wasserstein Distances [10 points]

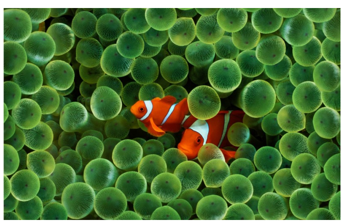
1.1. Let $\mathbb{P}$ be a probability distribution on the real line that assigns probabilities 1/2, 1/3 and 1/6 to the points 1, 2 and 3, respectively. Similarly, assume that $\mathbb{Q}$ assigns probabilities 2/5 and 3/5 to the points 1 and 2, respectively. Calculate the Wasserstein distance between $\mathbb{P}$ and $\mathbb{Q}$ numerically by implementing the transportation problem (1) using YALMIP and solving it with GUROBI. A skeleton of the code is provided in the MATLAB file `p1.m`

2. Color Transfer [40 points]

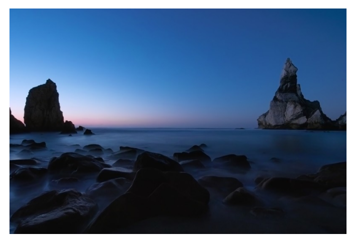
A digital image can be represented as a 3-dimensional array $X \in \mathbb{R}^{W \times H \times 3}$, where $W \in \mathbb{N}$ stands for the width and $H \in \mathbb{N}$ for the height of the image. Specifically, $(X_{wh1}, X_{wh2}, X_{wh3}) \in \mathbb{R}^3$ represents the vector of RGB (Red-Green-Blue) intensities of the pixel at position (w, h). In this exercise, you will perform a color transfer between two given RGB images by solving a transportation problem, which seeks the most natural way of transporting the color distribution of one image to that of another one.

Color transfer algorithms impose the color palette of the first image onto the second image and preserve the geometry of the second image. Color transfer is used in many computer vision and graphics problems. One main application is computational color constancy, which aims to remove color distortions due to illumination. It is also used to generate color-consistent image panoramas and 3D texture-maps, as well as to enhance and manipulate images by emulating the tone and the color style of other images.

We will solve a transportation problem between the color distributions of two images to find an optimal transportation map that allows us to convert one color palette to another one. Let $X \in \mathbb{R}^{W_1 \times H_1 \times 3}$ be the source image shown in Figure 2a and let $Y \in \mathbb{R}^{W_2 \times H_2 \times 3}$ be the target image shown in Figure 2b. We aim to modify the color distribution of the source image, which has the source geometry, to the color distribution of the target image, which has the target color palette.



(a) Source image



(b) Target image

Figure 2: Input images

2.1. [5 points] Plot the color distributions of the input images, which represent each pixel as a point in the 3-dimensional color space. For an example see Figure 3. To do so, you should reshape each image to a 2D array, i.e., you should transfer X to $X' \in \mathbb{R}^{W_1 \cdot H_1 \times 3}$ and Y to $Y' \in \mathbb{R}^{W_2 \cdot H_2 \times 3}$. A skeleton of the code you will have to implement is provided in the Matlab file `p2.m`.

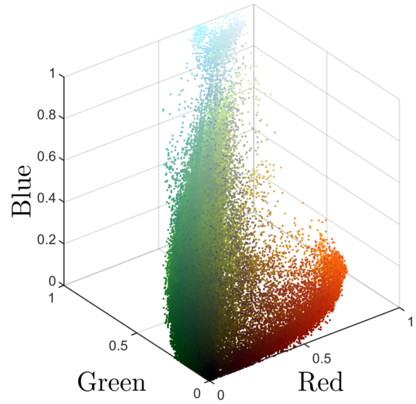


Figure 3: Color distribution of the source image

2.2. [10 points] Interpret X' as a probability distribution $\mathbb{P}$ that assigns each pixel in the color space $\mathbb{R}^3$ a probability $\frac{1}{W_1 \times H_1}$. Similarly, interpret Y' as a probability distribution $\mathbb{Q}$ that assigns each pixel in the color space $\mathbb{R}^3$ a probability $\frac{1}{W_2 \times H_2}$. Our ideal goal would be to compute the Wasserstein distance between $\mathbb{P}$ and $\mathbb{Q}$ and to determine the optimal transportation map between the color distributions. Formulate this transportation problem explicitly and determine the number of variables and constraints.

2.3. [25 points] Unfortunately, the exact transportation problem is huge and therefore computationally challenging. As a remedy, we can sub-sample the images X' and Y' using the MATLAB function `randperm` with a sample size of $N = 500$ to obtain the sampled source and target images $\hat{X} \in \mathbb{R}^{N \times 3}$ and $\hat{Y} \in \mathbb{R}^{N \times 3}$, respectively. Interpret $\hat{X}$ and $\hat{Y}$ as two uniform N-point distributions $\hat{\mathbb{P}}$ and $\hat{\mathbb{Q}}$ on $\mathbb{R}^3$. Calculate the Wasserstein distance between $\hat{\mathbb{P}}$ and $\hat{\mathbb{Q}}$ numerically using YALMIP and GUROBI, and let $\pi^\star \in \mathbb{R}^{N \times N}$ be the optimal transportation map. Define the re-colored sampled source image as $\hat{X}_{new} = N \cdot \pi^\star \cdot \hat{Y} \in \mathbb{R}^{N \times 3}$.
So far, we have re-colored only N pixels of the source image. To construct a full re-colored source image, $X'_{new} \in \mathbb{R}^{W_1 \cdot H_1 \times 3}$, we will try to find a linear relation between $\hat{X}$ and $\hat{X}_{new}$. Specifically, we will construct $B \in \mathbb{R}^{3 \times 3}$ such that $\hat{X}_{new} \approx \hat{X}B$ by solving the least squares regression problem:

$$\min_B \|\hat{X}_{new} - \hat{X}B\|_F^2, \tag{2}$$

where $\|A\|_F = \sqrt{\mathrm{Tr}(\mathrm{A}^\top \mathrm{A})}$. The optimal solution to (2) is $B^\star = (\hat{X}^\top \hat{X})^{-1} \hat{X}^\top \hat{X}_{new}$, where $(\hat{X}^\top \hat{X})^{-1} \hat{X}^\top$ represents the pseudo-inverse of $\hat{X}$, which can be computed using the MATLAB function `pinv`. Hence, $B^\star = \texttt{pinv}(\hat{X}) \cdot \hat{X}_{new}$. As $\hat{X}_{new} \approx \hat{X}B^\star$, it makes sense to set X'_{new} to $X'B^\star$. Construct the re-colored image X'_{new}, reshape X'_{new} to $X_{new} \in \mathbb{R}^{W_1 \times H_1 \times 3}$, and use the MATLAB function `imshow` to plot the re-colored image.

Apply the algorithm outlined above to the following three image pairs and show the re-colored source images in your report.

Target Image	**Source Image**
view.jpg	fish.jpg
sunset.jpg	coral.jpg
fall.jpg	spring.jpg

[BONUS] [10 points] Can you use the transportation map $\pi^\star$ found in Exercise 2.3 also to impose the color palette of the source image on the target image? If so, describe the precise procedure and use it to re-color the three target images in the above table.

3. Data-Driven Portfolio Optimization [50 points]

An investor wishes to invest into a portfolio of K assets with uncertain rates of return $\tilde{\boldsymbol{\xi}} \in [-1, +\infty)^K$. The investor's decision is the weight x_k to attribute to each asset k. Since short sales are forbidden, the vector of portfolio weights $\boldsymbol{x}$ ranges over the feasible set $\mathcal{X} = \left\{\boldsymbol{x} \in \mathbb{R}_+^K : \mathbf{1}^\top \boldsymbol{x} = 1\right\}$, where $\mathbf{1}$ stands for the vector of ones. The rate of return of the portfolio is given by $\boldsymbol{x}^\top \tilde{\boldsymbol{\xi}}$. Assume that the investor's objective is to maximize the expected value of the concave piecewise linear utility function

$$u(\boldsymbol{x}^\top \tilde{\boldsymbol{\xi}}) = \min_{l=1,\ldots,L} \left\{ a_l \boldsymbol{x}^\top \tilde{\boldsymbol{\xi}} + b_l \right\},$$

where $a_l \in \mathbb{R}_+$ and $b_l \in \mathbb{R}$ for all $l = 1, \ldots, L$. The expectation should be calculated under the distribution $\mathbb{P}$ of $\tilde{\boldsymbol{\xi}}$. However, this distribution is unknown, and the investor has only access to N independent training samples $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N$ from $\mathbb{P}$. These samples must be used to estimate the expected utility of an investment decision. In this exercise, we will compare (i) the sample average approximation, which approximates $\mathbb{P}$ by the empirical distribution $\hat{\mathbb{P}}$ that assigns the same probability $\frac{1}{N}$ to each training sample, with (ii) the distributionally robust approach, which evaluates the worst-case expected utility with respect to all discrete distributions $\mathbb{Q}$ on the training samples that have a Wasserstein distance of at most ρ from the empirical distribution $\hat{\mathbb{P}}$. In the following, we denote by $\mathcal{B}_\rho(\hat{\mathbb{P}})$ the family of all N-point distributions on $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N$ with $d(\hat{\mathbb{P}}, \mathbb{Q}) \leq \rho$.

3.1. [5 points] Formulate the following sample average approximation (SAA) problem as a linear program.

$$\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}_{\hat{\mathbb{P}}}\, u(\boldsymbol{x}^\top \tilde{\boldsymbol{\xi}}) = \max_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{N} \sum_{i=1}^{N} u\left(\boldsymbol{x}^\top \boldsymbol{\xi}_i\right) \tag{SAA}$$

3.2. [15 points] Formulate the following distributionally robust optimization (DRO) problem as a linear program.

$$\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\mathbb{Q} \in \mathcal{B}_\rho(\hat{\mathbb{P}})} \mathbb{E}_{\mathbb{Q}}\, u(\boldsymbol{x}^\top \tilde{\boldsymbol{\xi}}) = \max_{\boldsymbol{x} \in \mathcal{X}} \min_{\mathbb{Q} \in \mathcal{B}_\rho(\hat{\mathbb{P}})} \sum_{i=1}^{N} \mathbb{Q}_i\, u\left(\boldsymbol{x}^\top \boldsymbol{\xi}_i\right) \tag{DRO}$$

Hint: Use strong duality.

3.3. [5 points] Implement the SAA problem for the $N = 10{,}000$ samples provided in the file `test.mat` and report its optimal value. Use the skeleton code in the file `p33.m`.

3.4. [5 points] Implement the DRO problem for the $N = 30$ training samples provided in the file `train.mat` for a Wasserstein radius of $\rho = 0.9$. Report the mean out-of-sample utility of the optimal decision $\boldsymbol{x}^\star$ on the $N = 10{,}000$ test samples from the file `test.mat`. Use the skeleton code in the file `p34.m`.

3.5. [10 points] Solve the SAA and the DRO problems for 1,000 independent datasets, each containing 30 training samples, and plot the cumulative distribution function of each solutions' out-of-sample utility normalized with respect to the result of Exercise 3.3. Use the skeleton code in the file `p35.m`.

3.6. [10 points] Interpret the results of Exercise 3.5.