Project Report

Optimal Decision Making

Spring 2019

Team members :

Ke Dan
Reza Hosseini
Jangwon Park
Thammisetty Devakumar

Professor
Daniel Kuhn

Teaching Assistants
Bahar Taskesen
Dirk Lauinger

May 9, 2019

Contents

1 Computing Wasserstein Distance

Since Wasserstein distance is defined as the *square root* of the optimal objective to the transportation problem, we take the square root of value(obj) at the end. The Wasserstein distance between $\mathbb{P}$ and $\mathbb{Q}$ is 0.5164.

2 Color Transfer

2.1 Plotting Color Distributions

See p2.m.

2.2 Exact Formulation of Color Transfer Problem

2.2.1 Definitions

We clarify the following notations for the parameters in the problem formulation:

- S = number of pixels in each channel of source image = $W_1 \times H_1$.
- S' = number of pixels in each channel of target image = $W_2 \times H_2$.
- c_{ij} = squared Euclidean norm between the location of pixel i of source image and location of pixel j of target image, where $i = \{1, ..., S\}$ and $j = \{1, ..., S'\}$. Each pixel is described by a vector in $\mathbb{R}^3$.
- $\mathbb{P}$ = a vector in $\mathbb{R}^S$, which describes a probability distribution of each pixel color of source image. All elements are equal to $\frac{1}{S}$.
- $\mathbb{Q}$ = a vector in $\mathbb{R}^{S'}$, which describes a probability distribution of each pixel color of target image. All elements are equal to $\frac{1}{S'}$.

2.2.2 Formulation

Given the above, an element x_{ij} of the decision variable $\boldsymbol{X} \in \mathbb{R}^{S \times S'}$ represents the *quantity* of color values of pixel i from the source image to transport to the color values of pixel j in the target image. Therefore, the full formulation is as follows:

$$\min_x \sum_{i=1}^{S} \sum_{j=1}^{S'} c_{ij} x_{ij} \tag{1}$$

subject to:

$$\begin{aligned} \sum_{i=1}^{S} x_{ij} &= \frac{1}{S'}, \forall j = 1, ..., S' \\ \sum_{j=1}^{S'} x_{ij} &= \frac{1}{S}, \forall i = 1, ..., S \\ x_{ij} &\geq 0, \forall i = 1, ..., S, \forall j = 1, ..., S' \end{aligned} \tag{2}$$

The number of variables is equal to $S \times S' = W_1 \times H_1 \times W_2 \times H_2$. If the source and target images are both 500 pixels wide and 800 pixels tall, for example, then this amounts to $(500{\times}800)^2$ = 160,000,000,000 variables. The number of constraints is then equal to $S' + S + S \times S' = W_2 \times H_2 + W_1 \times H_1 + W_1 \times H_1 \times W_2 \times H_2$. With the same images, this amounts to $2{\times}500{\times}800 + (500{\times}800)^2$ = 160,000,800,000 including non-negativity constraints.

2.3 Recoloring Images by Subsampling

The following images are recolored source images based on the color palette of their designated target images. The left images are the original images while the images on the right are recolored versions.

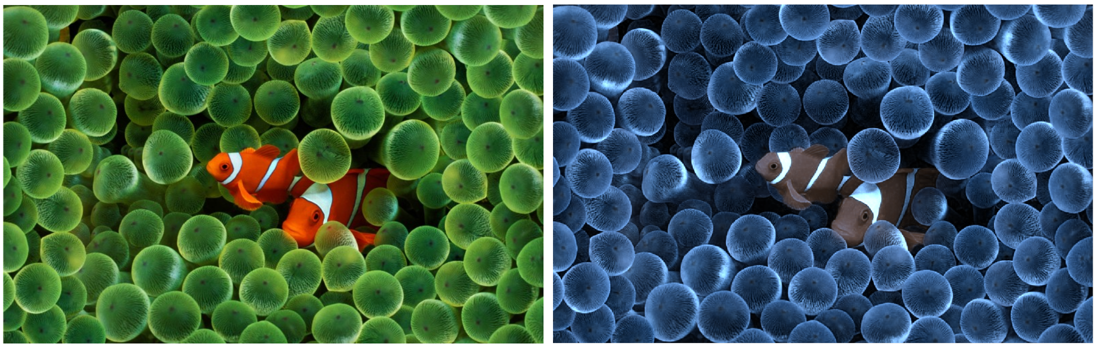

Figure 1: Recolored fish image based on view.jpg

Figure 2: Recolored coral image based on sunset.jpg

Figure 3: Recolored spring image based on fall.jpg

2.4 Bonus Question

The interpretation of π^* is the optimal quantity of color values of pixel i in the source image to *transport* to the color values of pixel j in the target image i.e. how much of pixel i to be recolored in the colors of pixel j. Therefore, the transpose of this matrix i.e. π^{*T} can be interpreted as the exact reverse of π^*, and can be used to impose the color palette of the source image on the target image.

To recolor target images, first we must define $\hat{Y}_{new} = N * \pi^{*T} * \hat{X}$. This imposes the color palette of the sub-sampled source image $\hat{X}$ to recolor the sub-sampled target image. Next, we solve for B in the least squares regression problem:

$$\min ||\hat{Y}_{new} - \hat{Y}B||_F^2 \tag{3}$$

where the optimal solution $B^* = (\hat{Y}^T\hat{Y})^{-1}\hat{Y}^T\hat{Y}_{new}$. The matrix $(\hat{Y}^T\hat{Y})^{-1}\hat{Y}^T$ is the pseudo-inverse of $\hat{Y}$, computed directly by the pinv() function in MATLAB. With this, we can state $\hat{Y}_{new} \approx \hat{Y}B^*$ which implies $Y'_{new} \approx Y'B^*$. The final step is just to reshape Y'_{new} into the original dimensions of the target image $W_2 \times H_2 \times 3$.

The following images are recolored target images based on the color palette of the designed source images. Again, the left images are the original images while the right ones are the recolored versions.

Figure 4: Recolored view image based on fish.jpg

Figure 5: Recolored sunset image based on coral.jpg

Figure 6: Recolored fall image based on spring.jpg

3 Data-Driven Portfolio Optimization

3.1 Sample Average Approximation (SAA)

SAA as a linear program:

$$\max_{x,z} \frac{1}{N} \sum_{i=1}^{N} z_i \tag{4}$$

subject to the following

$$z_i \leq a_l \mathbf{x}^T \zeta_i + b_l \qquad \forall l = 1, ..., L \text{ and } \forall i = 1, ..., N \tag{5}$$

$$x_k \geq 0 \qquad \forall k = 0, \ldots, K \tag{6}$$

$$\sum_{i=0}^{K} x_i = 1 \qquad \forall k = 0, \ldots, K \tag{7}$$

where ζ_i is the return vector of all K assets under scenario i.

3.2 Distributionally Robust Optimization (DRO)

In the DRO formulation, we first replace $u(\mathbf{x}^T \zeta_i)$ with z_i and some additional constraints as shown in the SAA formulation. Next, we introduce the constraints on the variables $\mathbb{Q}$ which state that its Wasserstein distance to $\mathbb{P}$ must not be greater than $\rho = 0.9$. The Wasserstein distance can be computed by taking the square root of the optimal value of the transportation problem.

$$\max_{x,z} \min_{Q,\pi} \sum_{i=1}^{N} \mathbb{Q}_i z_i \tag{8}$$

subject to:

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}\pi_{ij} &\leq \rho^2 \\
\sum_{i=1}^{N} \pi_{ij} - \mathbb{Q}_j &= 0, \qquad \forall j = 1, ..., N \\
\sum_{j=1}^{N} \pi_{ij} &= \mathbb{P}_i, \qquad \forall i = 1, ..., N \\
\pi_{ij} &\geq 0, \qquad \forall i = 1, \ldots N, \text{ and } j = 1, \ldots N \\
a_l \mathbf{x}^T \zeta_i + b_l &\geq z_i, \qquad \forall l = 1, \ldots L, \text{ and } i = 1, \ldots, N \\
\sum_{i=1}^{K} x_i &= 1 \\
x_k &\geq 0, \qquad \forall k = 1, ..., K
\end{aligned} \tag{9}
$$

where $c_{ij} = ||\zeta_i - \zeta_j||^2$ (i.e. squared Euclidean norm of the difference in return vectors i and j), and $\mathbb{P}_i = \frac{1}{N}$ as provided.

Prior to dualizing the inner problem, we introduce some notation for convenience:

- r: dual variable for the first primal constraint
- y_i: dual variable for the N constraints specified by the second constraint in (9)
- t_i: dual variable for the N constraints specified by the third constraint in (9)

Therefore, the DRO problem now becomes:

$$
\max_{x,z,r,t,y} \rho^2 r + \sum_{i=1}^{N} \frac{t_i}{N} \tag{10}
$$

subject to:

$$
\begin{aligned}
c_{ij} r + y_j + t_i &\leq 0, \qquad \forall i, j = 1, ..., N \\
-y_j &\leq z_j, \qquad \forall j = 1, ..., N \\
r &\leq 0 \\
a_l \mathbf{x}^T \zeta_i + b_l &\geq z_i, \qquad \forall l = 1 \ldots L, \forall i = 1 \ldots N \\
\sum_{i=1}^{K} x_i &= 1 \\
x_k &\geq 0, \qquad \forall k = 1, ..., K
\end{aligned} \tag{11}
$$

3.3 SAA Implementation

In this case, we have $N = 10000$ testing samples for the rate of return of $K = 20$ assets. The piecewise linear concave utility is comprised of two linear functions with the following parameters:

- $a_1 = 4$, $b_1 = 0$;
- $a_2 = 1$, $b_2 = 0$;

By implementing the SAA with the test data, we get the optimal mean utility of 0.5712.

3.4 DRO Implementation

In this problem, we first obtain the optimal decision vector $\mathbf{x}^*$ by applying the DRO linear program on the 30 training samples with $\rho^2 = 0.81$. Then $\mathbf{x}^*$ is used to compute the mean out-of-sample utility by averaging the utilities from its inner product with each of the same 10,000 test samples as those in the SAA Implementation, which is 0.5255.

3.5 SAA and DRO

We solve the SAA and DRO linear programs for each of the 1,000 independent samples generated by *sample_data.m* script. We use $\rho = 0.9$ for solving the DRO. Finally, we normalize the result with the value of objective function obtained in Section 3.3, which is 0.5712. The cumulative distribution function of the mean out-of-sample utilities is shown in Figure 7.

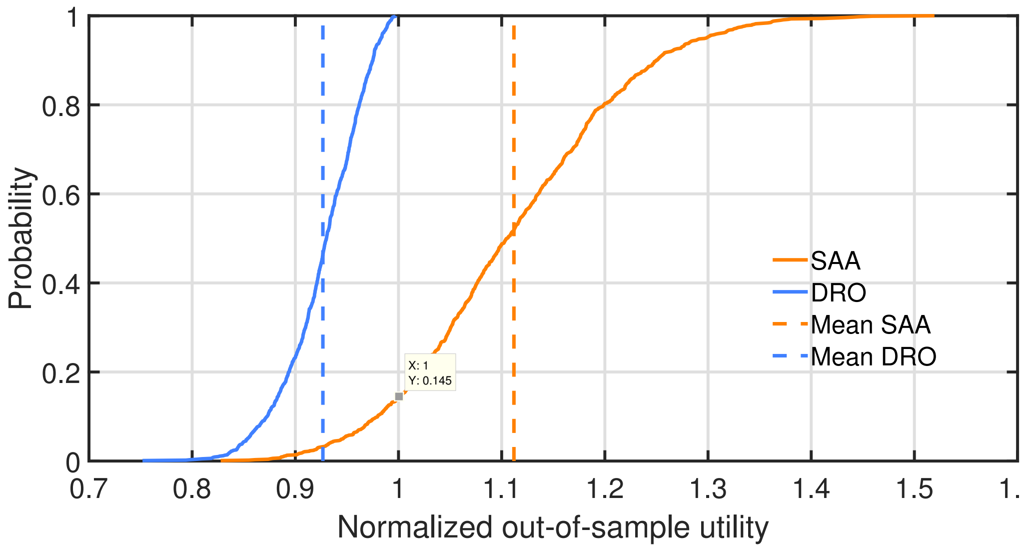

Figure 7: CDF of mean out-of-sample utility for both SAA and DRO derived from 1000 independent samples

3.6 Interpretation of Section 3.5

We refer to Figure 7 and make following observations:

- In this figure, we plot the cumulative distribution functions of mean out-of-sample utilities of both SAA and DRO obtained using each of the 1,000 independent data samples, and normalized by the SAA solution obtained in Section 3.3.

- Result of the SAA problem in Section 3.3 represents the 'true' expected utility of our optimal investment decision which we seek to approximate as closely as possible using only the 1,000 independent datasets each with only 30 training samples. In Figure 7, this is the represented as 1 on the x-axis.

- **Observation 1:** We see that the mean of normalized out-of-sample utilities using SAA over 1000 samples is 1.112 whereas the mean of the same quantity using DRO is 0.9267. Therefore, on average the DRO approach approximates the 'true' expected utility more accurately.

- **Observation 2:** It is also evident that by design the DRO approach always underestimates the 'true' expected utility, but can get very close (nearly identical) to it when we observe the maximum mean out-of-sample utility. In Figure 7, DRO's maximum expected utility is 0.9974 which is approximately equal to 1. On the other hand, the SAA approach overestimates the 'true' expected utility most of the time. In fact, roughly 85.5% of all approximations are overestimates. This is in general bad for an investor since it portrays an investment decision optimistically when in reality it may not be very good.

- **Observation 3:** The standard deviation of mean utilities from the SAA approach is 0.1065 while it is 0.0392 for the DRO approach. Approximations from the DRO approach are therefore much more robust and hence present less risk to the investor. This is because one can expect that investment decisions optimized by the DRO approach lead to less fluctuating approximations of the 'true' expected utility.

- We conclude that approximations from the DRO approach are not only more accurate on average, but are more robust to predict the 'true' expected utility compared to the SAA approach.