Project Optimal Decision Making

Yves Rychener, Jordan Metz, Simon Maksay and Tobias Barblan
Group E

Registration no.
E-mail:

Problem 1 : Computing Wasserstein distances

Using the MATLAB script *p1.m*, we can calculate the Wasserstein distance. It is 0.2667. Inspecting the transportation map obtained, we can confirm that it makes sense. To go from $\mathbb{P}$ to $\mathbb{Q}$, we take mass 0.1 from 1 to 2 and move all the mass from 3 to 2.

Problem 2 : Color transfer

The whole code can be found in the script *p2.m* in the appendix.

2.1 Color Distribution

We reshape the image and plot its histogram. In Figure 1 we plot the histogram for the image *fish.jpg*

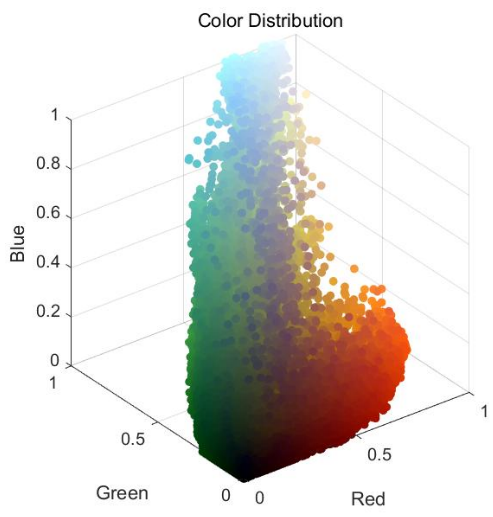


Figure 1: Color Histogram of the image *fish.jpg*

2.2 Formulation of the Transportation Problem

Define c_{ij} as the squared Euclidean distance between pixel i in the source image and pixel j in the target image. Then the transportation problem can be formulated in the following way:

$$\begin{aligned} \underset{\pi \in \mathbb{R}^{W_1 H_1 \times W_2 H_2}}{\text{minimize}} \quad & \sum_{i=0}^{W_1 H_1} \sum_{j=0}^{W_2 H_2} c_{ij} \pi_{ij} \\ \text{subject to} \quad & \sum_{j=0}^{W_2 H_2} c_{ij} \pi_{ij} = X'_i \qquad \forall i \\ & \sum_{i=0}^{W_2 H_2} c_{ij} \pi_{ij} = Y'_j \qquad \forall j \\ & \pi_{ij} \geq 0 \qquad\qquad \forall i, j \end{aligned} \tag{TP}$$

From Problem Formulation (TP) we can clearly see that the number of variables is $N_{var} = dim(\pi) = W_1 H_1 W_2 H_2$ and the number of constraints is $N_{constraints} = W_1 H_1 + W_2 H_2 + W_1 H_1 W_2 H_2$.

2.3 Solving The Transportation Problem And Performing The Color Transfer

We follow the method proposed in the question. For this, we first sub sample the images and find the optimal transportation map on the sub sampled points. After applying this map to the sub sampled points, we can use the resulting data-set of $[X, X_{transformed}]$ to calculate a linear estimator B for the color-change mapping. We can then apply the estimated mapping B to the original image to heave an (approximate) solution to the color transfer problem. The results are shown in Figure 2 together with the results for the Bonus question.

Bonus Question : Performing The Color Transfer On The Target Image

The transport map π^* we found specifies the transport to get from the color distribution of the source image to the distribution of the target image. More specifically, the entry $\pi^*_{i,j}$ specifies the amount of mass (proportion of pixels) we need to transport from entry i in the source histogram to the entry j in the target histogram. We can therefore use the transpose of π^* to give us information about the transport in the reverse direction. To transfer the colors from the source image to the target image we can perform the same procedure as in Section 2.3, but with π^{*T} instead of π^*. Figure 2 shows the result. Since the squared Euclidean distance is symmetric, the transportation costs in each direction are equal ($c_{ij} = c_{ji}$). The transportation map π^{*T} is therefore optimal in the problem we would formulate to transfer the source palette to the target image.

Problem 3 : Data-driven portfolio optimization

3.1 Formulating SAA as a linear program

The Problem can be reformulated to the following, using epigraphical reformulation for $u(x^T\xi)$:

$$\begin{aligned} \underset{u\in\mathbb{R}^L, x\in\mathbb{R}^K_+}{\text{maximize}} \quad & \mathbb{1}^T u \\ \text{subject to} \quad & \mathbb{1}^T x = 1 \\ & u_i \leq \underset{l=1,..,L}{min}\, a_l x^T \xi_i + b_l \quad \forall i \end{aligned} \tag{1}$$

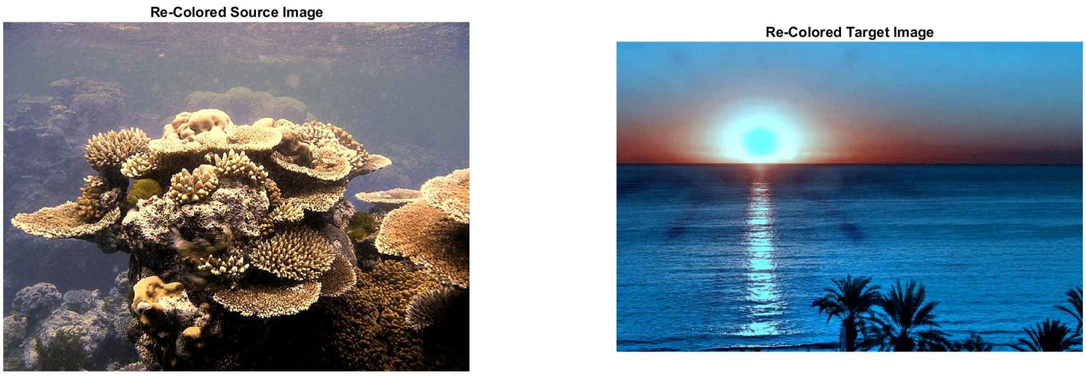
Since the constraint for u_i has to be satisfied for the minimum in the uncertainty set, it has to be satisfied for all right hand sides (all l):

$$\begin{aligned} \underset{u\in\mathbb{R}^L, x\in\mathbb{R}^K_+}{\text{maximize}} \quad & \mathbb{1}^T u \\ \text{subject to} \quad & \mathbb{1}^T x = 1 \\ & u_i \leq a_l x^T \xi_i + b_l \quad \forall i, l \end{aligned} \tag{LP-SAA}$$

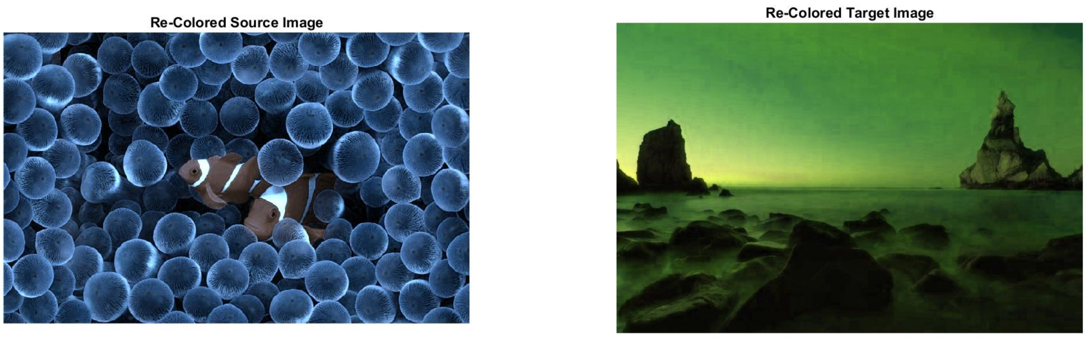
3.2 Formulating DRO as a linear program

To reformulate the distributionally robust optimization (DRO), we first need to get ride of the inner minimization problem.

$$\min_{\mathbb{Q}\in\mathcal{B}_\rho(\hat{\mathbb{P}})} \quad \sum_{i=1}^{N} \mathbb{Q}_i u(\mathbf{x}^T \xi) \tag{2}$$



a) *coral.jpg* and *sunset.jpg*



b) *fish.jpg* and *view.jpg*

c) *spring.jpg* and *fall.jpg*

Figure 2: Results After Color Transfer

Which is equivalent to

$$
\begin{aligned}
\min_{\mathbb{Q},\pi} \quad & \sum_{i=1}^{N} \mathbb{Q}_i u(\mathbf{x}^T \xi) \\
\text{s. t.} \quad & \sum_{i=1}^{S} \sum_{j=1}^{S'} c_{ij}\pi_{ij} \leq \rho \quad \text{(drop the min)} \\
& \sum_{i=1}^{S} \pi_{ij} = \mathbb{Q}_j \qquad\qquad\qquad \forall j \\
& \sum_{j=1}^{S} \pi_{ij} = \mathbb{P}_i \qquad\qquad\qquad \forall i \\
& \pi_{ij} \geq 0 \qquad\qquad\qquad\qquad \forall i, j
\end{aligned} \tag{P}
$$

For this purpose, we want to find its equivalent dual problem in order to combine the two maximization problems in one.
The Lagrangian can be written as:

$$
L(\mathbb{Q}, \pi, \lambda_1, \lambda_2, \mu_1, \mu_2) =
$$

$$
\sum_{i=1}^{N} \mathbb{Q}_i u(\mathbf{x}^T \xi) + \lambda_1 (\sum_{i,j} c_{ij}\pi_{ij} - \rho) + \mu_1^T (\begin{bmatrix} \vdots \\ \sum_{i=1}^{S} \pi_{ij} - \mathbb{Q}_j \\ \vdots \end{bmatrix}) + \mu_2^T (\begin{bmatrix} \vdots \\ \sum_{j=1}^{S} \pi_{ij} - \mathbb{P}_i \\ \vdots \end{bmatrix}) + \sum_{i,j} \lambda_{2_{i,j}} \pi_{i,j}
$$

By finding the infimum of the lagrangian with respect to the primal variables, and by using the same trick as for equation (LP-SAA), the dual can be written as follow

$$
\begin{aligned}
\max_{\lambda_1, \mu_2, j} \quad & -\lambda_1 \rho - \mu_2^T \mathbb{P} \\
\text{s.t.} \quad & \lambda_1 c + \mathbb{1}\mathbf{j}^T + \mu_2^T \mathbb{1} \geq 0 \\
& j_i \leq a_l \mathbf{x}^T \xi_{\mathbf{i}} + b_l \qquad \forall l, i \quad \text{(Drop the min)} \\
& \lambda_1 \geq 0
\end{aligned} \tag{D}
$$

Hence the final linear program corresponding to the (DRO) can be formulated as follow

$$
\begin{aligned}
\max_{x, \lambda_1, \mu_2, j} \quad & -\lambda_1 \rho - \mu_2^T \mathbb{P} \\
\text{s.t.} \quad & \lambda_1 c + \mathbb{1}\mathbf{j}^T + \mu_2^T \mathbb{1} \geq 0 \\
& j_i \leq a_l \mathbf{x}^T \xi_{\mathbf{i}} + b_l \qquad \forall l, i \\
& \lambda_1 \geq 0 \\
& \mathbb{1}^T \mathbf{x} = 1 \\
& \mathbf{x} \geq 0
\end{aligned} \tag{LP-DRO}
$$

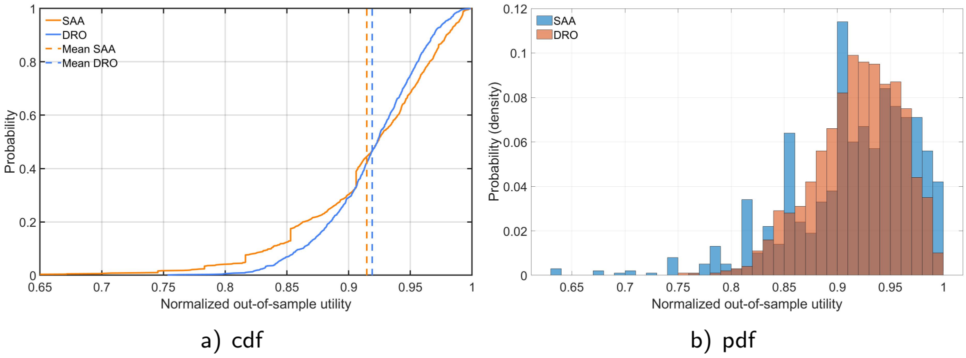


Figure 3: Out-of-sample Utility

With $j = \begin{bmatrix} \vdots \\ u(x^T\xi_i) \\ \vdots \end{bmatrix}$ and c_{ij} the Wasserstein distance between samples i and j. The distribution $\mathbb{Q}$ is assumed to have the same support as $\mathbb{P}$, so the distances are just computed between the sample returns.

3.3 Implementation of SAA

We changed the code of the skeleton to also print the decision vector x, since the portfolio allocation is also of interest. The resulting optimal utility we are getting when training on the whole dataset (test set) is 0.5712. This is obtained by allocating our portfolio (rounded values) 15.4% to asset 18, 34.8% to asset 19 and the rest to asset 20.

3.4 Implementation of DRO

As before, we are also interested in the portfolio allocation, so we print it. The resulting optimal average utility is 0.5262 compared to 0.5114 obtained when training the SAA only on the training set. The optimal portfolio Allocation obtained with DRO is 6.5% for asset 14, 7.9% for asset 15, 6.3% for asset 16, 26% for asset 17, 33.8% for asset 18, 6.3% for asset 19 and the rest to asset 20. We can see that the solution to the DRO is more diversified (7 assets vs 3 assets for the SAA trained on only the training set).

3.5 Putting everything together and solving for independent data sets

In order to avoid having too many printouts, we remove the portfolio print to console we used before. Figure (3a) shows the resulting plot. It is the empirical cdf of normalized utility after training on a random training set.

3.6 Interpretation

We also plot the cdf by plotting the histogram of portfolio utility, it is seen in Figure (3b). We can see that the SAA portfolio allocation results in heavier lower tails. This means that the chance of having very low utility (which is bad) is higher. Also, the chance of having very high utility is higher (heavier upper tails). Overall we can say that the utility of the DRO allocation has less variance. This means that the out of sample returns are more predictable, which is preferable. Also, the DRO has on average higher utility, which is what we want.

Conclusion

The DRO portfolio allocation is maximizing the utility for the worst possible case in an uncertainty set around the empirical training return distribution. The DRO is therefore maximizing the lower bound for all out of sample returns inside the uncertainty set. Since the empirical training return distribution is a sample of the true return distribution, the true return distribution is (with high probability) close to the empirical training return distribution. Therefore the DRO portfolio allocation suffers less from heavy lower tails than the SAA, where we just maximize the utility for the observed probability distribution. The DRO results, as is usually the case for robust optimization, in a more "save bet" result.

Limitations

To model the probability distribution of the returns using the sample returns, we have to assume stationnarity. This can for example be obtained with time series modelling or differenentiation in a real world scenario. The main limitation of the optimization approach used is the fact that we assumed the distributions in the uncertainty set to have the same support. This is of course not the case in reality since the returns follow a continuous distribution. Also, with this approach we cannot benefit from one key strength of the Wasserstein distance: the fact that it takes into account Black-Swan-Events, extreme events with very low probability. One approach that could somehow deal with this without changing the optimization problem too much would be to use different sample positions for $\mathbb{Q}$ than the ones from the training set. They could be sampled on a grid which would have to be big enough to include Black Swans, or one could randomly sample them, for example from a multivariate distribution fitted to the training samples. We would propose using a T-Distribution, since it has heavier tails than the normal distribution and would include more Black-Swan sampling positions.