EE 127 / EE 227AT 9/27/18
L. El Ghaoui

HW #3 Solutions

Exercise 1 (Kantorovich Problem)

Question: How to move mass μ to ν with minimal cost?

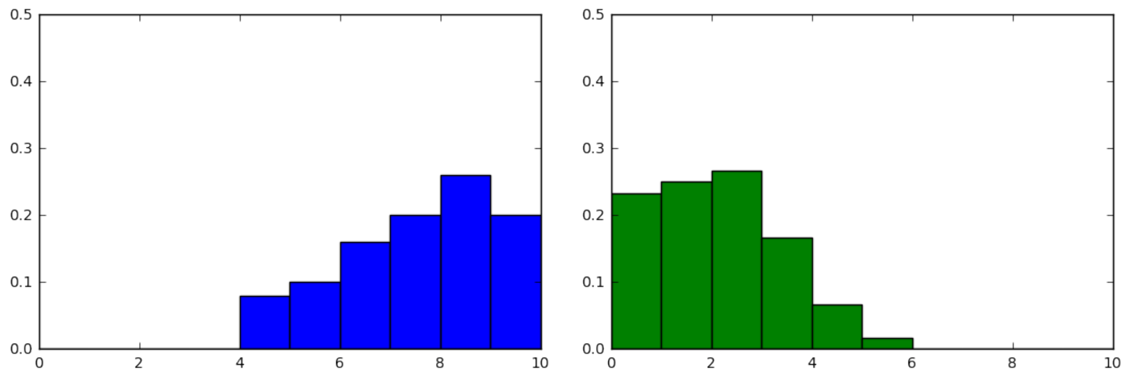


Figure 1: Visualization of μ histogram on left and ν histogram on right.

More rigorously, let $n \in \mathbb{N}$. We define two discrete probability distributions $\mu = (\mu_1, \cdots, \mu_n)$ and $\nu = (\nu_1, \cdots, \nu_n)$ with $\sum_i \mu_i = \sum_i \nu_i = 1$. We define $C = (c_{ij})_{\substack{1\cdots n \\ 1\cdots n}} \in \mathbb{R}_+^{n^2}$ be the cost matrix for transporting one unit of mass from location $i \in [1, \cdots, n]$ to location $j \in [1, \cdots, n]$. The flow matrix $P = (p_{ij})_{\substack{1\cdots n \\ 1\cdots n}}$ denotes the quantity of mass to be moved from location i to location j. P satisfies the limit conditions:

$$P\mathbb{1}_n = \mu$$
$$P^T\mathbb{1}_n = \nu$$

1. What is the total cost of transporting the mass μ into ν by following the transportation plan P?
2. Write the optimization problem of finding the transportation plan P with minimal cost. What type of optimization problem is it? (LP, QP, $\cdots$?).

The optimal value of the optimization problem you found is a well-defined distance named *Kantorovich* or Wasserstein distance noted $\mathcal{W}(\mu, \nu)$. That distance can be readily applied in the context of document similarity. In particular, while computating the Kantorovich between two documents, we obtain a flow matrix. The point of this question is to show how this matrix can be interpreted. For that, we provide you with a certain prior distance on words (word embeddings) that can be used to compute the transportation cost in the notebook `text-kantorovich.ipynb`.

3. Calculate the Wasserstein distance in the notebook using CVX and use visualize the resulting flow matrix P using the provided code. Interpret the results.
4. Compute the cosine distance between the two given documents. Comment on which distance is more insightful on this precise example and why.

   Note: Cosine distance between two documents (or sets of words) is computed as follows. Collect ordered sets of all word counts from both documents. Then compute the cosine of the angle between the two vectors. For example from "black blue black" and "blue blue red", one would obtain the sets $\{black : 2, blue : 1, red : 0\}$ *and* $\{black : 0, blue : 2, red : 1\}$. *We can then compute the cosine distance between the vectors* $[2, 1, 0]$ *and* $[0, 2, 1]$.

We are now interested in computing different barycenters between two discrete vectors μ and ν shown in Figure 1. Let us note that the convex combination $t\mu + (1-t)\nu$ has the variational formulation $\mathrm{argmin}_x\, t(\mu - x)^2 + (1-t)(\nu - x)^2$. Inspired from this fact, we define the variational formulation for convex combinations on the Wasserstein space $\mathrm{argmin}_x\, t\mathcal{W}(x, \mu) + (1-t)\mathcal{W}(x, \nu)$.

5. Write this optimization problem as a LP.
6. Solve the optimization problem using CVX in the notebook `barycenter.ipynb`. Visualize the Waserstein barycenter for $t \in [0, 0.25, 0.5, 0.75, 1.0]$.
7. Use your results to comment on the differences between Wasserstein convex combination and euclidean convex combination.

Solution 1

1. The total cost can be written as:

$$\langle P, C \rangle = \sum_i \sum_j P_{ij} C_{ij}$$

2. The problem can be formulated as a LP:

$$\min_{\substack{P \in \mathbb{R}_+^{n \times m} \\ P\mathbb{1}_n = \mu, P^T\mathbb{1}_n = \nu}} \quad \langle P, C \rangle$$

Where $C = (c_{ij})_{\substack{1..n \\ 1..n}}$ is the known cost matrix and we are solving for transportation matrix P.

3. Each word is mapped to its closest word in semantic domain (word2vec embedding). See code solution in text_kantorovich_sols.ipynb.

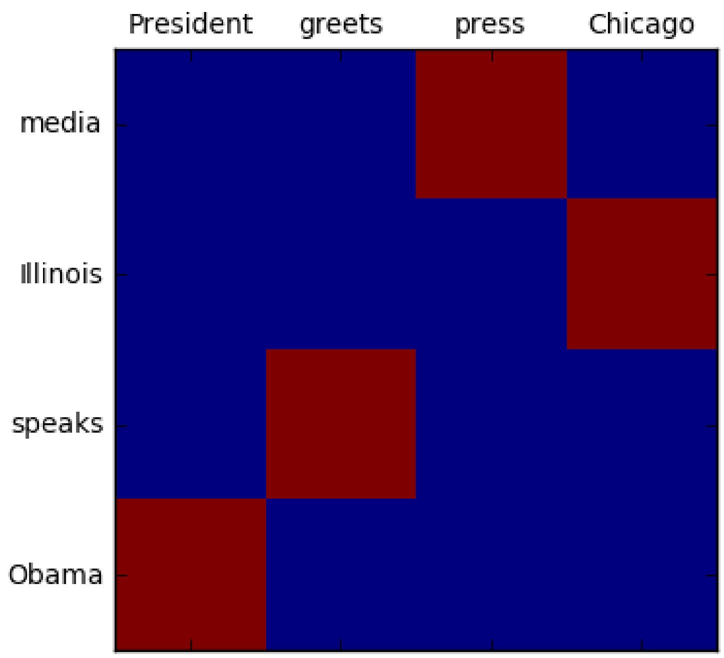


Figure 2: Visualization of transportation matrix with document 1 on y-axis, document 2 on x-axis.

4. We can compute the cosine distance as:

$$\frac{\langle \mu, \nu \rangle}{\|\mu\|_2 \|\nu\|_2}$$

Since there are no words in common between document 1 and 2, $\langle \mu, \nu \rangle = 0$. The cosine distance in this case is zero, which is why Wasserstein + word2vec is much better.

Note: we can also compute the cosine distance as: 1 - cosine similarity:

$$1 - \frac{\langle \mu, \nu \rangle}{\|\mu\|_2 \|\nu\|_2}$$

5. The problem can be formulated a LP. We wish to compute the Wasserstein barycenter a of measures μ and ν:

$$\min_{\substack{a \in \mathbb{R}_+^n, P_1 \in \mathbb{R}_+^{n \times n}, P_2 \in \mathbb{R}_+^{n \times n}, \\ P_1 \mathbb{1}_n = a,\ P_1^T \mathbb{1}_n = \mu, P_2 \mathbb{1}_n = a,\ P_2^T \mathbb{1}_n = \nu}} \quad t\langle P_1, C \rangle + (1-t)\langle P_2, C \rangle$$

Where a is the barycenter of the distributions μ, ν, and P_1 (resp. P_2) is the transportation matrix from a to distribution μ (resp. ν).

6. For code solution, see `barycenter-sols.tex`.

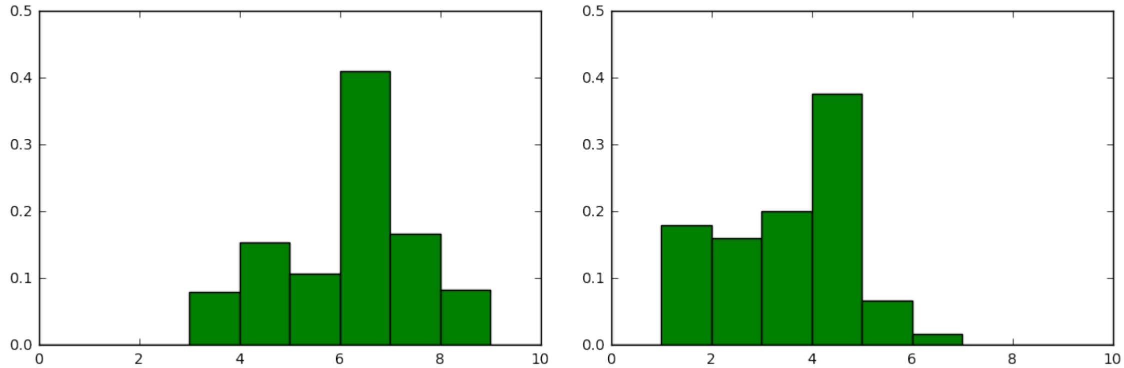


Figure 3: Visualization of a histogram for t=0.75 on left. Visualization of μ histogram for t=0.25 on right.

7. We can see how the mass is literally being transported with the Kantorovich distance while it is just weighted continuously with the euclidean distance.

Exercise 2 (Fast CV for Least-Squares) In this exercise, we consider a regularized least-squares problem:

$$w_\lambda := \arg\min_w \|y - Xw\|_2^2 + \lambda\|w\|_2^2, \tag{1}$$

with $X \in \mathbb{R}^{n\times p}$ the data matrix (with one data point per row), $y \in \mathbb{R}^n$ is the response vector, and $\lambda > 0$ a "ridge" regularization parameter. Solving the above problem leads to a prediction model: for a new data point $x \in \mathbb{R}^p$, $\hat{y}(x) = w_\lambda^\top x$.

We would like to choose this regularization parameter based on the notion of "leave-one-out" (LOO) cross-validation, whereby for a given candidate value of $\lambda > 0$, we estimate the resulting prediction error, averaged across all the models given by the above, when leaving out one data point. Precisely, we set, for $i = 1, \ldots, n$

$$w_\lambda^{(i)} := \arg\min_w \|y_{\setminus i} - X_{\setminus i}w\|_2^2 + \lambda\|w\|_2^2,$$

with $X_{\setminus i}$ (resp. $y_{\setminus i}$) is equal to X (resp. y), with the i-th row (resp. element) removed, and evaluate the prediction error on the point we just left out, with $\hat{y}(x_i) = x_i^\top w_\lambda^{(i)}$.

Obviously, we can compute the LOO error by simply solving n problems of the form (1), with the appropriate data. In this exercise we investigate a faster method, which is based on solving the above full problem *once*, then performing cheap updates to get the LOO error.

1. Show that the solution to the full problem is of the form

$$w_\lambda = (X^\top X + \lambda I)^{-1} X^\top y$$

2. Prove a (simple) version of the Sherman-Morrison-Woodbury identity. Given $M = A + uv^\top$ (with A symmetric/invertible, $A + uv^\top$ also invertible) show that

$$M^{-1} = A^{-1} - \frac{1}{1 + v^\top A^{-1} u}(A^{-1}u)(A^{-1}v)^\top \tag{2}$$

   Namely, it is sufficient to just show that $MM^{-1} = I$ (since showing $M^{-1}M = I$ is similar).

3. Argue both $X_{\setminus i}^\top X_{\setminus i}$ and $X_{\setminus i}^\top y_{\setminus i}$ can be written as rank-one modifications of $X^\top X$ and $X^\top y$ and show that,

$$w_\lambda^{(i)} = w - \frac{\Sigma^{-1} x_i e_i}{1 - h_i}$$

   where we define $\Sigma = X^\top X + \lambda I$, $h_i = x_i^\top \Sigma^{-1} x_i$, and $e_i = y_i - x_i^\top w_\lambda$ for convenience. *Hint: use the Sherman-Morrison-Woodbury identity.*

4. Compute (and simplify) the LOO prediction error $\frac{1}{n}\sum_{i=1}^n (y_i - (w_\lambda^{(i)})^\top x_i)^2$ into an expression consisting of e_i, h_i.

5. What is the complexity of the method for computing the LOO prediction error investigated in this problem relative the naive method where we compute all $w_\lambda^{(i)}$ without reusing computations? Highlight the leading dependencies in terms of n, p for both methods in big-O notation.

Solution 2 1. The first-order stationary conditions give that

$$X^\top(X^\top w_\lambda - y) + \lambda w_\lambda = 0 \implies (X^\top X + \lambda I)w = X^\top y \implies w_\lambda = (X^\top X + \lambda I)^{-1}X^\top y.$$

2. We can directly show $MM^{-1} = I$. We have that,

$$\begin{aligned}
&MM^{-1} = (A + uv^\top)(A^{-1} - \frac{1}{1 + v^\top A^{-1}u}(A^{-1}u)(A^{-1}v)^\top) = \\
&I + uv^\top A^{-1} - \frac{1}{1 + v^\top A^{-1}u}A(A^{-1}u)(A^{-1}v)^\top - \frac{1}{1 + v^\top A^{-1}u}uv^\top(A^{-1}u)(A^{-1}v)^\top = \\
&I + uv^\top A^{-1} - \frac{1}{1 + v^\top A^{-1}u}uv^\top A^{-1} - \frac{v^\top A^{-1}u}{1 + v^\top A^{-1}u}uv^\top A^{-1} = \\
&I
\end{aligned}$$

since the last 3 terms combine to cancel.

3. We can write $X_{\backslash i}^\top X_{\backslash i} = X^\top X - x_i x_i^\top$ and $X_{\backslash i}^\top y_{\backslash i}$ and $X_{\backslash i}^\top y_{\backslash i} = X^\top y - x_i y_i$. Define $\Sigma = X^\top X + \lambda I$. Applying the identity from the previous part we have that,

$$(X_{\backslash i}^\top X_{\backslash i} + \lambda I)^{-1} = (\Sigma - x_i x_i^\top)^{-1} = \left[\Sigma^{-1} + \frac{(\Sigma^{-1}x_i)(x_i^\top \Sigma^{-1})}{1 - x_i^\top \Sigma^{-1} x_i}\right] \tag{3}$$

A short computation shows,

$$\begin{aligned}
w_\lambda^{(i)} &= \left[\Sigma^{-1} + \frac{(\Sigma^{-1}x_i)(x_i^T \Sigma^{-1})}{1 - x_i^\top \Sigma x_i}\right](X^\top y - x_i y_i) \\
&= \Sigma^{-1}X^T y - \Sigma^{-1}x_i y_i + \frac{\Sigma^{-1}x_i}{1 - x_i^\top \Sigma^{-1}x_i}((x_i^T\Sigma^{-1})X^\top y - (x_i^T\Sigma^{-1})x_i y_i) \\
&= w - \Sigma^{-1}x_i y_i + \frac{\Sigma^{-1}x_i}{1 - x_i^\top \Sigma^{-1}x_i}((x_i^T\Sigma^{-1})X^\top y - (x_i^T\Sigma^{-1})x_i y_i) \\
&= w - \Sigma^{-1}x_i y_i + \frac{\Sigma^{-1}x_i}{1 - x_i^\top \Sigma^{-1}x_i}(x_i^\top w - x_i^T\Sigma^{-1}x_i y_i) \\
&= w - \frac{\Sigma^{-1}x_i(1 - h_i)y_i}{1 - h_i} + \frac{\Sigma^{-1}x_i}{1 - h_i}(x_i^\top w - h_i y_i) \\
&= w - \frac{\Sigma^{-1}x_i(\underbrace{y_i - x_i^\top w_i}_{e_i})}{1 - h_i}
\end{aligned}$$

4. Using the previous part, we have that $y_i - (w_\lambda^{(i)})^\top x_i = y_i - (w_\lambda)^\top x_i + \frac{e_i x_i^\top \Sigma^{-1} x_i}{1-h_i} = e_i + \frac{e_i h_i}{1-h_i} = \frac{e_i}{1-h_i}$. Thus the LOO prediction error is:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - (w_\lambda^{(i)})^\top x_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(\frac{e_i}{1-h_i})^2$$

5. The complexity of forming the matrix $X^\top X$ is $O(np^2)$, and inverting it is $O(p^3)$ (and similarly for $X_{\setminus i}^\top X_{\setminus i}$); the other matrix-vector products used in both methods are lower-order in complexity.

   For the naive method we perform this computation n times for each of i datapoints. So the total complexity is $O(np^3 + n^2p^2)$.

   For the fast method we only construct and invert the matrix $X^\top X$ once. The total complexity (including computing the sum) we can see is $O(p^3 + np^2)$.

Exercise 3 (Least norm estimation on traffic flow networks) In this problem, we want to estimate the traffic given the road network as well as the historical average of flows on each road segment. We call q_i the flow of vehicles on each road segment $i \in I$. At each intersection, the sum of all incoming flows must be equal to the sum of all outgoing flows. We construct the matrix $A \in \mathbb{R}^{J \times I}$ such that the element on the jth line and ith column is

- 0 if link i does not arrive or leave intersection j;
- 1 if link i arrives at intersection j;
- -1 if link i leaves intersection j.

1. Write down the linear equation that corresponds to the conservation of vehicles at each intersection $j \in J$.
2. The goal is to estimate the traffic flow on each of the road segment. The flow estimates should satisfy the conservation of vehicles exactly at each intersection. Among the solutions that satisfy this constraint, we are searching for the estimate that is the closest to the historical average, $\overline{q}$, in the l_2-norm sense. The vector $\overline{q}$ has size I and the i-th element represent the average for the road segment i. Pose the optimization problem.
3. Find a closed form solution to this problem. Detail your answer (do not only give a formula but explain where it comes from).

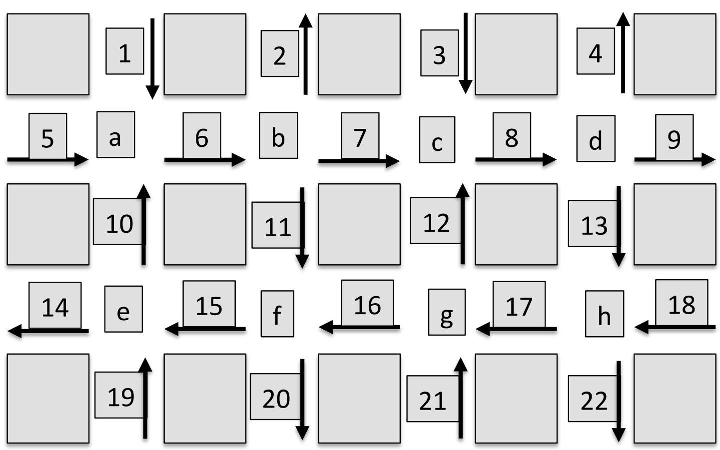


Figure 4: Example of traffic estimation problem. The intersections are labeled a to h. The road segments are labeled 1 to 22. The arrows indicate the direction of traffic.

4. Formulate the problem for the small example of Figure 4 and solve it using the historical average given in Table 1. What is the flow that you estimate on road segments 1, 3, 6, 15 and 22?

segment	average	measured
1	2047.6	2028
2	2046.0	2008
3	2002.6	2035
4	2036.9	
5	2013.5	2019
6	2021.1	
7	2027.4	
8	2047.1	
9	2020.9	2044
10	2049.2	
11	2015.1	
12	2035.1	
13	2033.3	
14	2027.0	2043
15	2034.9	
16	2033.3	
17	2008.9	
18	2006.4	
19	2050.0	2030
20	2008.6	2025
21	2001.6	
22	2028.1	2045

Table 1: Table of flows: historical averages $\overline{q}$ (center column), and some measured flows (right column).

5. Now, assume that besides the historical averages, you are also given some flow measurements on some of the road segments of the network. You assume that these flow measurements are correct and want your estimate of the flow to match these measurements perfectly (besides matching the conservation of vehicles of course). The right column of Table 1 lists the road segments for which we have such flow measurements. Do you estimate a different flow on some of the links? Give the difference in flow you estimate for road segments 1,3, 6, 15 and 22. Also check that you estimate gives you the measured flow on the road segments for which you have measured the flow. *Hint:* Your solution will comment on the feasibility of solving such a problem.

Solution 3

For this problem, we also have python solution code. See jupyter notebook `traffic_flow_sols.ipynb`

1. With such a construction, the conservation of flow is written $Aq = 0$.

2. The problem can be formulated as follows:

$$\min_{q \in \mathbb{R}^I} \quad ||q - \overline{q}||_2 \text{ subject to } Aq = 0.$$

3. To solve this problem we introduce the variable $x = q - \overline{q}$. With this change of variable, we solve the following optimization problem:

$$\min_{x \in \mathbb{R}^I} \quad ||x||_2 \text{ subject to } Ax = b,$$

   where $b = -A\overline{q}$. Our estimation problem is a least norm problem. The matrix A is full row rank, that is, the matrix AA^T is invertible. We can solve the minimum norm problem in closed-form $x^* = A^T(AA^T)^{-1}b$. We get the flow estimate as $q^* = x^* + \overline{q}$.

4. We first need a function that constructs the matrix A that will represent the incidence matrix of links at intersections.

```
function A = construct_network
A = zeros(8,22);
for i=1:4
A(i,i) = -(-1)^i;
A(i,4+i) = 1;
A(i,5+i) = -1;
A(i,9+i) = -(-1)^i;
end
for i=5:8
A(i,9+i-4) = (-1)^i;
A(i,13+i-4) = 1;
A(i,14+i-4) = -1;
A(i,18+i-4) = -(-1)^i;
end
```

   Then, we solve the estimation problem as a least norm problem. The function below loads the historical flows, constructs the incidence matrix and estimates the flow that is closest to the historical averages while satisfying the conservation of vehicles.

```
function x_hat = estimate_historic
```

```
A = construct_network;
hist = load('historical.mat');
hist = hist.historical;

b = zeros(8,1);

z_hat = A\(b-A*hist);
x_hat = z_hat + hist;
```

5. The additional flow measurements are incorporated into the constraints. Let B be the matrix of size N times I where N represents the number of flow measurements. This matrix has a one on row n and column i if the nth measurement concerns link i. It has zeros everywhere else. We call q_m the vector of measurements. The optimization problem reads:

$$\min_{q\in\mathbb{R}^I} \quad ||q-\overline{q}||_2 \text{ subject to } Aq=0,\;\; Bq=q_m.$$

With the same change of variable as before ($x = q-\overline{q}$), defining $\tilde{A}$ the matrix obtained by stacking A and B vertically, $\tilde{A} = [A^T\ B^T]^T$ and $\tilde{b}$ the vector obtained by stacking b and $q_m - B\overline{q}$, $\tilde{b} = [b^T\ (q_m - B\overline{q})^T]^T$, we solve again a least norm problem:

$$\min_{x\in\mathbb{R}^I} \quad ||x||_2 \text{ subject to } \tilde{A}x = \tilde{b}.$$

In the data provided, the measurements are such that $\tilde{A}$ is still full row rank, and we can solve the problem with the same method as before. In the general case, $\tilde{A}$ may not be full row rank anymore. In this case, two situations may arise:

- If $\tilde{b}$ is not in the range of $\tilde{A}$, then the problem is not feasible (the optimal value is $+\infty$).
- Otherwise, any solution of the linear equation $\tilde{A}x = \tilde{b}$ can be written $x = x^* + z$. Here, z is any element of the nullspace of $\tilde{A}$. Also, x^* is *the* element in the orthogonal of the nullspace which is solution to the linear equation *i.e.* $\tilde{A}x^* = \tilde{b}$ and for all z in the nullspace of $\tilde{A}$, $z^Tx^* = 0$.

Exercise 4 (A Portfolio Design Problem) The returns on $n = 4$ assets are described by a Gaussian (normal) random vector $r \in \mathbb{R}^n$, having the following expected value $\hat{r}$ and covariance matrix Σ:

$$\hat{r} = \begin{bmatrix} 0.12 \\ 0.10 \\ 0.07 \\ 0.03 \end{bmatrix}, \; \Sigma = \begin{bmatrix} 0.0064 & 0.0008 & -0.0011 & 0 \\ 0.0008 & 0.0025 & 0 & 0 \\ -0.0011 & 0 & 0.0004 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The last (fourth) asset corresponds to a risk-free investment. An investor wants to design a portfolio mix with weights $x \in \mathbb{R}^n$ (each weight x_i is non-negative, and the sum of the weights is one) so as to obtain the best possible expected return $\hat{r}^T x$, while guaranteeing that:

(i) No single asset weights more than 40%
(ii) The risk-free assets should not weight more than 20%
(iii) No asset should weight less than 5%
(iv) The probability of experiencing a return lower than $q = -3\%$ should be no larger than $\epsilon = 10^{-4}$

1. What is the maximal achievable expected return, under the above constraints?

   Hint: Constraint (iv) is known as a "chance constraint."

   $$a \sim N(\hat{a}, \Sigma) \implies a^T x - b \sim N(\hat{a}^T x - b, x^T \Sigma x)$$

   We then have:

   $$\Pr(a^T x \leq b) \geq \eta \iff b - \hat{a}^T x \geq \Phi^{-1}(\eta) ||\Sigma^{1/2} x||_2$$

2. Solve the problem for a large number of values of ϵ between 10^{-4} and 10^{-1}, and plot the optimal values of $\hat{r}^T x$ versus ϵ. Also make an area plot of the optimal portfolios x versus ϵ.

3. *Monte Carlo simulation.* Let x be the optimal portfolio found in part 1, with $\epsilon = 10^4$. This portfolio maximizes the expected return, subject to the probability of a loss being no more than 3 %. Generate 1000 samples of r, and plot a histogram of the returns. Find the empirical mean of the return samples, and calculate the percentage of samples for which a loss occurs.

Solution 4

Each of the given constraints can be converted into a mathematical constraint

1. Since each x_i represents the weight of each asset, the constraint is that

$$\forall i \; x_i \leq 0.4$$

The risk free asset is the fourth one, hence $x_4 \leq 0.2$

All weights must be at least 5%:

$$\forall i \; x_i \geq 0.05$$

The return is $r^T x$, where $r \sim N(\hat{r}, \Sigma)$

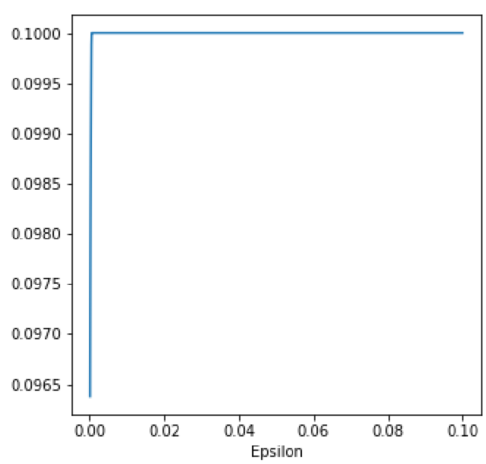
A way of writing the last constraint is: $Pr(\bar{r}^T x \leq -0.03) \leq 10^{-4}$.
So we have now a chance constraint: $\hat{r}^T x + \Phi^{-1}(10^{-4})||\Sigma^{1/2} x||_2 \geq -0.03$. This is the constraint that characterizes the problem as an SOCP rather than an LP.

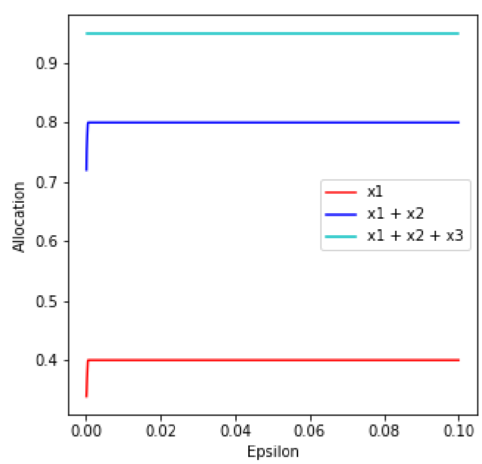
We see that the optimal allocation of weights is:

$$x = \begin{bmatrix} 0.34 \\ 0.38 \\ 0.23 \\ 0.05 \end{bmatrix}$$

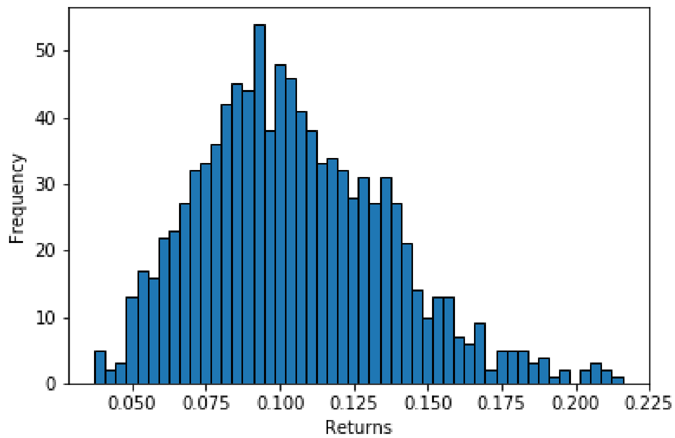
and the optimal return $\hat{r}^T x = 0.1$

2. Here we see that the more risk-averse one is, the average returns are less compared to if one takes more risk.





3. The empirical mean calculated was 0.107 and 1.1% of the samples generated a loss (return was < 0).



Exercise 5 (A slalom problem) A two-dimensional skier must slalom down a slope, by going through n parallel gates of known position (x_i, y_i), and of width c_i, $i = 1, \ldots, n$. The initial position (x_0, y_0) is given, as well as the final one, (x_{n+1}, y_{n+1}). Here, the x-axis represents the direction down the slope, from left to right, see Figure 5.

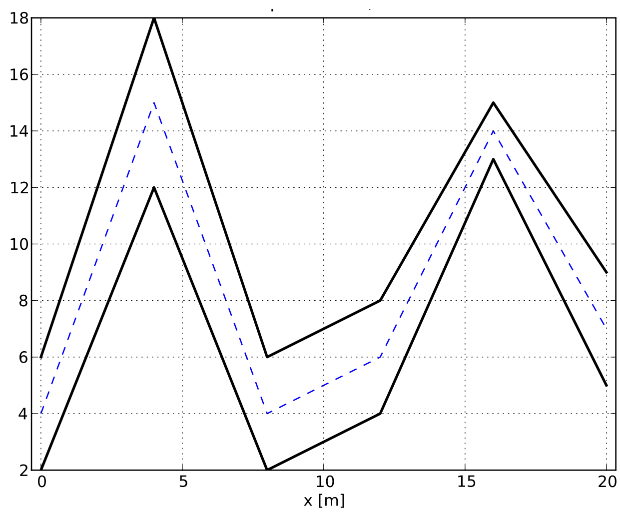


Figure 5: Slalom problem with $n = 5$ obstacles. "Uphill" (resp. "downhill") is on the left (resp. right) side. The middle path is dashed, initial and final positions are not shown.

i	x_i	y_i	c_i
0	0	4	N/A
1	4	5	3
2	8	4	2
3	12	6	2
4	16	5	1
5	20	7	2
6	24	4	N/A

Table 2: Problem data for Exercise 5.

1. Find the path that minimizes the total length of the path. Your answer should come in the form of an optimization problem.

2. Solve the problem numerically, with the data given in Table 2.

Solution 5

1. Assume that (x_i, z_i) is the crossing point of gate i, the path length minimization problem is thus

$$\begin{array}{ll} \min\limits_{z} & \sum\limits_{i=1}^{n+1} \left\| \begin{pmatrix} x_i \\ z_i \end{pmatrix} - \begin{pmatrix} x_{i-1} \\ z_{i-1} \end{pmatrix} \right\|_2 \\ \text{subject to} & y_i - c_i/2 \le z_i \le y_i + c_i/2, \text{ for } i = 1, \ldots, n \\ & z_0 = y_0, z_{n+1} = y_{n+1}, \end{array}$$

   which is equivalent to

$$\begin{array}{ll} \min\limits_{z} & \sum\limits_{i=1}^{n+1} t_i \\ \text{subject to} & y_i - c_i/2 \le z_i \le y_i + c_i/2, \text{ for } i = 0, \ldots, n+1 \\ & \left\| \begin{pmatrix} x_i \\ z_i \end{pmatrix} - \begin{pmatrix} x_{i-1} \\ z_{i-1} \end{pmatrix} \right\|_2 \le t_i, \text{ for } i = 1, \ldots, n+1. \end{array}$$

   with the convention $c_0 = c_{n+1} = 0$. Hence, the problem is an SOCP.

2. A CVX code for the problem is

```
x = [0 4 8 12 16 20 24]';
y = [4 5 4 6 5 7 4]';
c = [0 3 2 2 1 2 0]';
n = 5;
cvx_begin
    variable z(n+2);
    variable t(n+1);
    minimize sum(t)
    subject to
        z <= y + c/2
        z >= y - c/2
        for i=1:n+1
            norm([x(i+1); z(i+1)] - [x(i); z(i)]) <= t(i)
        end
cvx_end
z = [4.0000 4.3749 4.7498 5.1248 5.5000 6.000 4.0000];
```