EE5121
CVX Assignment Report

Varun Sundar
ee16b068

Indian Institute of Technology Madras

1 Implementation Details

- All questions have been done in **python 3.6**
- Run as : **python q_x.py**
- Install packages as: **pip install -r requirements.txt**
- All plots are saved to the **logs** directory.

2 Q1

We formulate the problem as follows:

$$\min_{x} \quad \min ||y - x||^2 \tag{1a}$$
$$\text{subject to} \quad ||Ax||_1 <= b. \tag{1b}$$

This is not in the standard SOCP form. In order to recast it appropriately, we modify it as: (mainly in the epigraph form)

$$\min_{t} \quad t \tag{2a}$$
$$\text{subject to} \quad ||y - x||^2 < t^2, \tag{2b}$$
$$||a_j^T x||_1 <= t_i \quad \forall i \in \{1, 2, ..., n-1\}, \tag{2c}$$
$$\sum_{i=1}^{n-1} t_i <= b. \tag{2d}$$

We tune b such that only 20 jumps are observed in the resultant estimate. (Notice that $b = 20$ is an overestimate of the constraint, since the L1 convex relaxation is a slack approximation to L0).

Here, A is the matrix as indicated in the question's hint. We observe the following plot with $b = 5.3825$. The error is around 15.211 and the L1 value is around b. Note that as b decreases, the error increases. Hence, there is a trade-off between MSE and jumps in the signal (very much similar to the bias-variance trade-off). So we choose the maximum b which preserves 20 jumps. We

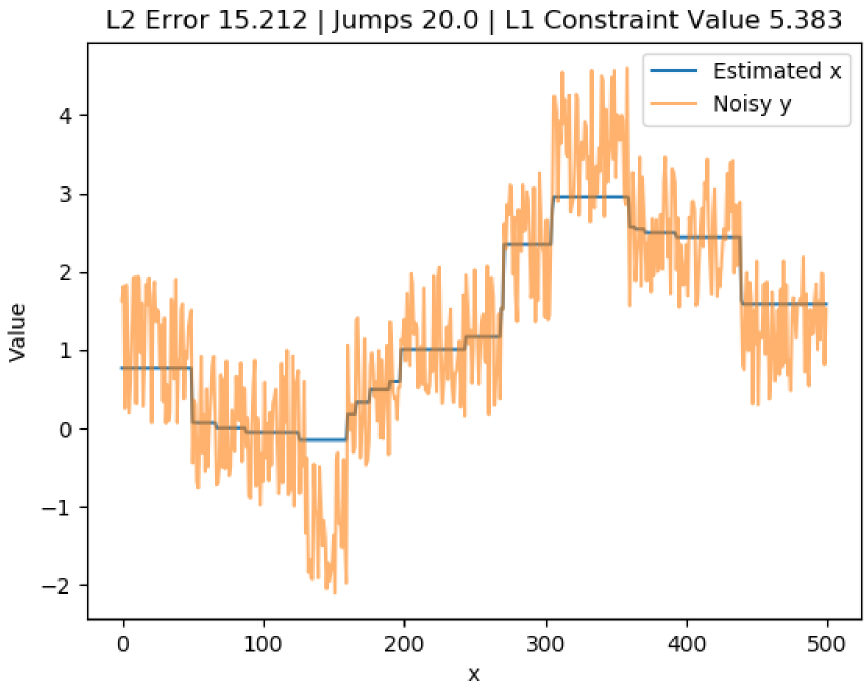

Figure 1: Plot of $\hat{x}$ and noisy y. We have indicated the error and L1 constraint value.

Figure 2: Screen-shot of question 1

arrive at our chosen value of b via a binary grid-search.

Notice the fact that computing jumps visually is slightly ambiguous, the ”jumps” are of order 0.8. So we do this by $||Ax||_0$, using numpy, an d consider anything > 0.01 as a jump (ie... tolerance is 0.01).

3 Q2

We formulate the following LP:

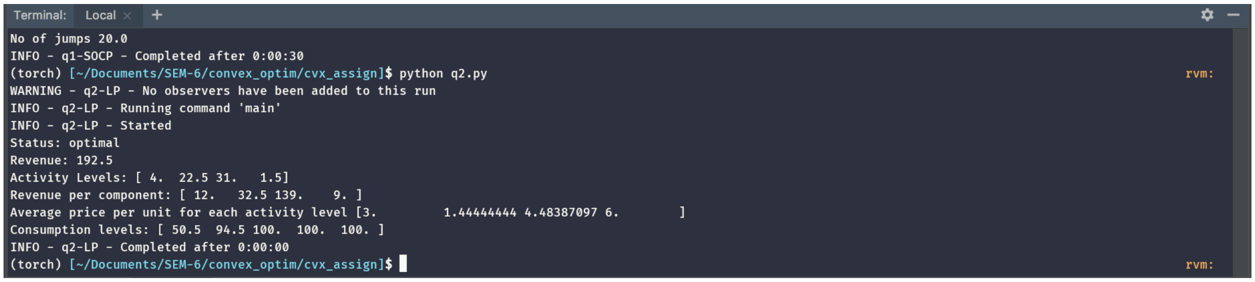

Figure 3: Screen-shot of question 2

$$
\begin{aligned}
\min_{t,x} \quad & 1^T t && \text{(3a)}\\
\text{subject to} \quad & t_j \le p_j * x_j \quad \forall j \in \{1,2,3,4\}, && \text{(3b)}\\
& t_j \le p_j * q_j + p_{disc} * (x_j - q_j) \quad \forall j \in \{1,2,3,4\}, && \text{(3c)}\\
& Ax \le c_{max}, && \text{(3d)}\\
& 0 \le x. && \text{(3e)}
\end{aligned}
$$

We obtain the following optimal values:

- Revenue: 192.5
- Activity Levels: $[4, 22.5, 31., 1.5]$
- Revenue per component: $[12., 32.5, 139., 9.]$
- Average price per unit for each activity level: $[3., 1.44444444, 4.48387097, 6.]$

We make the following observations:

- Consumption levels are met for $c_{3,4,5}$.
- Discounted price is used for $x_{2,3}$. By noting the average price per unit activity.
- Activity levels cross the threshold for $x_{2,3}$.
- Further, revenue per component is max for $x_{2,3}$.

4 Q3

With the convex relaxation of rank and the lemma provided to constrain rank, we obtain the following SDP formulation:

Figure 4: Screen-shot of question 3

$$
\begin{aligned}
\min_{r,\hat{X},Y,Z} \quad & r && \text{(4a)}\\
\text{subject to} \quad & tr(Y) + tr(Z) \leq 2r && \text{(4b)}\\
& \hat{X}_{i,j} = X_{i,j} \quad \forall i,j \in J \quad \text{set of non-zero entries of } X, && \text{(4c)}\\
& \begin{pmatrix} Y & X \\ X^T & Z \end{pmatrix} \geq 0, && \text{(4d)}\\
& Y \geq 0. && \text{(4e)}\\
& Z \geq 0. && \text{(4f)}
\end{aligned}
$$

We obtain the rank to be 19, by using **np.linalg.matrix_rank** by thresholding to $1e-6$.

Note that the optimisation formulation is cast in the epigraph form - we may also eliminate r entirely and recast as:

$$
\begin{aligned}
\min_{\hat{X},Y,Z} \quad & tr(Y) + tr(Z) && \text{(5a)}\\
& \hat{X}_{i,j} = X_{i,j} \quad \forall i,j \in J \quad \text{set of non-zero entries of } X, && \text{(5b)}\\
& \begin{pmatrix} Y & X \\ X^T & Z \end{pmatrix} \geq 0, && \text{(5c)}\\
& Y \geq 0. && \text{(5d)}\\
& Z \geq 0. && \text{(5e)}
\end{aligned}
$$