EE-588: Homework # 4

Due on Wednesday, November 6, 2019

Saket Choudhary
2170058637

Contents

6.2

$$\text{minimize } ||x\mathbf{1} - b||$$

l_1 norm:

$$\text{minimize } ||x\mathbf{1} - b||$$

is equivalent to

$$L(x) = \sum_{i=1}^{n} |x - b_i|$$
$$\nabla L(x) = \sum_{i=1}^{n} \text{sign}(x - b_i)$$
$$\text{sign}(x - b_i) = 1 \quad \text{when } x > b_i$$
$$\text{sign}(x - b_i) = -1 \quad \text{when } x < b_i$$

$\nabla L(x) = 0$ when the number of positive and negative signs are equal which is possible when $\boxed{x = \text{ median } (b_1, b_2 \dots b_n)}$.

l_2 case:

$$L(x) = ||x\mathbf{1} - b||_2^2$$
$$\nabla L(x) = \mathbf{1}^T x\mathbf{1} - \mathbf{1}^T b$$
$$\nabla L(x) = 0$$
$$\implies \boxed{x = \mathbf{1}^T b/m}$$

l_∞ case:

$$L(x) = \text{minimize } \max |x - b_i|$$

equivalent to the LP:

$$\text{minimize } t$$
$$\text{such that } \mathbf{1}x - b \leq t$$
$$\mathbf{1}x - b \geq -t$$

Since $-t \leq x - b_i \leq t$ and we want to minimize t, $t > x - b_{(1)}$ and $t \geq b_{(n)} - x \implies t \geq \frac{b_{(n)} - b_{(1)}}{2}$ where $b_{(1)} = \min b_i$ and $b_{(n)} = \max b_i$. Thus, $\boxed{x = \frac{1}{2} \text{ range } (b_1, b_2, \dots, b_n) = \frac{b_{(n)} - b_{(1)}}{2}}$

6.9

$$\text{minimize } \max_{i=1,\ldots,k} \left|\frac{p(t_i)}{q(t_i)} - y_i\right|$$

As $q(t_i) > 0$,

$$\max_{i=1,\ldots,k} |p(t_i) - yq(t_i)| \leq sq(t_i)$$

iff,

$$-sq(t_i) \leq |p(t_i) - yq(t_i)| \leq sq(t_i)$$

Since the domain of the rational function is convex and $-sq(t_i) \leq |p(t_i) - yq(t_i)| \leq sq(t_i)$ is linear iequality, the rational function is quasiconvex.

7.3

Likelhood is given by:

$$l(a,b) = \prod_{y_i=1} P(v_i \leq -a^T u_i - b) \prod_{y_i=0} P(v_i \geq -a^T u_i - b)$$

Define $\phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$. Then,

$$\begin{aligned} P(v_i \leq -a^T u_i - b) &= \phi(-a^T u_i - b) \\ P(v_i \geq -a^T u_i - b) &= 1 - \phi(-a^T u_i - b) \\ &= \phi(a^T u_i + b) \end{aligned}$$

And hence log likelihood function $L(a,b)$ is given by,

$$L(a,b) = \sum_{i=1}^{n} \phi(-a^T u_i - b) + \sum_{i=1}^{n} \phi(a^T u_i + b)$$

$L(a,b)$ is a concave function as $\phi(z)$ is concave and z is affine in a,b.

7.8

Similar to problem 7.3, the log likelihood is given by:

$$
\begin{aligned}
L(a,b) &= \sum_{y_i=-1} \log P(v_i \leq -a_i^T x - b_i) + \sum_{y_i=1} \log P(v_i \geq -a_i^T x - b_i) \\
&= \sum_{y_i=-1} \log F(-a_i^T x - b_i) + \sum_{y_i=1} \log(1 - F(-a_i^T x - b_i))
\end{aligned}
$$

where $F(z)$ represents the cumulative distribution fucntion of the log-concave probability density $P(z)$
To maximize $L(a,b)$ is equivalent to maximizing $F(z)$ and its affine which are both log-concave and hence the problem is convex.

Total variation image interpolation

```
Ul2 = ones(m, n);
Utv = ones(m, n);
cvx_begin
  variable Ul2(m,n);
  Ul2(Known) == Uorig(Known);
  dist1 = Ul2(2:m,2:m)-Ul2(1:m-1,2:m);
  dist2 = Ul2(2:m,2:m)-Ul2(2:m,1:m-1);
  % vectorize
  minimize(norm([dist1(:); dist2(:)],2));
cvx_end


cvx_begin
  variable Utv(m,n);
  % Ensure known are equal
  Utv(Known) == Uorig(Known);
  dist1 = Utv(2:m,2:m)-Utv(1:m-1,2:m);
  dist2 = Utv(2:m,2:m)-Utv(2:m,1:m-1);
  % vectorize
  minimize(norm([dist1(:); dist2(:)],1));
cvx_end
```

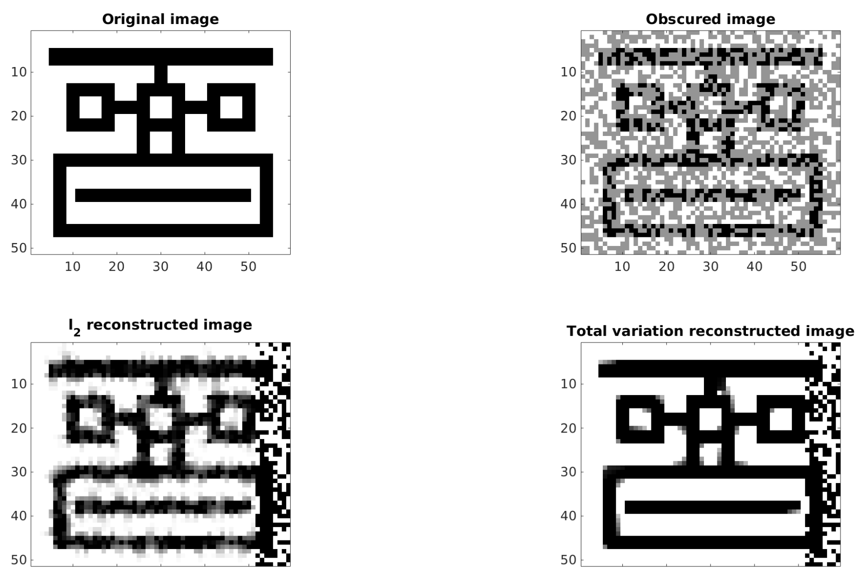

Figure 1: Output of `tv_img_interp.m`

Piecewise Linear Filtering

To ensure convexity, we need: $\alpha_1 \leq \alpha_2 \leq \alpha_2 \dots \alpha_K$ To ensure continuity at the knot points, we need: $\alpha_i a_i + \beta_i = \alpha_{i+1}a_{i+1} + \beta_{i+1} \quad \forall i \in [1, K-1]$
The optimization problem is now equivalent to:

$$\begin{aligned}
\text{minimize}(\sum_{i=1}^{m} f(x_i) - y_i)^2 \\
\text{subject to} \alpha_i \leq \alpha_{i+1} \quad i \in [1, K-1] \\
\alpha_i a_i + \beta_i = \alpha_{i+1}a_{i+1} + \beta_{i+1} \quad i \in [1, K-1] \\
f(x_i) = \alpha_i x_i + \beta_i \quad i \in [1, K-1]
\end{aligned}$$

In the matrix form:

$$\begin{aligned}
\text{minimize}||\text{diag}(X)F_{ij}\alpha + \mathbf{y} - \mathbf{y}||_2^2 \\
\text{subject to} \alpha_i \leq \alpha_{i+1} \quad i \in [1, K-1] \\
\alpha_i a_i + \beta_i = \alpha_{i+1}a_{i+1} + \beta_{i+1} \quad i \in [1, K-1] \\
f(x_i) = \alpha_i x_i + \beta_i \quad i \in [1, K-1]
\end{aligned}$$

Since f is langragian, we have:

$$f_{ij} = \begin{cases} 1 & x_i = a_0 \\ 1 & \text{if } a_{j-1} \leq x \leq a_j \\ 0 & \text{otherwise} \end{cases}$$

$\alpha \in \mathbb{R}^k$, $\beta \in \mathbb{R}^k$
and $\mathbf{x} \in \mathbb{R}^k$, $\mathbf{x} \in \mathbb{R}^k$

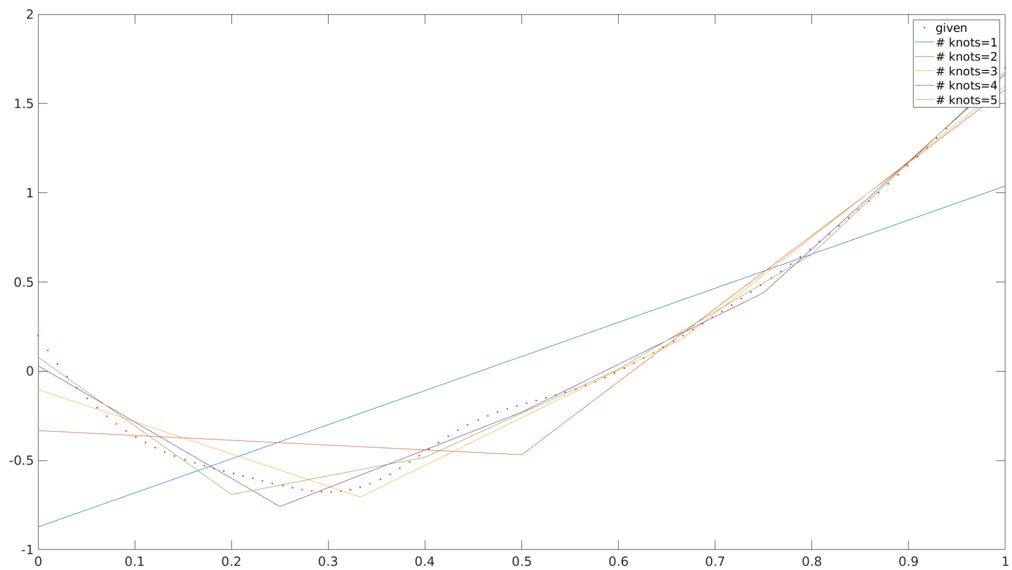

Figure 2: Comparison of different Knots (1-4)

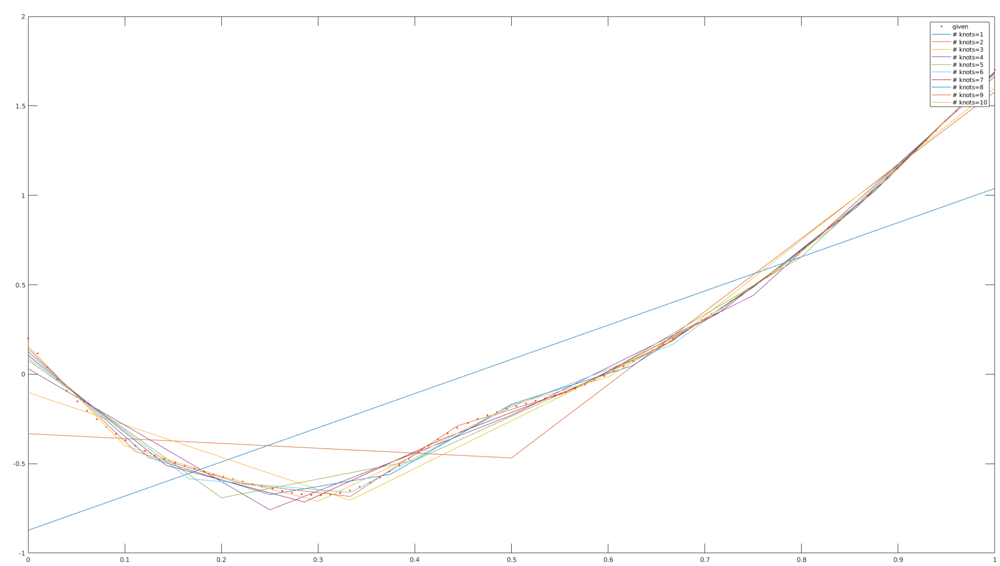

Figure 3: Comparison of different Knots (1-10)

```
pwl_fit_data;
m = length(x);


%knots = [1 2 3 4 5 6 7 8 9 10];
knots = [1 2 3 4 5];

x_plot = 0:0.0001:1;

m_plot = length(x_plot);

y_plot = [];
figure('DefaultAxesFontSize',20);
plot(x, y, 'r.', 'DisplayName', 'given');
hold on;
for k = knots
  a = [0:1/k:1]';

  %S = sparse(i,j,s,m,n,nzmax) uses vectors i, j, and s to generate an
  %  m-by-n sparse matrix such that S(i(k),j(k)) = s(k), with space
  %  allocated for nzmax nonzeros.  Vectors i, j, and s are all the same
  %  length.

  F = sparse(1:m, max(1, ceil(x*k)), 1, m, k);
  cvx_begin
    variables myalpha(k) mybeta(k);
    minimize(norm(diag(x)*F*myalpha + F*mybeta-y, 2))
    subject to
    if (k>=2)
        myalpha(1:k-1).*a(2:k) + mybeta(1:k-1) == myalpha(2:k).*a(2:k) + mybeta(2:k)
        a(1:k-1) <= a(2:k)
    end
  cvx_end

  F_plot = sparse(1:m_plot, max(1, ceil(x_plot*k)), 1, m_plot, k);
  y_temp = diag(x_plot)*F_plot*myalpha + F_plot*mybeta;
  y_plot = [y_plot y_temp];
  plot(x_plot, y_temp, 'DisplayName', strcat('# knots=', int2str(k)));
  hold on;
end

hold off;
legend;
```