Robust Low-Rank Matrix Recovery

Chapter Intended Learning Outcomes:

(i) Understand the limitation of singular value decomposition (SVD) in low-rank matrix approximation and completion applications

(ii) Know the use of ℓ_p-norm in enhancing robustness to outliers

(iii) Able to solve problems with nonsmooth and/or nonconvex functions

(iv) Able to apply robust matrix decomposition in relevant real-world applications

Limitation of Singular Value Decomposition

Recall that truncated SVD of a given matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ is to find the best rank-r matrix $\boldsymbol{X}_s$ in the least squares (LS) sense:

$$\boldsymbol{X}_s = \arg\min_{\tilde{\boldsymbol{X}}} \left\| \boldsymbol{X} - \tilde{\boldsymbol{X}} \right\|_F^2, \quad \text{s.t.} \quad \text{rank}(\tilde{\boldsymbol{X}}) = r$$

That is, $\boldsymbol{X}_s$ is computed by minimizing the sum of all mn components of $(\tilde{\boldsymbol{x}}_{i,j} - \boldsymbol{x}_{i,j})^2$ subject to the rank constraint.

If some entries of mn contain outliers or abnormal values, e.g., values with large magnitudes, then $\boldsymbol{X}_s$ may not be able to accurately represent the low-rank component.

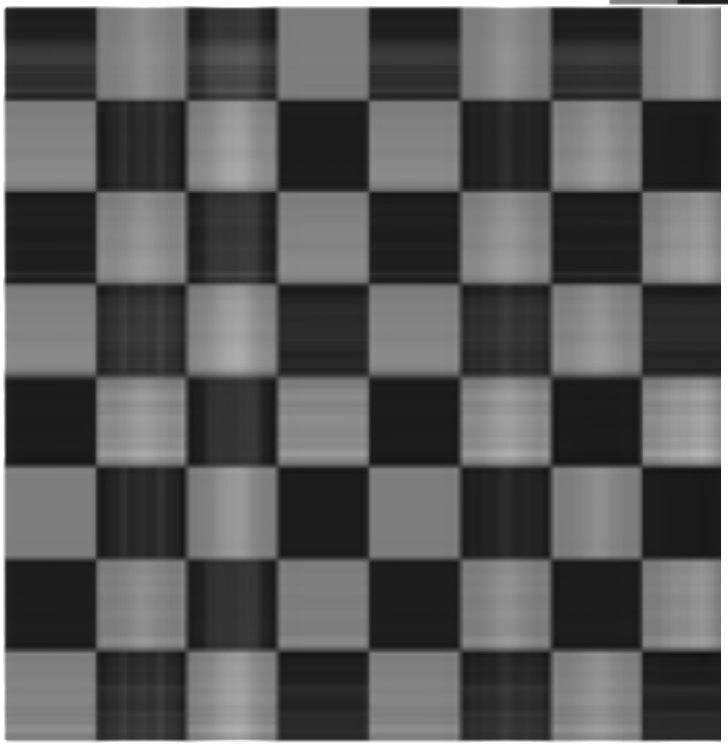
Example 1

Extract chessboard from chess pieces using truncated SVD.

Here $\boldsymbol{X} \in \mathbb{R}^{461 \times 449}$. The regular structure of the chessboard corresponds to a low-rank matrix and we set $r = 2$. However, the result is not satisfactory.

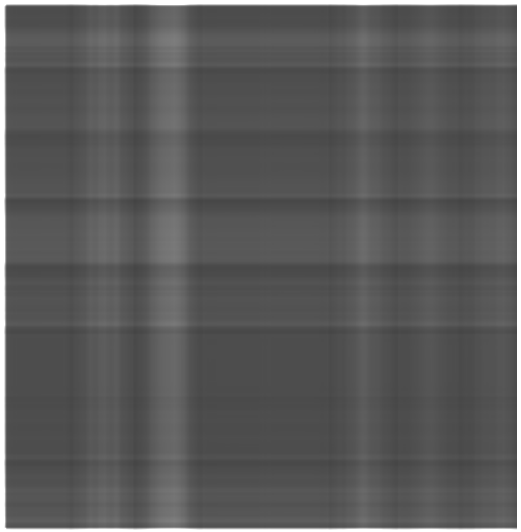
It is because the chess pieces correspond to outlier components, and the LS method cannot yield a good solution. Also, the chess pieces cannot be well extracted from $\boldsymbol{X} - \boldsymbol{X}_s$.

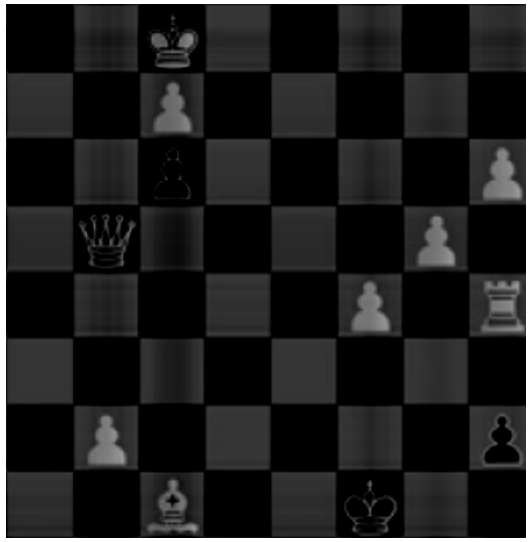
```
>> img = imread('board_chess.png');
  combined = im2double(rgb2gray(img));
  [U,S,V] = svd(combined,'econ');
  r=2;
  board = U(:,1:r)*S(1:r,1:r)*V(:,1:r).';
  imwrite(board, 'board.png', 'png');
  chess = combined-board;
  imwrite(chess, 'chess.png','png');
```



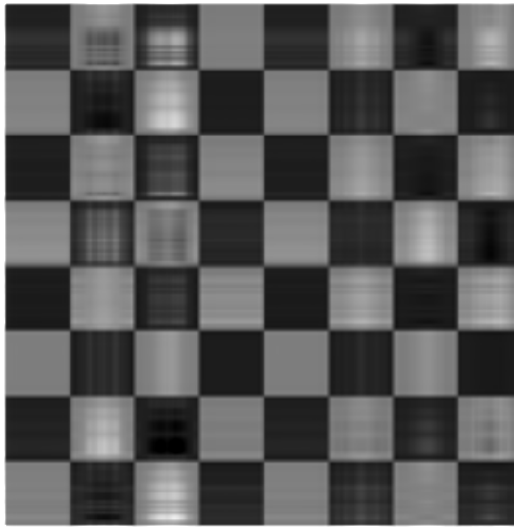
The choice of rank is crucial and $r = 2$ is the best.

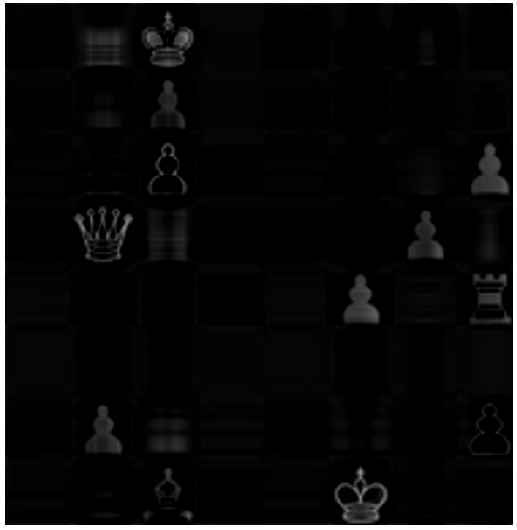
$r = 1$:





$r = 3$:

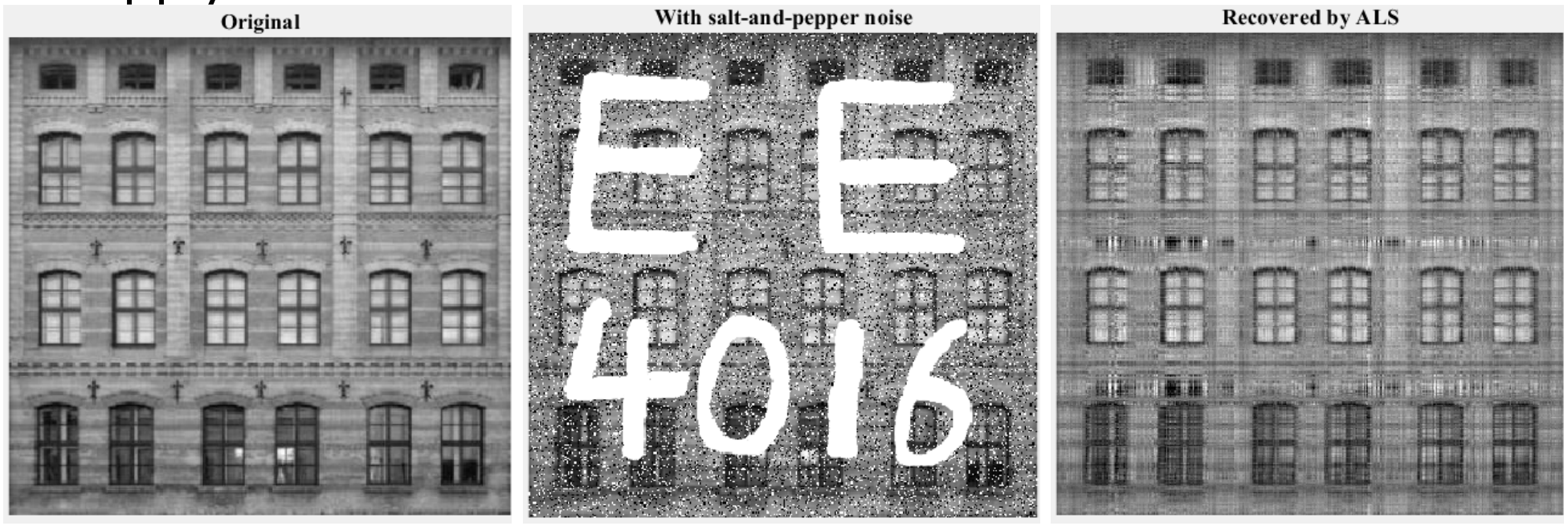




Example 2

Perform image inpainting in salt-and-pepper noise. Note that this noise is sometimes seen on images, which presents itself as sparsely occurring white and black pixels, with values 255 and 0 in 8-bit integer representation, respectively.

We apply ALS with rank $r = 7$:



Unsatisfactory performance is obtained because the LS criterion is not robust to outliers.

A simple idea to deal with the outliers is to use ℓ_p-norm where $0 < p < 2$. The ℓ_p-norm criterion generalizes the LS by setting $p = 2$.

The ℓ_p-norm of a vector $\boldsymbol{x} = [x_1 \ \cdots \ x_N]$ is defined as:

$$||\boldsymbol{x}||_p = (|x_1|^p + \ \cdots \ + |x_N|^p)^{1/p} = \left(\sum_{n=1}^{N} |x_n|^p \right)^{1/p} \tag{1}$$

Note that (1) can also be used for $p = 0$ and $p > 2$.

The special cases include

$$||\boldsymbol{x}||_2 = (|x_1|^2 + \ \cdots \ + |x_N|^2)^{1/2}$$

$$||\boldsymbol{x}||_1 = (|x_1| + \ \cdots \ + |x_N|)$$

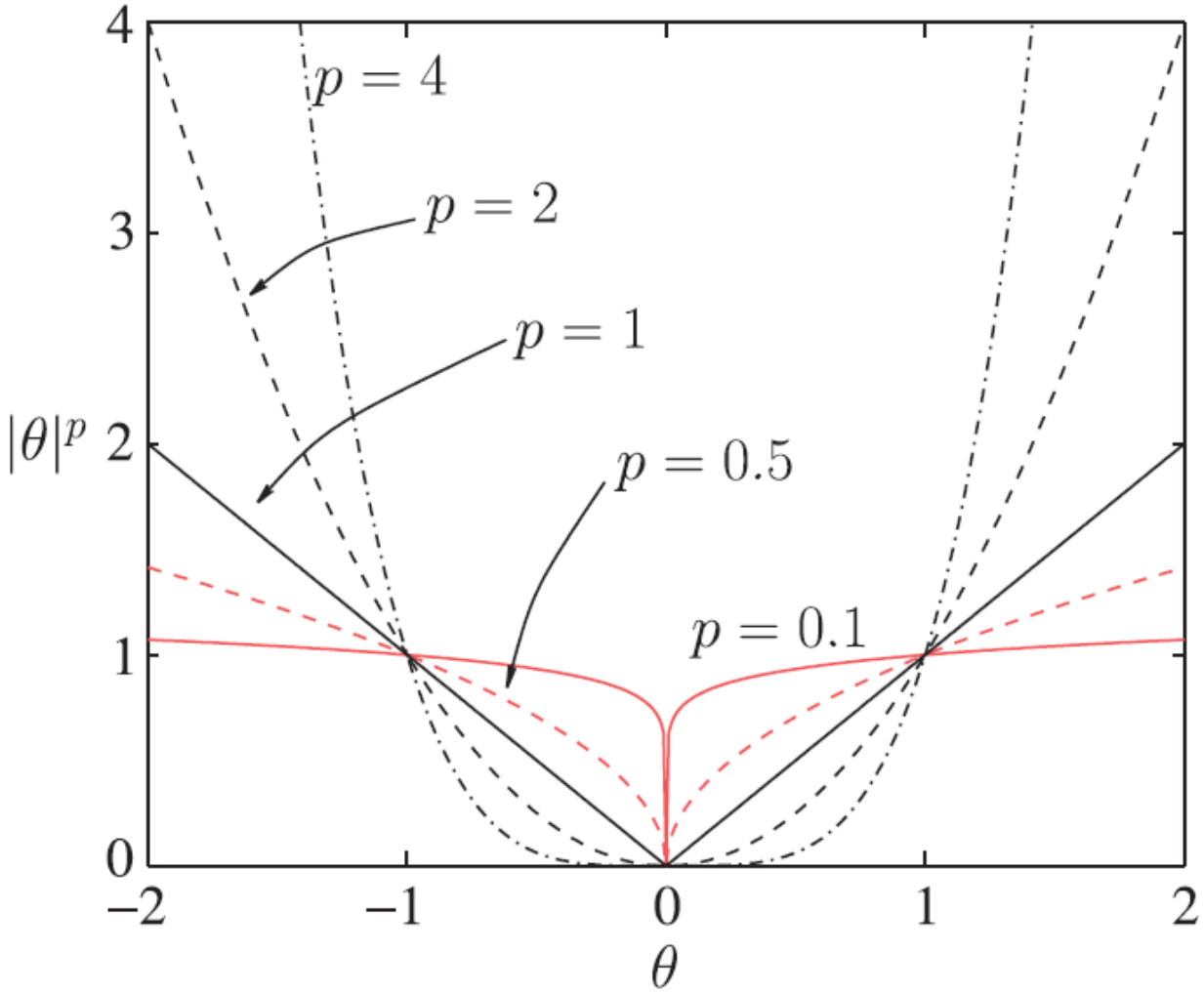
$$||\boldsymbol{x}||_\infty = \max\{|x_1|, \ \cdots \ |x_N|\}$$

$||\boldsymbol{x}||_0$ is the number of nonzero elements in $\boldsymbol{x}$.

As the pth power of an error term (if >1 in magnitude) is less than the squared error, hence the ℓ_p-norm with $0 < p < 2$ is less sensitive to outliers or large-magnitude components.

While in the ℓ_0-norm, a small increase of a component from 0 makes its contribution to the norm large (close to 1 for any values other than 0). Hence minimizing the ℓ_0-norm means minimizing the number of nonzero elements.

Suppose $\boldsymbol{x} = [0\ 0\ \ -3\ 0.1\ 0]$. What is $||\boldsymbol{x}||_0$? What is $||\boldsymbol{x}||_\infty$?

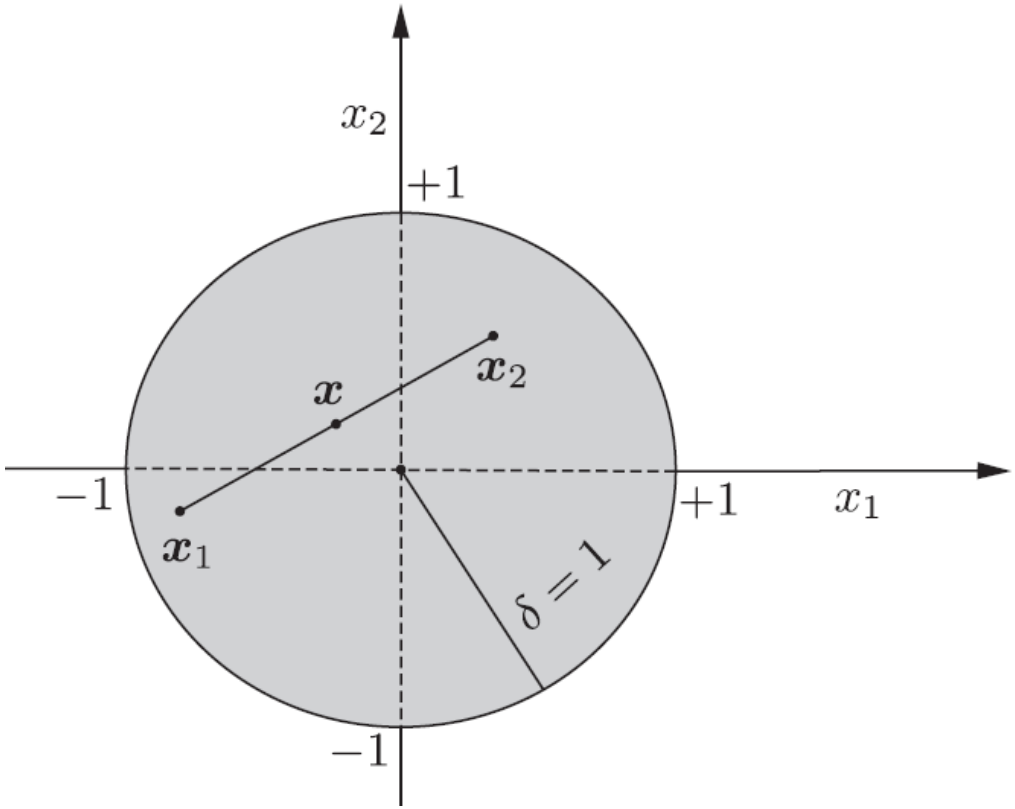


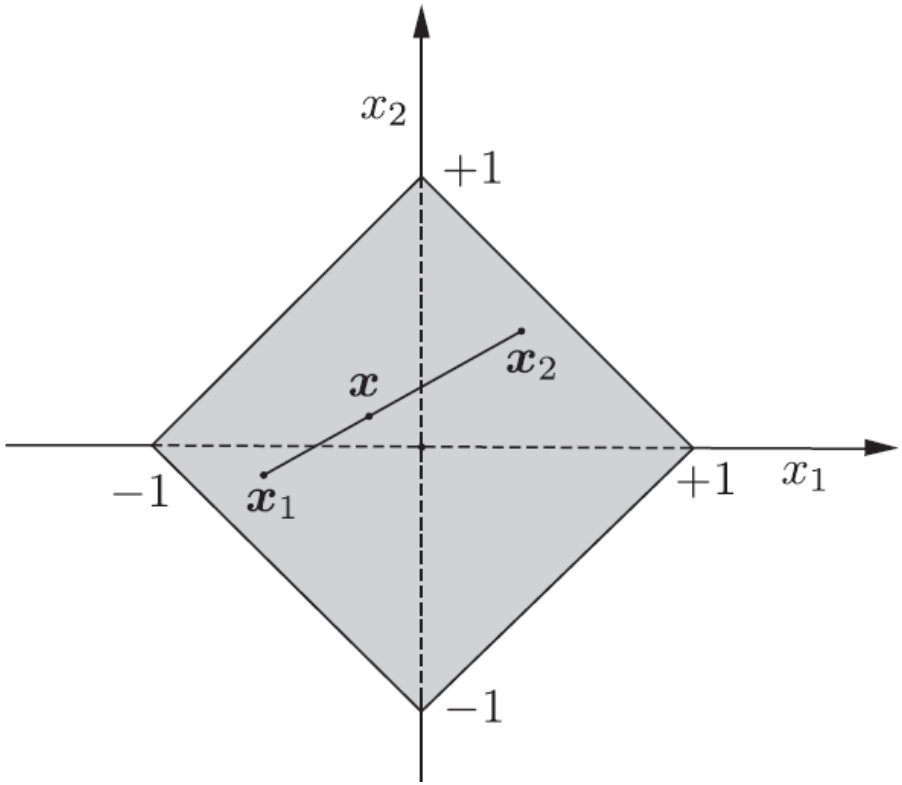
The ℓ_p-norm function is convex for $p \geq 1$. While it is nonconvex for $p < 1$.

A set $\mathcal{C}$ is called convex, if $\forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$ and if $\forall \lambda \in [0,1]$, the following holds true:

$$\boldsymbol{x} := \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \in \mathcal{C}$$

Geometrically, $\boldsymbol{x}$ lies in the line segment joining $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$

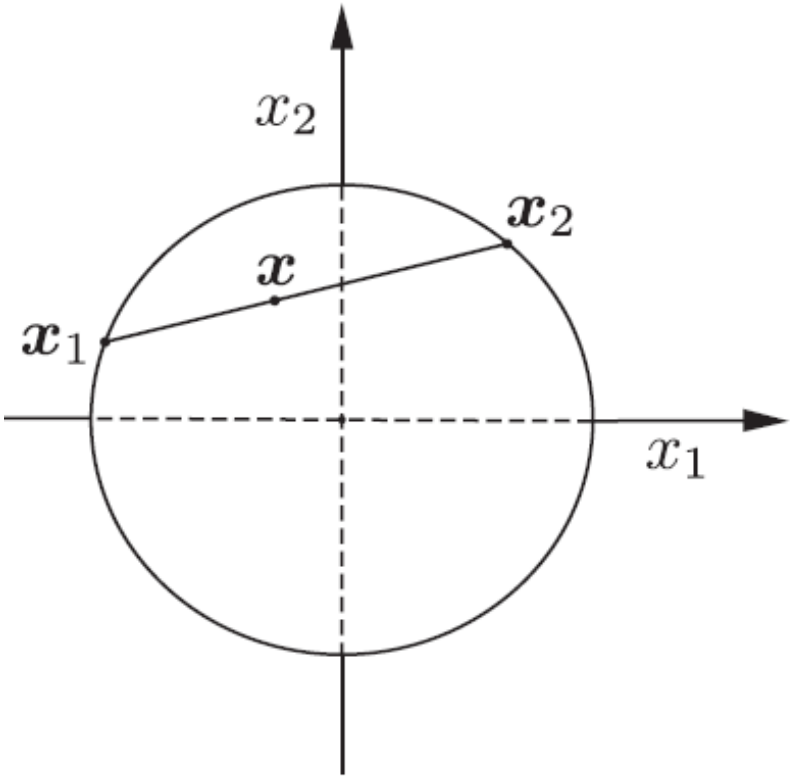


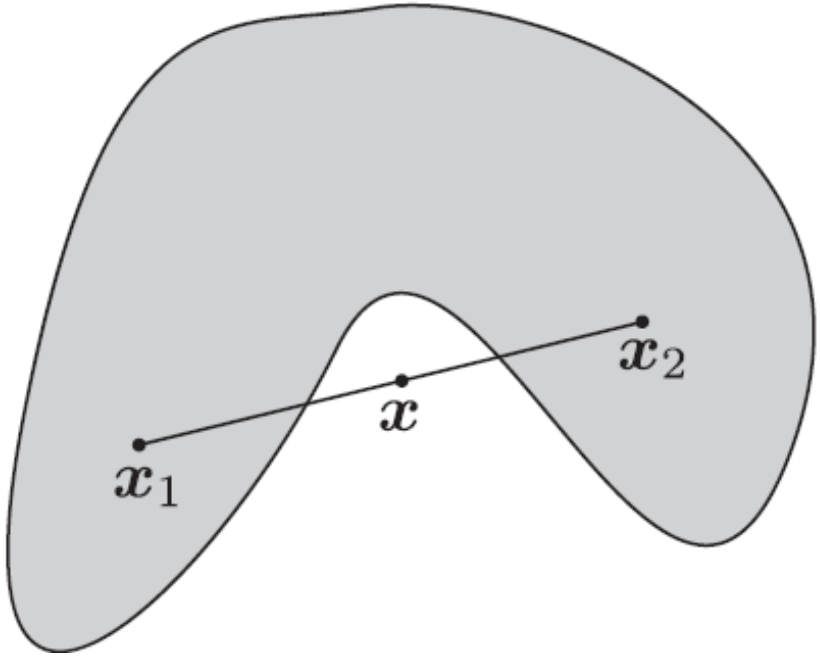


Is $\mathcal{C} = \left\{ \boldsymbol{x} := \sqrt{x_1^2 + x_2^2} \leq 1 \right\}$ **convex?**

Is $\mathcal{C} = \{ \boldsymbol{x} := |x_1| + |x_2| \leq 1 \}$ **convex?**

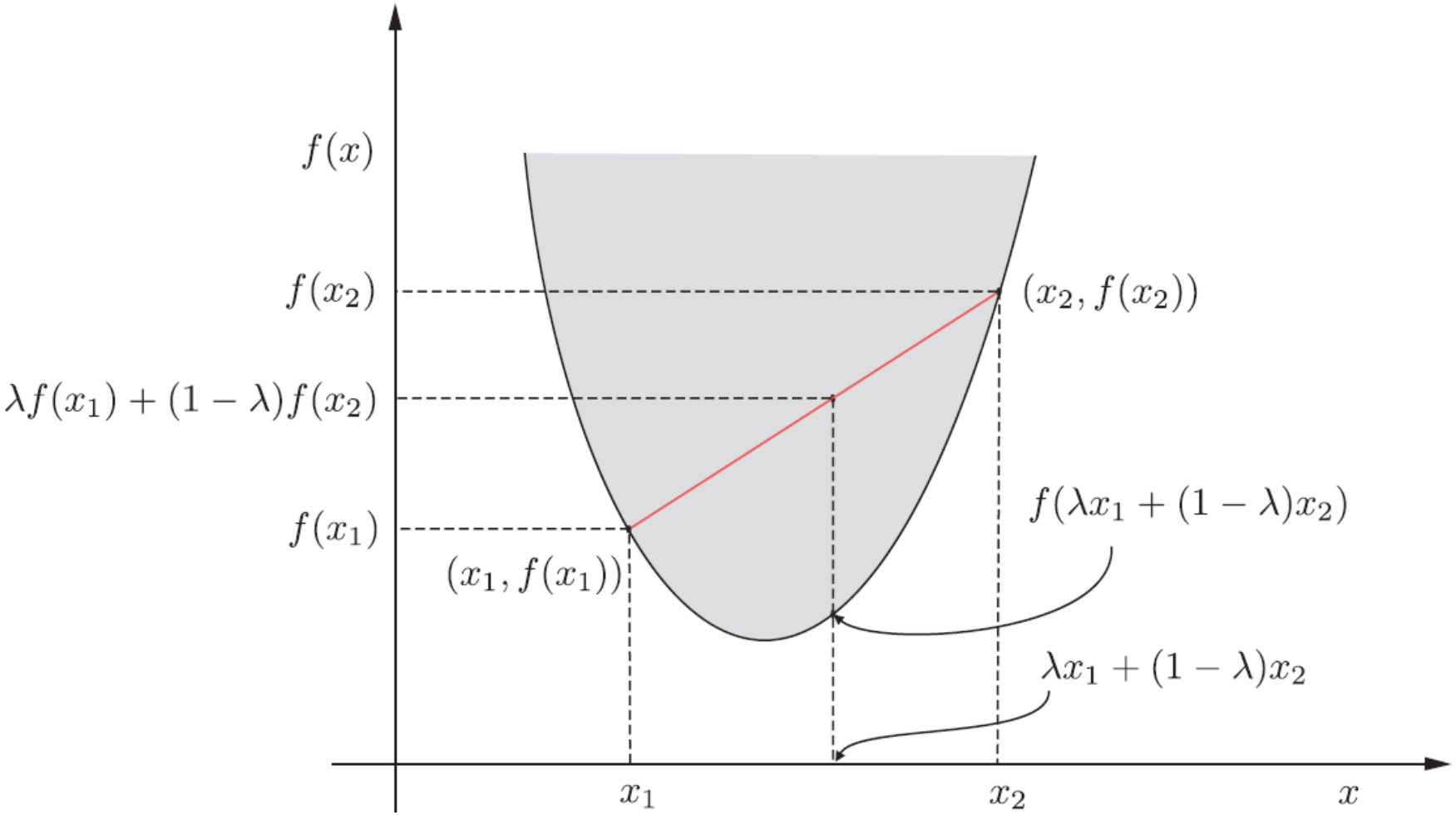
Is $\mathcal{C} = \left\{ \boldsymbol{x} := \sqrt{x_1^2 + x_2^2} = 1 \right\}$ **convex?**





A function f is called convex, if $\forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$ is a convex set and if $\forall \lambda \in [0, 1]$, the following holds true:

$$f(\lambda \boldsymbol{x}_1 + (1 - \lambda)\boldsymbol{x}_2) \leq \lambda f(\boldsymbol{x}_1) + (1 - \lambda) f(\boldsymbol{x}_2)$$



In this simple case that f is a quadratic function while x is just any point on the x-axis, it is easily seen that a convex function corresponds to a unique global minimum.

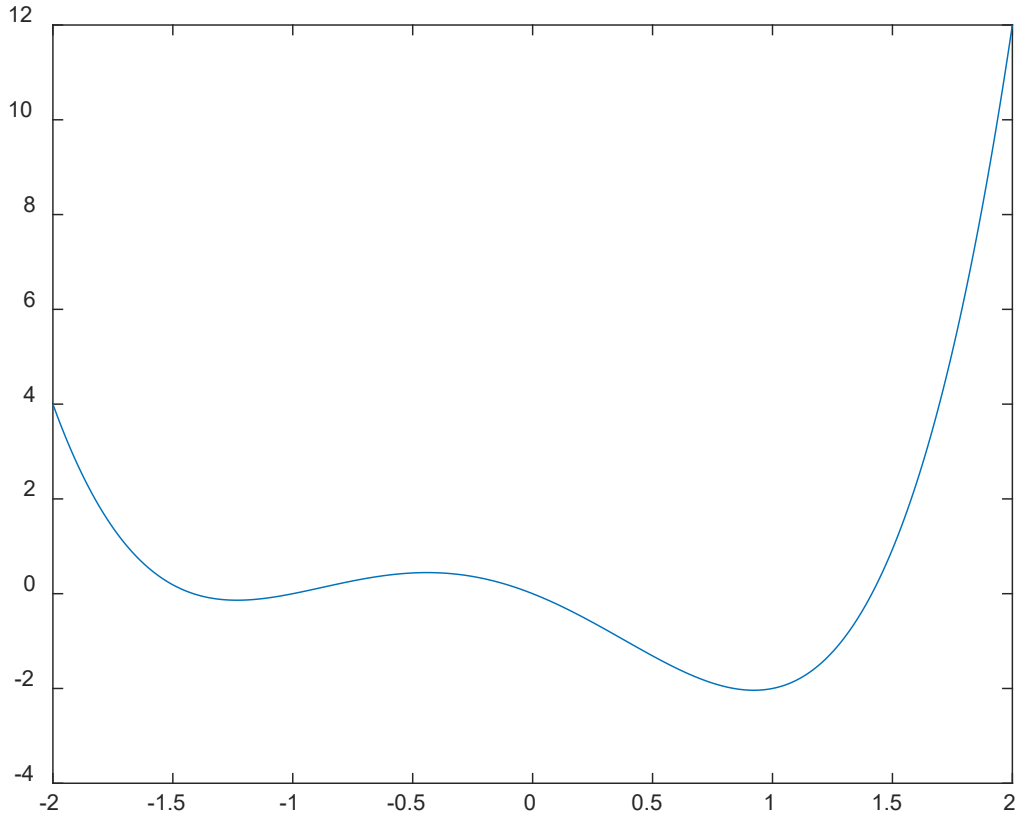
This indicates that if we perform optimization on a convex function, the global solution can be easily obtained. In particular, the gradient descent is guaranteed to obtain the solution from any initial points.

If a function is not convex, we call it nonconvex. If a function is nonconvex, then there are multiple minima, including the global minimum.

In this case, a difficulty is that we cannot ensure if the global solution is obtained unless all minima have been checked.

A simple nonconvex function example is:

$$f(x) = x^4 + x^3 - 2x^2 - 2x$$

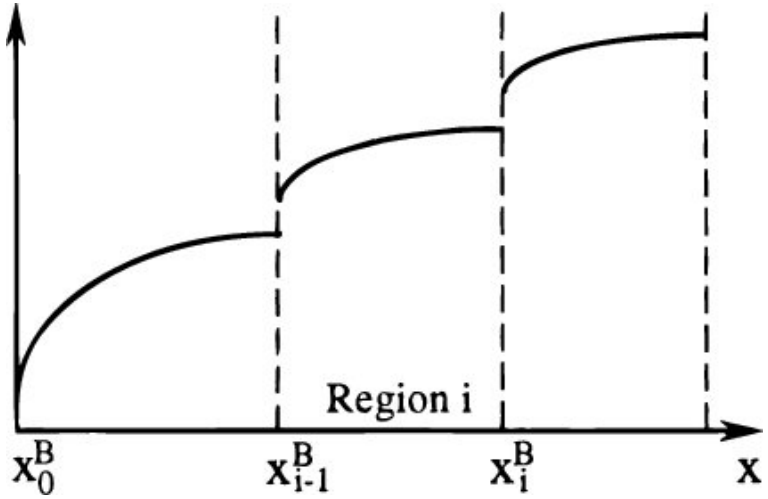


If we use gradient descent and start with $\hat{x}(0) = -1$, the global solution cannot be obtained.

A function can be classified as a continuous or discontinuous function.

f is continuous if there are no abrupt changes in value. It is clear that the above examples are all continuous functions as each graph is a single unbroken curve.

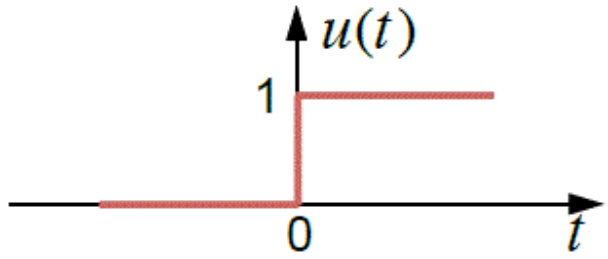
On the other hand, an example of discontinuous function can be an investment cost function:



Source:
https://www.researchgate.net/publication/231391851_Disjunctive_Programming_Techniques_for_the_Optimization_of_Process_Systems_with_Discontinuous_Investment_Costs-Multiple_Size_Regions/figures?lo=1

Also, the unit step function $u(t)$ is discontinuous as there is a sudden change from 0 to 1 at $t = 0$.

$$u(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases}$$

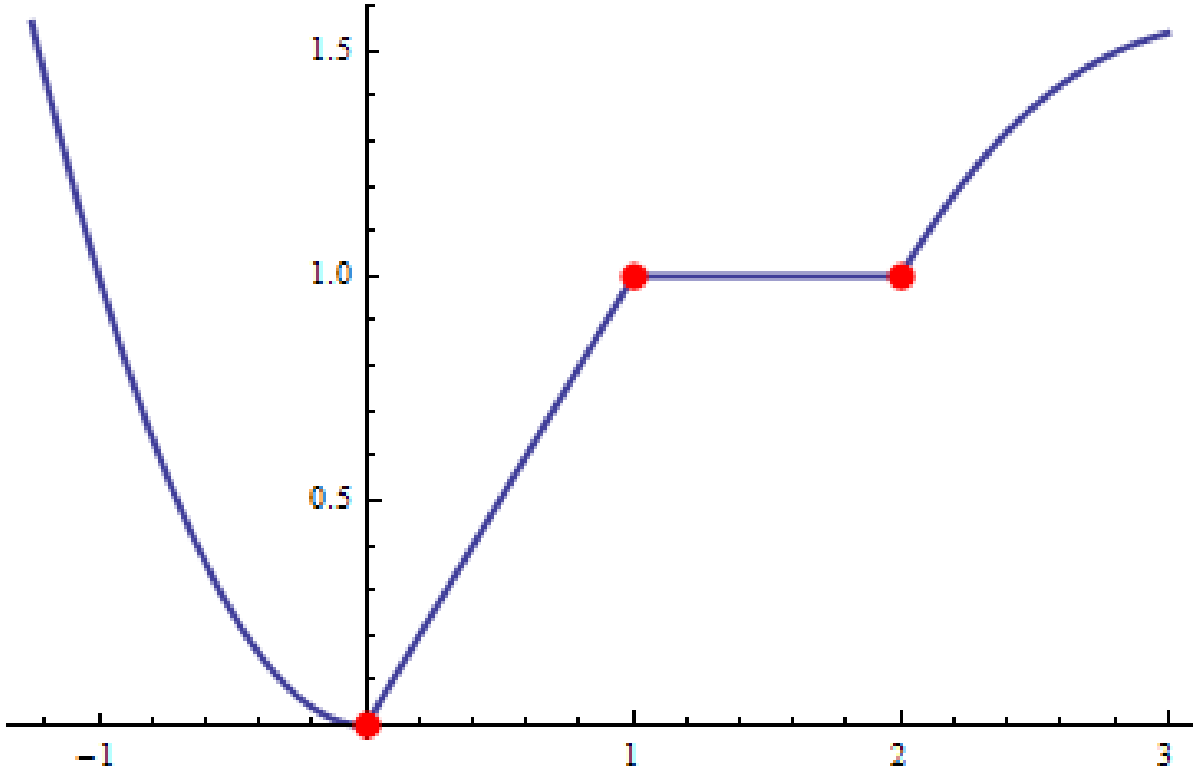


Furthermore, a function can be classified as a smooth or nonsmooth function.

If f is a smooth function, it has a unique defined first derivative at every point. Otherwise, it is a nonsmooth function.

It is clear that if f is smooth, it should be differentiable at all points. Hence a smooth function is basically a differentiable function.

Is this function smooth or nonsmooth?



Source: https://mathematica.stackexchange.com/questions/32872/how-to-find-the-non-differentiable-points-of-a-given-continuous-function

Example 3

Examine whether the function $f(x) = |x|^p$ is smooth or nonsmooth. Different values of $0 < p < \infty$ should be considered.

Using chain rule:

$$\frac{df(x)}{dx} = \frac{d|x|^p}{d|x|} \cdot \frac{d|x|}{dx} = p|x|^{p-1}\frac{d|x|}{dx}$$

We can see that if $x < 0$, the slope of $|x|$ is -1. On the other hand, the gradient is 1 for $x > 0$.

$$\frac{d|x|}{dx} = \text{sign}(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \\ [-1, 1], & x = 0 \end{cases}$$

where sign is sign function. Combining the results, we have:

$$\frac{df(x)}{dx} = p|x|^{p-1}\text{sign}(x)$$

For $p > 1$ or $p - 1 > 0$, $f(x)$ is differentiable for all values of x even at $x = 0$. Hence $f(x)$ is smooth for $p > 1$.

While for $p < 1$ or $1 - p > 0$, $f(x)$ is nondifferentiable at $x = 0$ because the term $|x|^{1-p}$ appears in the denominator, indicating an undefined derivative.

Even for $p = 1$, the slope of $|x|$ can be any values between 1 and -1 at $x = 0$, indicating that the derivative is not unique. Hence $f(x)$ is nonsmooth for $p \leq 1$.

Example 4
Given

$$x_n = A + q_n, \quad n = 1, \cdots, N$$

Consider the problem of estimating the constant A in the presence of noise q_n with ℓ_1-norm and ℓ_2-norm minimization.

The ℓ_2-norm minimization is in fact LS estimation:

$$\hat{A} = \arg\min_{\tilde{A}} J_2(\tilde{A}), \quad J_2(\tilde{A}) = \sum_{n=1}^{N} \left(x_n - \tilde{A}\right)^2$$

$$\left.\frac{dJ_2(\tilde{A})}{d\tilde{A}}\right|_{\tilde{A}=\hat{A}} = 2\sum_{n=1}^{N}\left(x_n - \hat{A}\right)(-1) = 0 \Rightarrow \hat{A} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

which is simply the average. While ℓ_1-norm minimization, also known as least absolute deviation (LAD), refers to:

$$\hat{A} = \arg\min_{\tilde{A}} J_1(\tilde{A}), \quad J_1(\tilde{A}) = \sum_{n=1}^{N} \left|x_n - \tilde{A}\right|$$

$$\left.\frac{dJ_1(\tilde{A})}{d\tilde{A}}\right|_{\tilde{A}=\hat{A}} = 2\sum_{n=1}^{N}\text{sign}\left(x_n - \hat{A}\right)(-1) = 0 \Rightarrow \hat{A} = \text{med}(x_n)$$

where we assign $\text{sign}(0) = 0$.

That is, we just arrange the values of $\{x_n\}$ in ascending (or descending) order, and then take the middle one. If N is an even number, the median can be determined as the average of the two middlemost numbers.

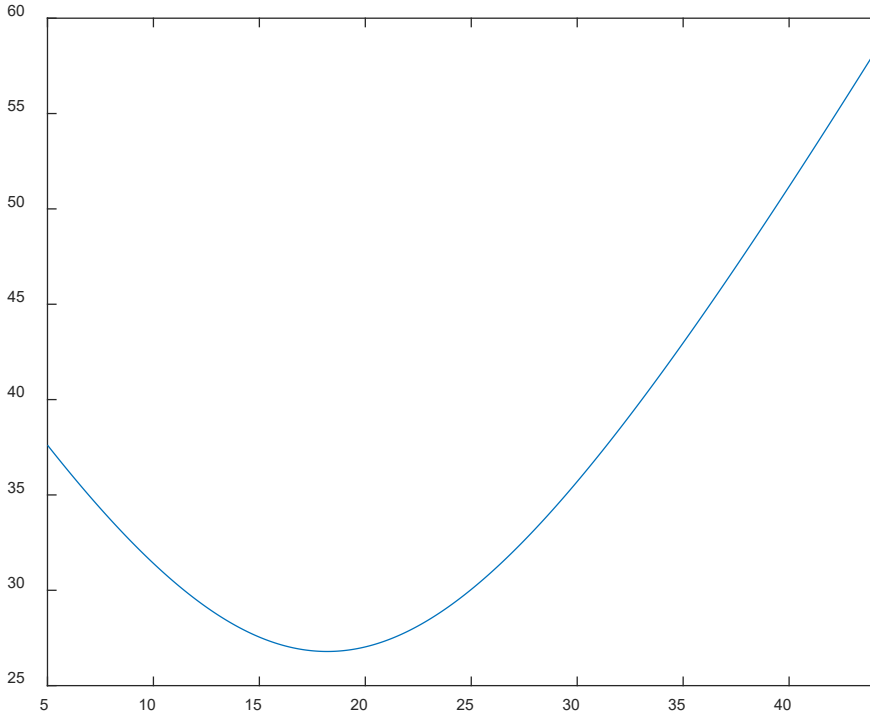
Suppose $N = 4$ with $x_1 = 10.2$, $x_2 = 8.9$, $x_3 = 41.3$ and $x_4 = 12.4$.

We easily see that x_3 is the outlier, and the nominal value of A may be around 10.

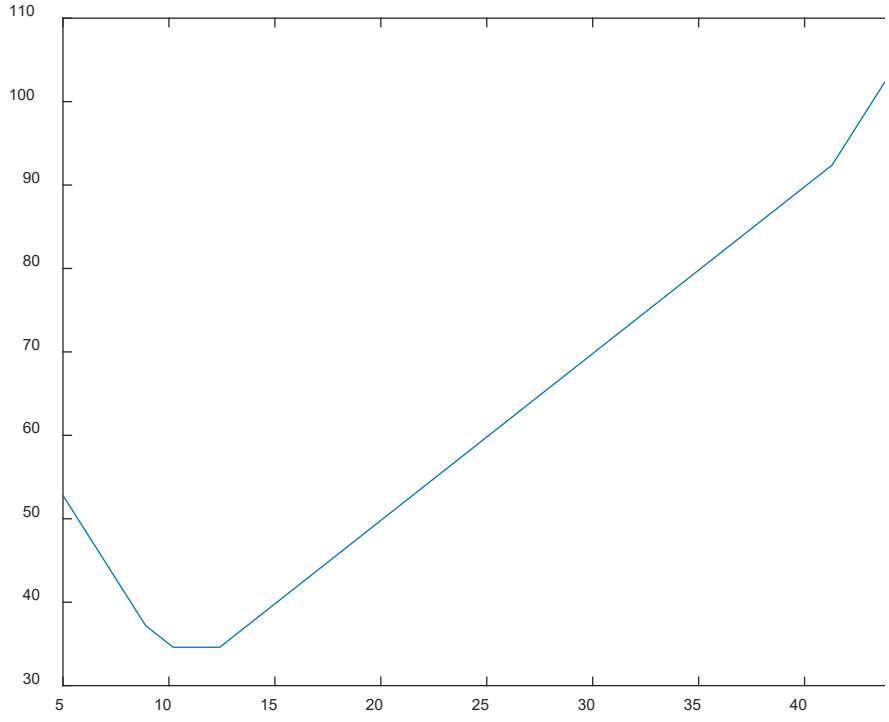
Using the LS, $\hat{A} = 18.2$ while using LAD, $\hat{A} = (10.2 + 12.4)/2 = 11.3$.

This illustrates that in the presence of outlier-contaminated measurements, ℓ_p-norm minimization is able to provide a more accurate solution.

We also see that the LAD solution may not be unique because any values between 10.2 and 12.4 is a valid solution.

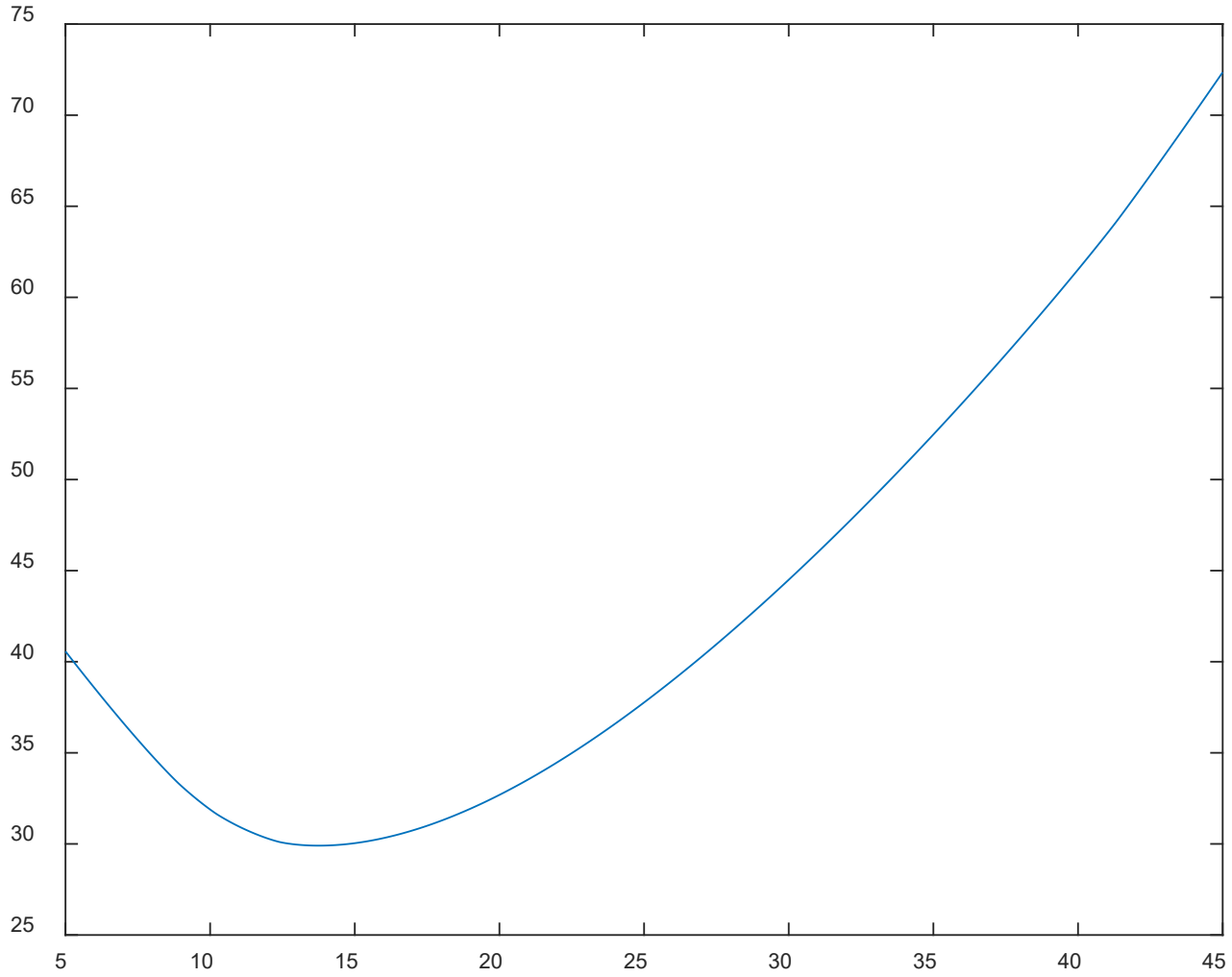


$p = 2$

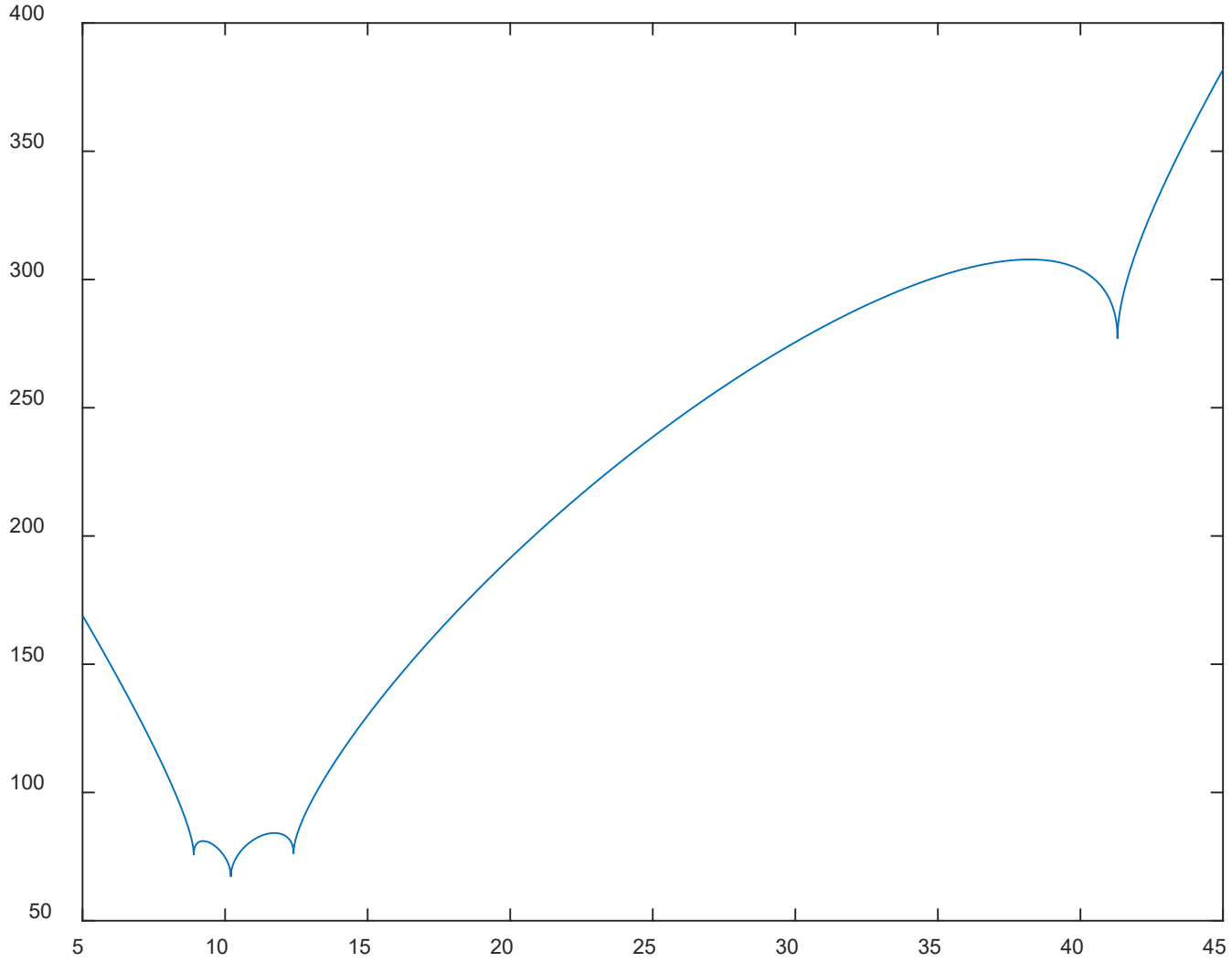


$p = 1$

Using $p = 1.5$, we get $\hat{A} = 13.75$



Using $p = 0.5$, we get $\hat{A} = 10.2$



For $p > 1$, ℓ_p-norm minimization yields a unique global minimum. We can start from any point and use gradient technique to find the minimum.

While for $p < 1$, ℓ_p-norm minimization can result in multiple minima and we might use an exhaustive search to find the global minimum. If gradient technique is employed, we can obtain the global solution if the initial guess is sufficiently close to it. Otherwise, the obtained solution corresponds to a local minimum.

To solve general ℓ_p-norm minimization problems, even for the linear cases, iterative technique is generally required.

One popular method for solving ℓ_p-norm minimization with linear variables is the iteratively reweighted least squares (IRLS).

To understand IRLS, it may be easier to learn weighted least squares (WLS) first.

Consider the linear model:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{q} \tag{2}$$

where $\boldsymbol{y} \in \mathbb{R}^M$ is the observation vector, $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is known matrix, $\boldsymbol{x} \in \mathbb{R}^N$ is the parameter vector to be determined, and $\boldsymbol{q} \in \mathbb{R}^M$ contains noise.

Given $\boldsymbol{y}$, the task is to find $\boldsymbol{x}$. To obtain a unique solution, we assume $M \geq N$, i.e., the number of measurements is at least equal to the number of unknowns.

The LS cost function is constructed as:

$$J_2(\tilde{\boldsymbol{x}}) = (\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}})^T(\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}}) = ||\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}}||_2^2$$

Minimizing $J_2(\tilde{\boldsymbol{x}})$ means that we basically assume that the noise levels of all elements in $\boldsymbol{q} \in \mathbb{R}^M$ are same or even i.i.d.

It is because we use the same weighting of one in each element of $\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}$ (or each row of $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{q}$).

The solution has been derived as:

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y} \tag{3}$$

However, when the elements of $\boldsymbol{q}$ are not of same power and are even correlated, the WLS is able to provide a more accurate solution.

In WLS, a symmetric weighting matrix $\boldsymbol{W} = \boldsymbol{W}^T$ is included in the ℓ_2-norm cost function:

$$J_2(\tilde{\boldsymbol{x}}) = (\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}})^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}}) = \| \boldsymbol{W}^{1/2} (\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}}) \|_2^2 \tag{4}$$

One basic idea is to put a larger weight where the noise level is small while a smaller weight where the noise level is large.

Expanding $J_2(\tilde{\boldsymbol{x}})$ yields:

$$J_2(\tilde{\boldsymbol{x}}) = \boldsymbol{y}^T \boldsymbol{W} \boldsymbol{y} - 2\tilde{\boldsymbol{x}}^T \boldsymbol{A}^T \boldsymbol{W} \boldsymbol{y} + \tilde{\boldsymbol{x}}^T \boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A} \tilde{\boldsymbol{x}}$$

Differentiating $J_2(\tilde{\boldsymbol{x}})$ w.r.t. $\tilde{\boldsymbol{x}}$ and then setting the resultant expression to 0, we get:

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{W} \boldsymbol{y} \tag{5}$$

Example 5

Repeat Example 4 using WLS with

$$\boldsymbol{W} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Recall the measurements:

$$x_n = A + q_n, \quad n = 1, \cdots, N$$

Using the linear model of (1), $\boldsymbol{A}$ becomes a vector of one, i.e., $\mathbf{1}_N$. According to (3), the LS estimate is:

$$\hat{A} = (\mathbf{1}_N^T \mathbf{1}_N)^{-1} \mathbf{1}_N^T \boldsymbol{x} = \frac{\mathbf{1}_N^T \boldsymbol{x}}{\mathbf{1}_N^T \mathbf{1}_N} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

Using (5), the WLS estimate is:

$$\hat{A} = \frac{\mathbf{1}_N^T \boldsymbol{W} \boldsymbol{x}}{\mathbf{1}_N^T \boldsymbol{W} \mathbf{1}_N}$$

For the 4-element case, the WLS estimate is:

$$\hat{A} = \frac{10.2 + 8.9 + 0.01 \times 41.3 + 12.4}{1 + 1 + 0.01 + 1} = 10.60$$

We see that the WLS estimate is able to provide higher accuracy than the LS and its value is comparable with the robust ℓ_p-norm based estimate.

It is because a very small weighting is used in the measurement with possibly large noise level. In doing so, the contribution of x_3 is very small.

The ℓ_p-norm minimization for the linear model is:

$$J_p(\tilde{\boldsymbol{x}}) = ||\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}}||_p^p$$

Let the residual be:

$$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}} = \begin{bmatrix} r_1 & \cdots & r_M \end{bmatrix}^T$$

Then $J_p(\tilde{\boldsymbol{x}})$ can be rewritten as:

$$J_p(\tilde{\boldsymbol{x}}) = \sum_{m=1}^{M} |r_m|^p = \sum_{m=1}^{M} |r_m|^{p-2} r_m^2$$

Considering $|r_m|^{p-2}$ as the weight in $\boldsymbol{W}$, we can write:

$$J_p(\tilde{\boldsymbol{x}}) = (\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}})^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}}) = \| \boldsymbol{W}^{1/2} (\boldsymbol{y} - \boldsymbol{A}\tilde{\boldsymbol{x}}) \|_2^2$$

where

$$\boldsymbol{W} = \begin{bmatrix} |r_1|^{p-2} & 0 & 0 & 0 \\ 0 & |r_2|^{p-2} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & |r_M|^{p-2} \end{bmatrix} = \begin{bmatrix} \dfrac{|r_1|^p}{r_1^2} & 0 & 0 & 0 \\ 0 & \dfrac{|r_2|^p}{r_2^2} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dfrac{|r_M|^p}{r_M^2} \end{bmatrix}$$

Assuming that $\boldsymbol{W}$ is independent of $\boldsymbol{x}$, the WLS solution is:

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{W}\boldsymbol{y}$$

Because $\boldsymbol{W}$ is a function of $\boldsymbol{x}$, we estimate $\boldsymbol{x}$ iteratively until convergence:

$$\hat{\boldsymbol{x}}^{k+1} = (\boldsymbol{A}^T\boldsymbol{W}(\hat{\boldsymbol{x}}^k)\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{W}(\hat{\boldsymbol{x}}^k)\boldsymbol{y} \qquad (6)$$

That is, we start from an initial estimate $\hat{\boldsymbol{x}}^0$, then construct $\boldsymbol{W}(\hat{\boldsymbol{x}}^0)$, compute $\hat{\boldsymbol{x}}^1$, and so on.

For simplicity, $\hat{\boldsymbol{x}}^0$ can be set as random numbers or computed using the LS:

$$\hat{\boldsymbol{x}}^0 = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$$

Note that for $p < 1$, global solution can be obtained if $\hat{\boldsymbol{x}}^0$ is sufficiently close to the global minimum.

To avoid division by zero issue if $r_m = 0$, we may modify the weight as:

$$\frac{|r_m|^p}{r_m^2 + \epsilon}$$

where $\epsilon > 0$ is a very small value.

Example 6

Consider the problem of line fitting with N data points. That is, given the measurement y_n, which is modelled as:

$$y_n = ax_n + b + q_n, \quad n = 1, \cdots, N$$

where x_n is known and q_n is noise term. We need to find a and b, and then construct the line:

$$\hat{y} = \hat{a}x + \hat{b}$$

It is clear that the observations align with the linear model of (2):

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix}$$

Suppose the ideal straight line is:

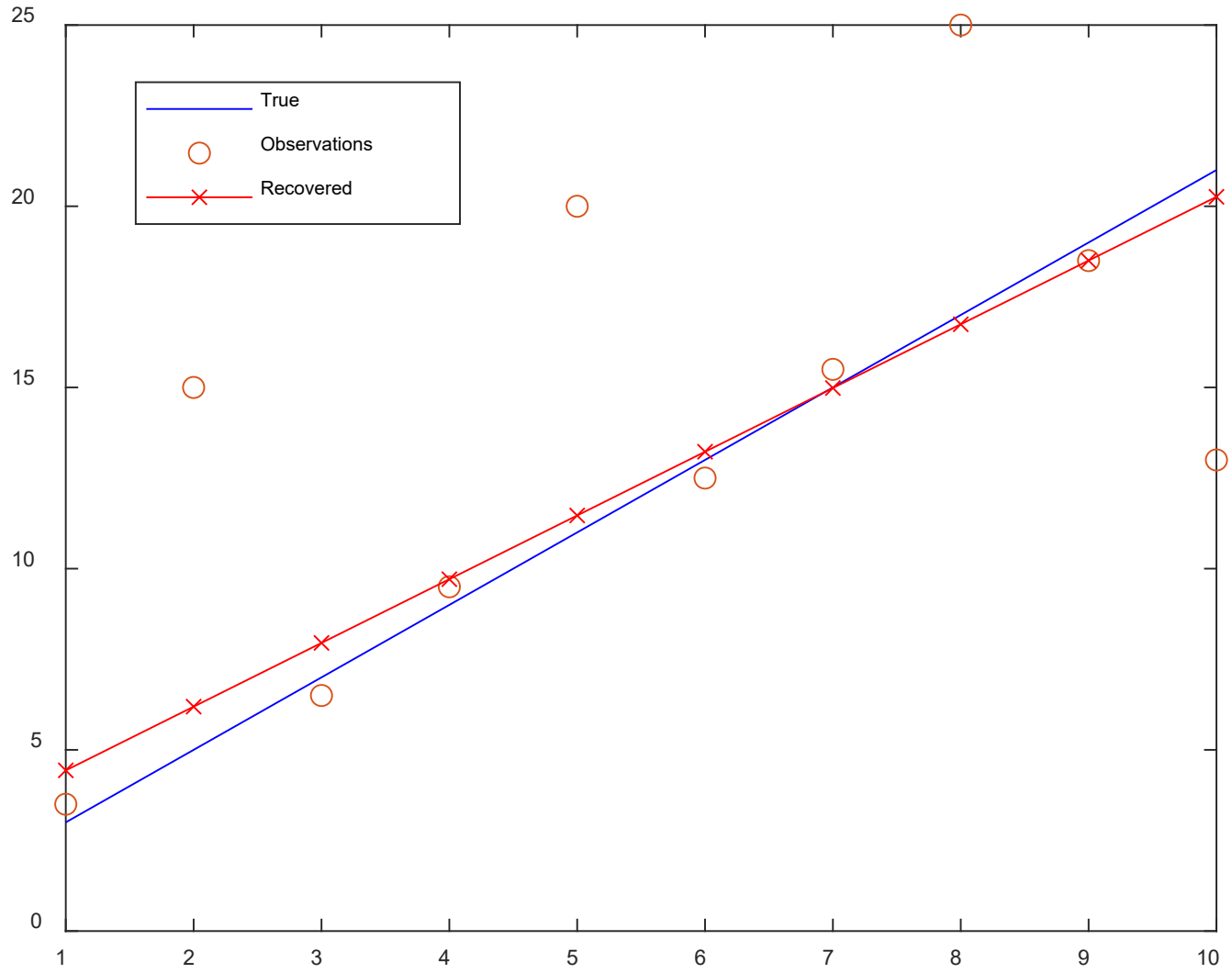
$$y = 2x + 1$$

Let

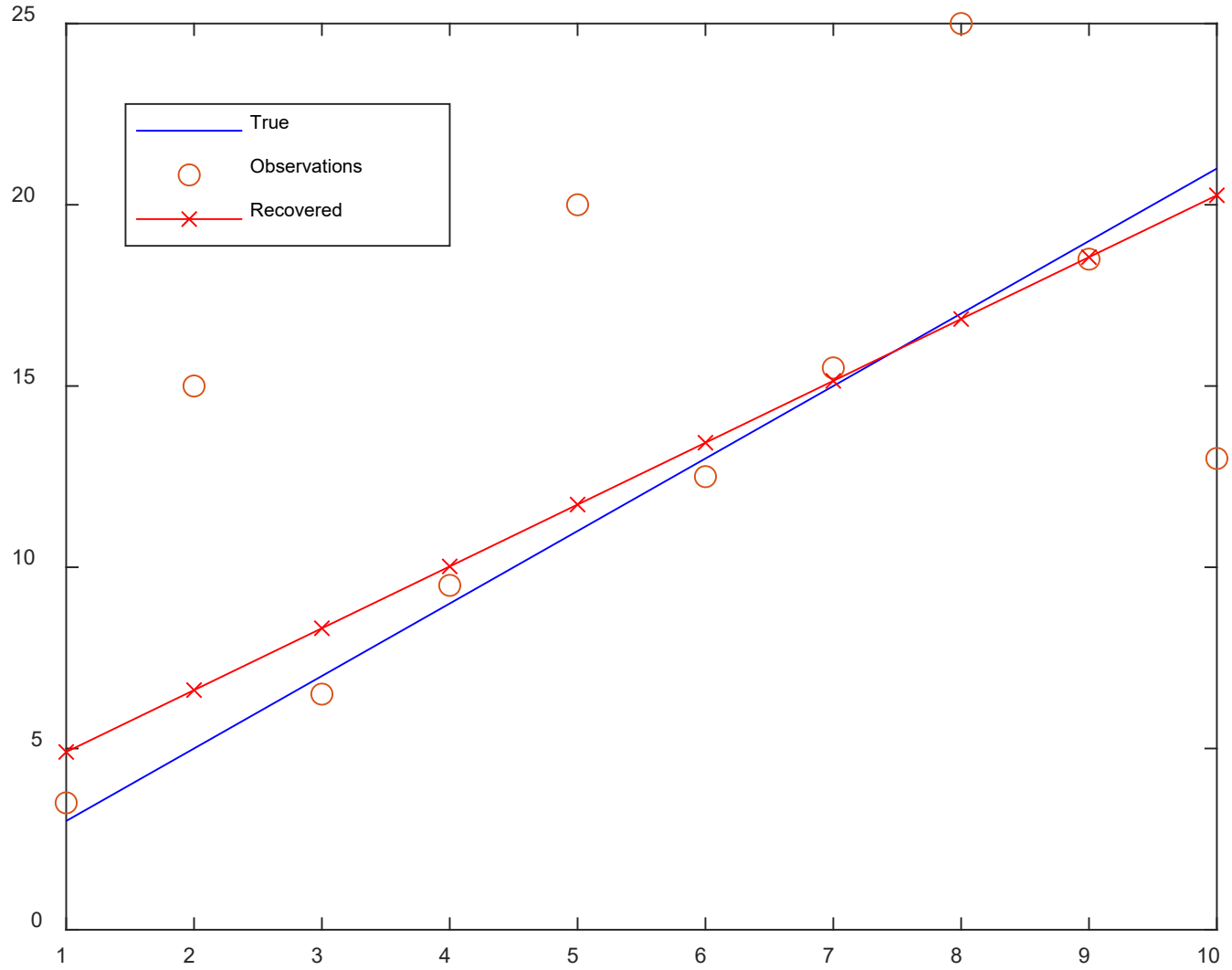
```
x =
     1     2     3     4     5     6     7     8     9    10

y=
    3.5000   15.0000    6.5000    9.5000    6.0000   12.5000
15.5000   25.0000   18.5000   13.0000

q=
0.5000   10.0000   -0.5000    0.5000    9.0000   -0.5000
0.5000    8.0000   -0.5000   -8.0000
```

We use IRLS with 5 iterations initialized by the LS solution. The outliers are very strong and we see that the smaller the value of p, more accurate line fitting is achieved.

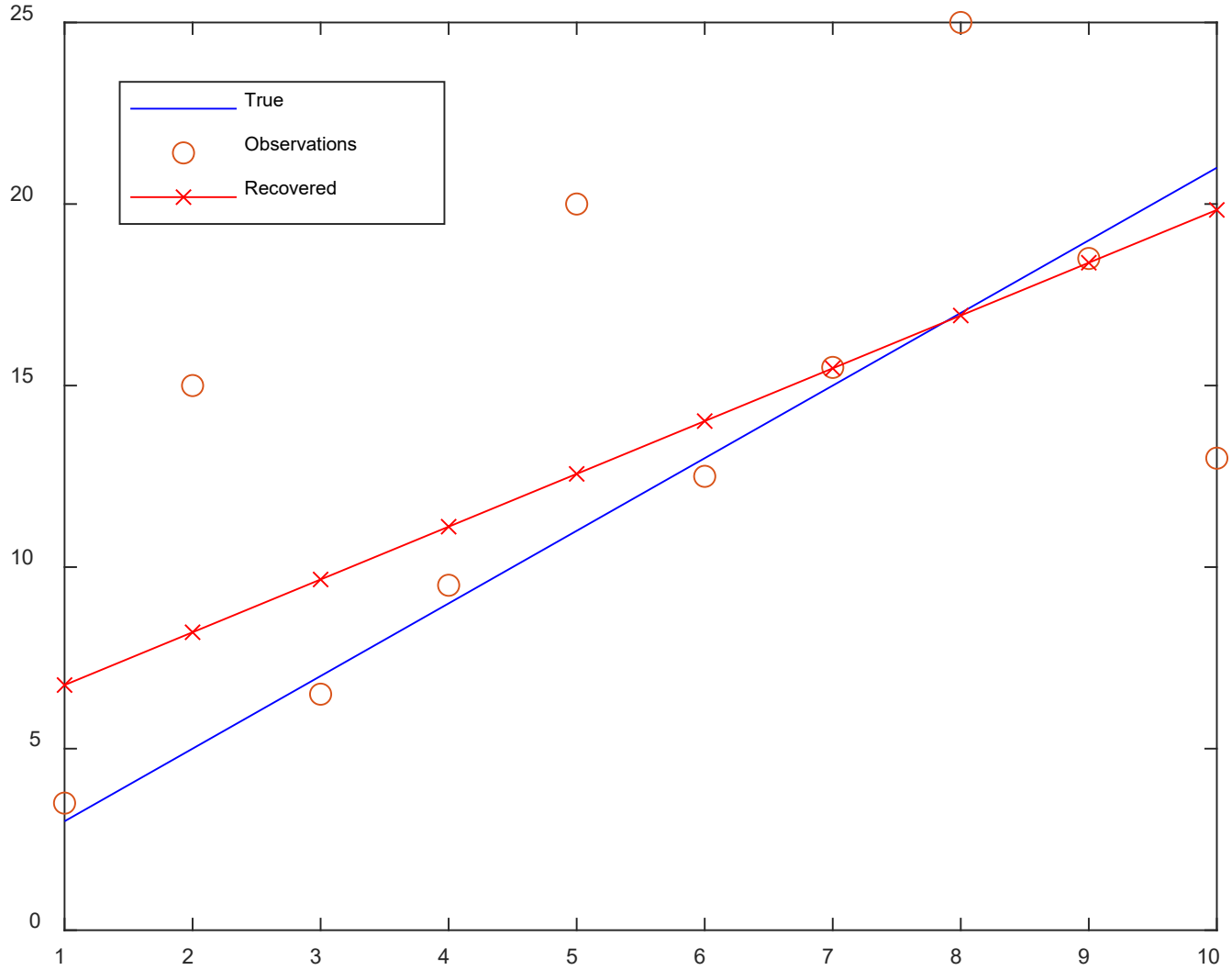
$p = 0.5$: $a = 1.7585$ and $b = 2.6755$



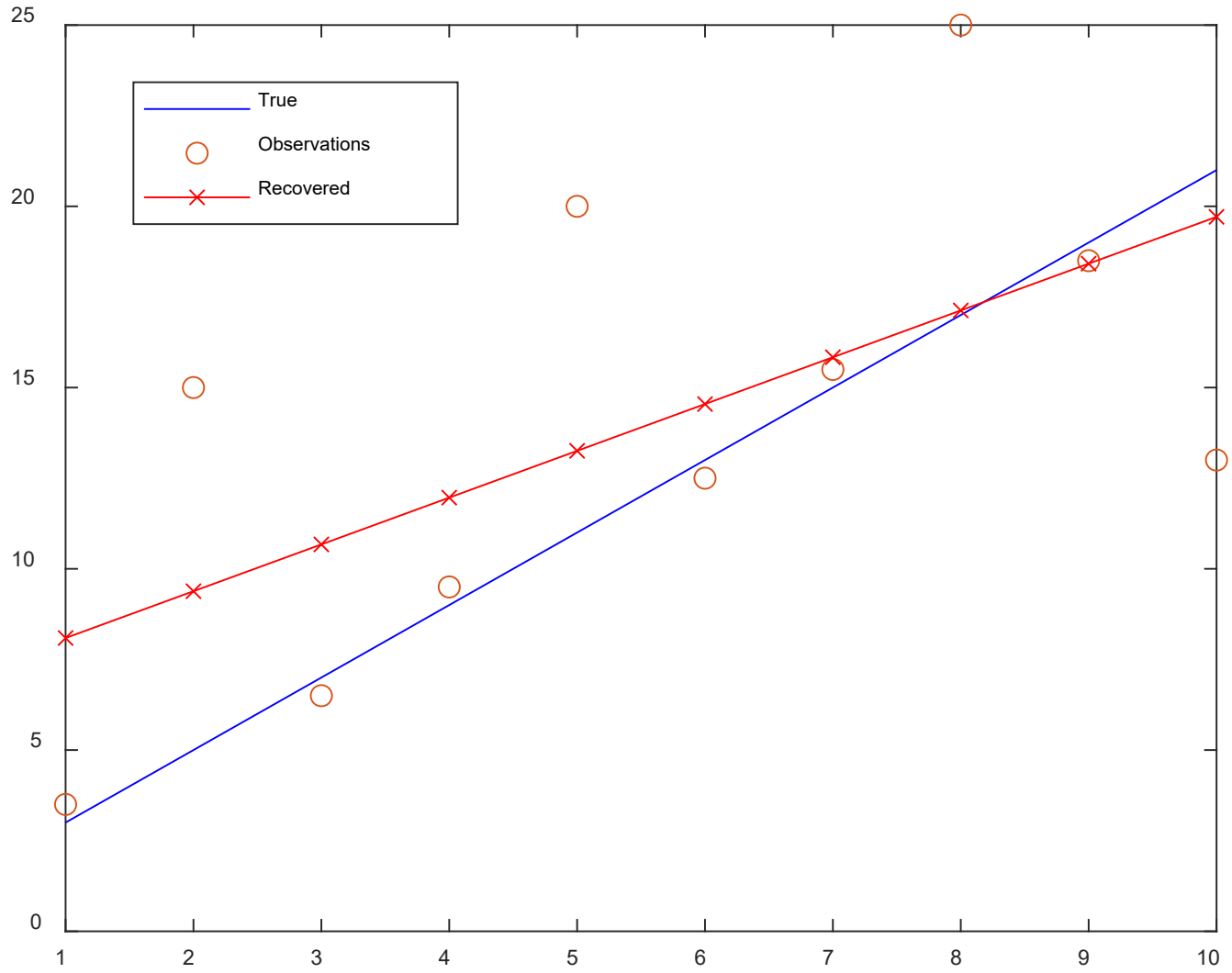
$p = 1$: $a = 1.7064$ and $b = 3.1978$



$p = 1.5$: $a = 1.4540$ and $b = 5.2976$



$p = 2$: $a = 1.2909$ and $b = 6.8000$



As the ℓ_2-norm minimization based matrix decomposition do not provide accurate results in the presence of outliers, we can apply ℓ_p-norm minimization to achieve better results.

A matrix measurement $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ embedded in outliers can be modelled as:

$$\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{Q}$$

where $\boldsymbol{L}$ is the low-rank component with $\text{rank}(\boldsymbol{L}) = r \ll \min(n_1, n_2)$ and $\boldsymbol{Q}$ contains outliers.

In image processing, $\boldsymbol{L}$ may be the texture component in the image.

In video processing, $\boldsymbol{L}$ corresponds to the background component in the video frames.

Combining the ideas of low-rank matrix factorization and ℓ_p-norm minimization, we can find $\boldsymbol{L}$ via

$$\min_{\boldsymbol{U},\boldsymbol{V}} \; J(\boldsymbol{U},\boldsymbol{V}) := \|\boldsymbol{U}\boldsymbol{V} - \boldsymbol{X}\|_p^p \tag{7}$$

where $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times n_2}$. The matrix ℓ_p-norm is defined as:

$$\|\boldsymbol{X}\|_p = \left(\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} |x_{i,j}|^p \right)^{1/p}$$

As in the ALS for recommender systems, we apply alternating minimization strategy:

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \; \|\boldsymbol{U}^k\boldsymbol{V} - \boldsymbol{X}\|_p^p$$

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \; \|\boldsymbol{U}\boldsymbol{V}^{k+1} - \boldsymbol{X}\|_p^p$$

For a fixed $\boldsymbol{U}$, we solve $\boldsymbol{V}$ via:

$$\min_{\boldsymbol{V}} \; J(\boldsymbol{V}) := \|\boldsymbol{U}\boldsymbol{V} - \boldsymbol{X}\|_p^p = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \|\boldsymbol{u}_i^T \boldsymbol{v}_j - \boldsymbol{X}_{ij}\|_p^p$$

Note that $(\cdot)^k$ is dropped for notational simplicity. Hence we can find $\boldsymbol{v}_j$ using the IRLS independently:

$$\min_{\boldsymbol{v}_j} \; J(\boldsymbol{v}_j) := \sum_{i=1}^{n_1} \|\boldsymbol{u}_i^T \boldsymbol{v}_j - \boldsymbol{X}_{ij}\|_p^p = \|\boldsymbol{U}\boldsymbol{v}_j - \boldsymbol{b}_j\|_p^p, \quad j = 1, \cdots, n_2$$

where $\boldsymbol{b}_j = [\boldsymbol{X}_{1j}, \cdots, \boldsymbol{X}_{n_1 j}]^T$ is the jth column of $\boldsymbol{X}$. Similarly, $\boldsymbol{u}_i$ is computed independently using IRLS for a fixed $\boldsymbol{V}$.

To summarize, given $\boldsymbol{U}^0$, we estimate $\boldsymbol{v}_j$, $j = 1, \cdots, n_2$, and $\boldsymbol{u}_i$, $i = 1, \cdots, n_1$, in an alternative manner using the IRLS, until convergence.

Then the low-rank component $\boldsymbol{L}$ is estimated as:

$$\hat{\boldsymbol{L}} = \boldsymbol{U}^k \boldsymbol{V}^k$$

If the outlier component is of interest, we can estimate it as:

$$\hat{\boldsymbol{Q}} = \boldsymbol{X} - \hat{\boldsymbol{L}}$$

When the observation matrix contains missing entries, we can easily modify the optimization cost function as:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \; J(\boldsymbol{U},\boldsymbol{V}) := \left\| (\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega \right\|_p^p \tag{8}$$

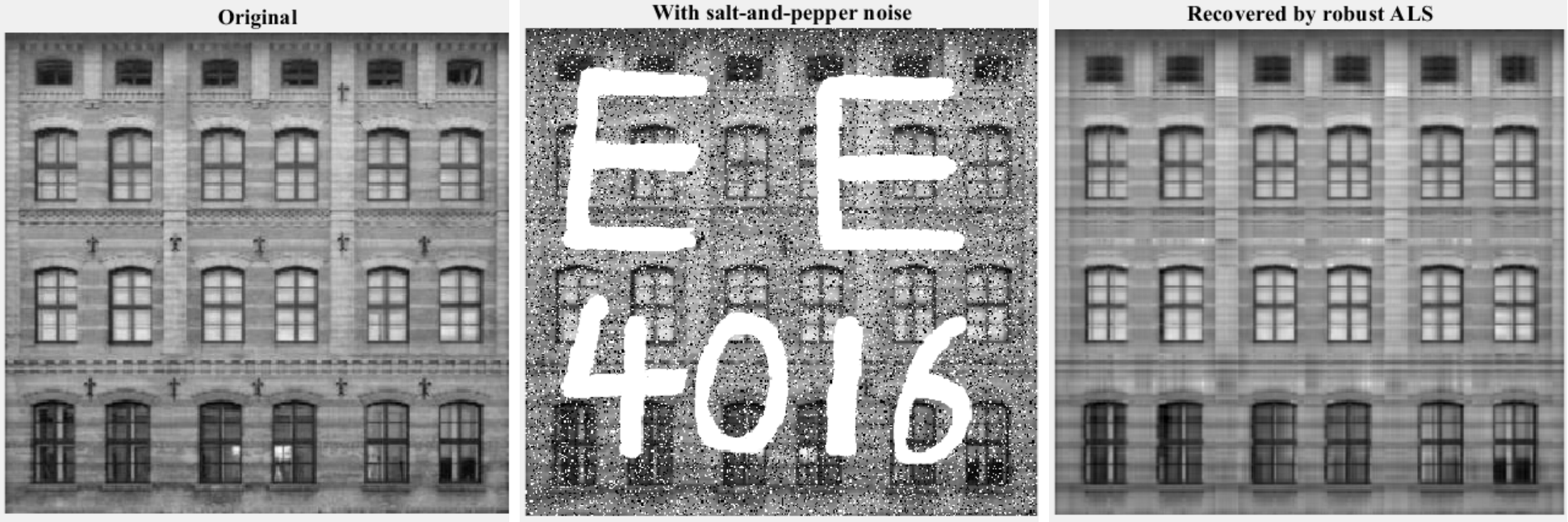
Adopting alternating minimization, matrix factorization and IRLS, ℓ_p-norm minimization solution can be obtained. We may say (8) generalizes the matrix factorization/recovery problem as $p \in (0, 2]$.

Example 7

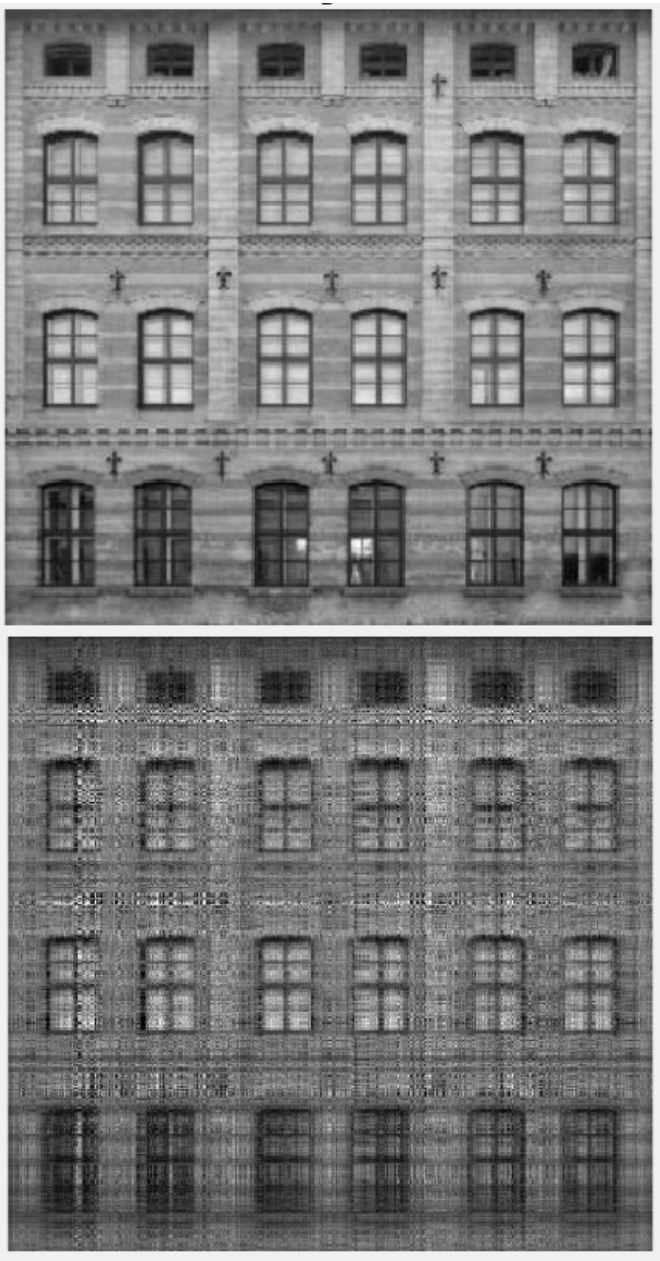
Repeat Example 1 using the ℓ_p-norm minimization based matrix factorization approach with $p = 1$ and $r = 2$.

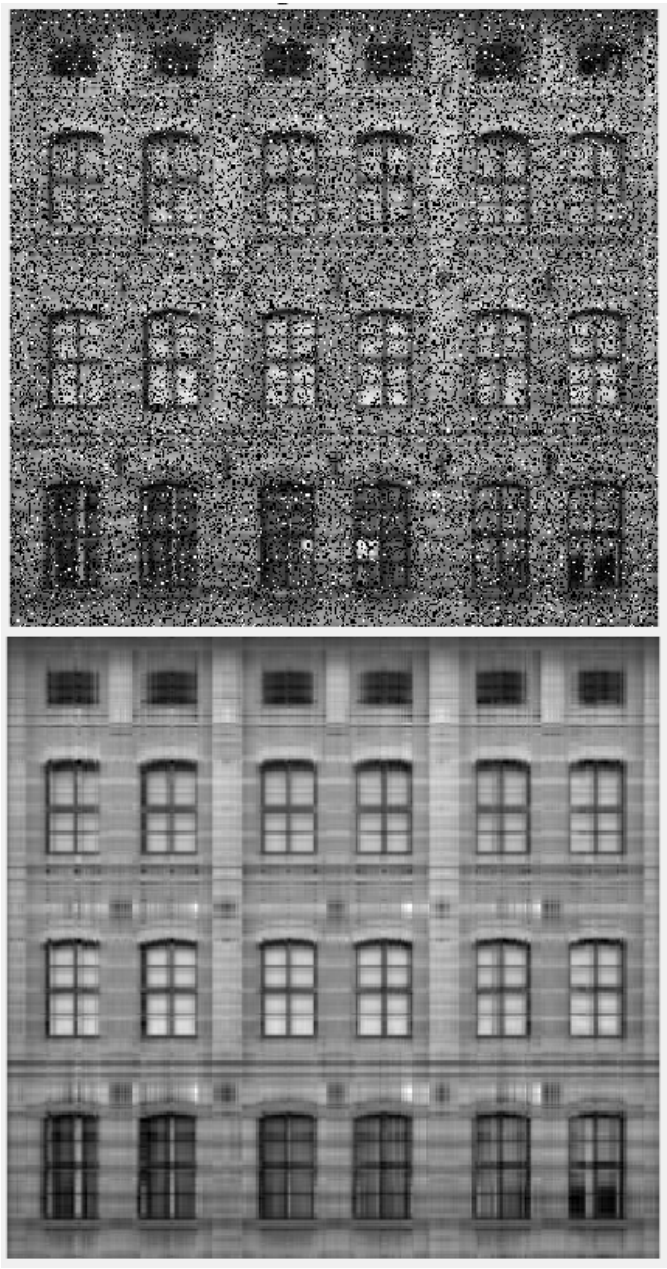
Example 8

Repeat Example 2 using the ℓ_p-norm minimization based matrix factorization approach with $p = 1$ and $r = 7$.



In fact, similar results are obtained when there are no missing entries in the image:





Example 9

We consider the application of video surveillance. From two open datasets, we select the first 200 frames where each frame has dimensions of 240×320, corresponding to 76,800 pixels. Thus, the observed matrix constructed from each video is $\boldsymbol{X} \in \mathbb{R}^{76800 \times 200}$ where $m = 76,800$ and $n = 200$.

That is, we convert each frame of a video as a column of a matrix, the resultant matrix due to the background is of low-rank.

Foreground objects such as moving cars or walking pedestrians, generally occupy only a fraction of the image pixels and hence can be treated as sparse outliers.

Background is $\boldsymbol{L}$ and foreground is $\boldsymbol{Q}$ and we set the rank as $r = 1$.

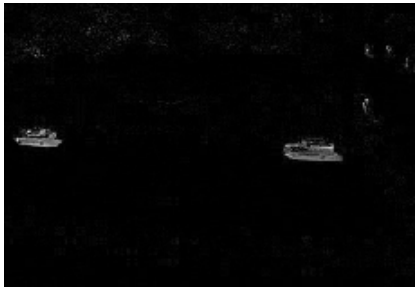


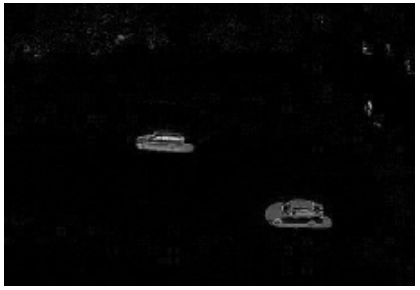


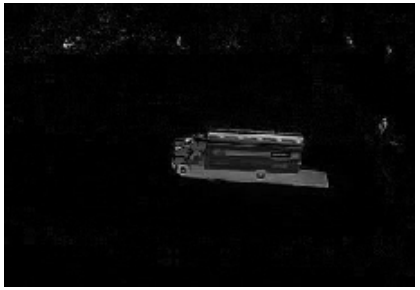








References:

1. S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*, Academic Press, 2015
2. http://jacarini.dinf.usherbrooke.ca/dataset2014/