Deep Learning and Inverse Problems
Summer 2024
Lecturer: Reinhard Heckel

Problem Set 1

Issued: Tuesday April 16, 2024, 1:00 pm.
Due: Tuesday April 23, 2024, 1:00 pm.

Problem 1 (denoising)**.** In the first chapter, we saw that signal models are central for solving inverse problems. Here, we consider a denoising problem and show that if a n-dimensional signal lies in a k-dimensional subspace, we can remove a fraction of $\frac{n-k}{k}$ of additive Gaussian noise. Consider denoising problem, where we are given a noisy measurement of a signal $\mathbf{x}^*$ as

$$\mathbf{y} = \mathbf{x}^* + \mathbf{z}.$$

We assume that $\mathbf{x}^* \in \mathbb{R}^n$ is a signal that lies in a k-dimensional subspace, and $\mathbf{z}$ is zero-mean Gaussian noise with co-variance matrix $(\sigma^2/n)\mathbf{I}$. Let $\mathbf{U} \in \mathbb{R}^{n \times k}$ be an orthonormal basis of the signal subspace. We denoise the signal by projecting the measurement onto the subspace, i.e., we consider the estimate $\hat{\mathbf{x}} = \mathbf{U}\mathbf{U}^T\mathbf{y}$.

1. Show that
$$\mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2\right] = \sigma^2 \frac{k}{n},$$
where expectation is over the random noise $\mathbf{z}$.

   Hint: Recall that if $\mathbf{V} \in \mathbb{R}^{n \times n}$ is a unitary matrix (i.e., a matrix with orthonormal columns) and $\mathbf{z}$ has iid, zero-mean Gaussian entries, then $\mathbf{V}\mathbf{z}$ has the same distribution as $\mathbf{z}$.

2. Does the algorithm $\hat{\mathbf{x}} = \mathbf{U}\mathbf{U}^T\mathbf{y}$ denoise more or less if the dimension of the subspace becomes smaller, and what is your intuition on whether a better algorithm exists?

3. Next, we study this denoising algorithm numerically (ideally with python in a jupyter notebook using the libary numpy; if you are not familiar with those, this exercise is a good exercise to familiarize yourself).

   Towards this goal, generate a random k-dimensional subspace in $\mathbb{R}^{1000}$, and generate 500 random points in that subspace. Next, denoise each of those data points with the method above, and plot the average of the mean-squared error $\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2/\|\mathbf{x}^*\|_2^2$ along with corresponding standard deviations as error bar for different values of $k = 1, 100, 200, \ldots, 1000$.

   We deliberately did not specify exactly how to generate a random subspace and how to generate random points in the subspace; please think about a sensible choice yourself.

Problem 1 (denoising). In the first chapter, we saw that signal models are central for solving inverse problems. Here, we consider a denoising problem and show that if a n-dimensional signal lies in a k-dimensional subspace, we can remove a fraction of $\frac{n-k}{k}$ of additive Gaussian noise. Consider denoising problem, where we are given a noisy measurement of a signal $\mathbf{x}^*$ as

$$\mathbf{y} = \mathbf{x}^* + \mathbf{z}.$$

We assume that $\mathbf{x}^* \in \mathbb{R}^n$ is a signal that lies in a k-dimensional subspace, and $\mathbf{z}$ is zero-mean Gaussian noise with co-variance matrix $(\sigma^2/n)\mathbf{I}$. Let $\mathbf{U} \in \mathbb{R}^{n\times k}$ be an orthonormal basis of the signal subspace. We denoise the signal by projecting the measurement onto the subspace, i.e., we consider the estimate $\hat{\mathbf{x}} = \mathbf{U}\mathbf{U}^T\mathbf{y}$.

1. Show that
$$\mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2\right] = \sigma^2\frac{k}{n},$$
where expectation is over the random noise $\mathbf{z}$.

   Hint: Recall that if $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a unitary matrix (i.e., a matrix with orthonormal columns) and $\mathbf{z}$ has iid, zero-mean Gaussian entries, then $\mathbf{V}\mathbf{z}$ has the same distribution as $\mathbf{z}$.

$x^* \in \mathbb{R}^n$, $x^* \in \text{span}\{U\}$ $\quad z \in \mathbb{R}^n, z \sim \mathcal{N}(0, \frac{\sigma^2}{n} I)$

The rule is:

if A is the left inverse of B $\Rightarrow AB = I$

then BA is the projection onto the range(B)

Assume $B = U$, $A = U^T$ $\Rightarrow \underline{BA = UU^T}$

$\hat{x} = UU^T y$

$$\|\hat{x} - x^*\|_2^2 = \|UU^T y - x^*\|_2^2$$
$$= \|UU^T(x^* + z) - x^*\|_2^2$$
$$= \|\underbrace{(UU^T - I_n)x^*}_{a} + z\|_2^2$$

$$\mathbb{E}\left[\|a+z\|_2^2\right] = \mathbb{E}\left[(a+z)^T(a+z)\right]$$
$$= \mathbb{E}\left[a^Ta + a^Tz + z^Ta + z^Tz\right]$$

$$= \underbrace{E[a^T a]}_{\substack{(*)\\ =0}} + \underbrace{2E[a^T z]}_{\substack{(**)\\ =0}} + \underbrace{E[z^T z]}_{\text{Variance for zero mean}}$$

$$\Rightarrow tr\left(\frac{\sigma^2}{n} I\right)$$

$$= \frac{\sigma^2}{n} \cdot k$$

$$(*) : E\left[x^{*T}(UU^T - I)^T (UU^T - I) x^*\right]$$

$$= E\left[x^{*T}\underbrace{(UU^T - I)(UU^T - I)}x^*\right]$$

$$\underbrace{UU^TUU^T}_{UU^T} - UU^T - UU^T + I$$

$$= E\left[x^{*T}(I - UU^T)x^*\right] = E\left[x^{*T} I x^* - x^{*T} UU^T x^*\right]$$

$$= E\left[x^{*T}x^*\right] - E\left[x^{*T}UU^T x^*\right] = 0$$

orthonormal matrix doesn't change the mean

$$(**) : E\left[x^{*T}(UU^T - I)^T z\right] = E\left[x^{*T}(UU^T - I) z\right]$$

$$E\left[x^{*T}UU^T z\right] - E\left[x^{*T} z\right] = 0$$

orthonormal matrix ~~does~~ does not change the mean

2. Does the algorithm $\hat{\mathbf{x}} = \mathbf{U}\mathbf{U}^T\mathbf{y}$ denoise more or less if the dimension of the subspace becomes smaller, and what is your intuition on whether a better algorithm exists?

Intuitively, if the subspace has less dimensions then, the vector is supposed be easier to represent. By easier, I mean that the range of the subspace will be smaller, thus it will be easier to represent than a vector in a subspace, whose range is greater. Furthermore, in the formula it is seen that the mean of the error gets smaller with a small k.

On the other hand, I can also argue that if k gets greater, then we will have a more generic representation of x^*. Therefore, the values it can take increases.

3. Next, we study this denoising algorithm numerically (ideally with python in a jupyter notebook using the libary numpy; if you are not familiar with those, this exercise is a good exercise to familiarize yourself).

   Towards this goal, generate a random k-dimensional subspace in $\mathbb{R}^{1000}$, and generate 500 random points in that subspace. Next, denoise each of those data points with the method above, and plot the average of the mean-squared error $\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2/\|\mathbf{x}^*\|_2^2$ along with corresponding standard deviations as error bar for different values of $k = 1, 100, 200, \ldots, 1000$.

   We deliberately did not specify exactly how to generate a random subspace and how to generate random points in the subspace; please think about a sensible choice yourself.

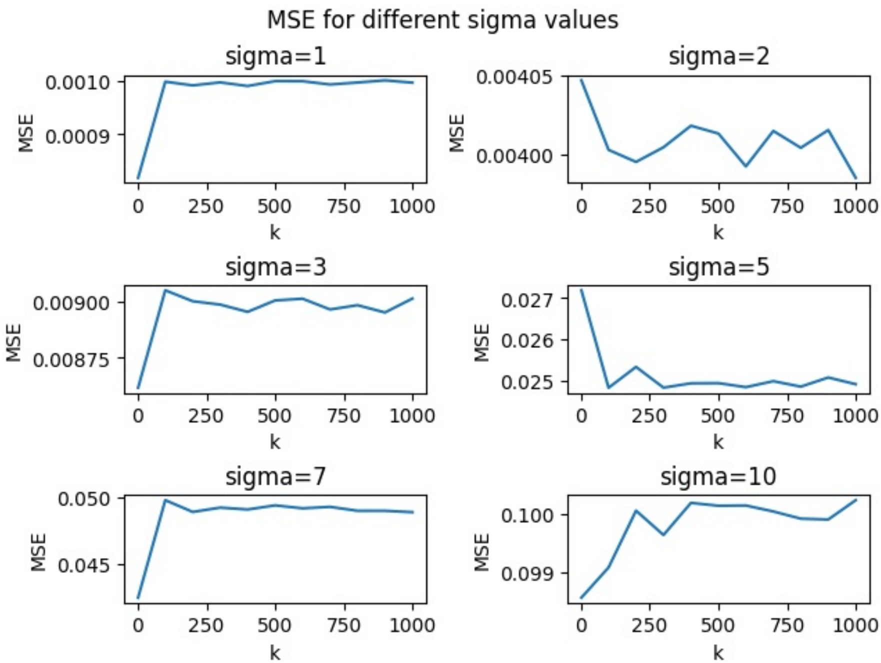
$0{,}1 \cdot u_1 + 0{,}1 \cdot u_2$

$-0{,}1 \cdot u_1 + (-0{,}1) \cdot u_2$

$0\, u_1 + 1 \cdot u_2$

$U = [u_1 \;,\; u_2]$

$$U \cdot C^T = \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \begin{bmatrix} 0{,}1 & -0{,}1 & 0 \\ 0{,}1 & -0{,}1 & 1 \end{bmatrix}$$



I cannot explain the behaviour with sigma = 2 and sigma = 5

With increasing sigma, the MSE gets larger. It fits to the proof above. However, I couldn't figure out to get a stable code, in the sense that the course of MSE changes for some sigma values.