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.