ELEN 6886 Homework 1

Due Thursday, February 22, 11:59 PM, via CourseWorks

Exercise 1. (Random vectors and subspaces.)

a) Let $\boldsymbol{x} \in \mathbb{R}^n$ be a Gaussian random vector with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{I}$. Prove that, for any orthogonal matrix $\boldsymbol{U}$ (i.e. an $n \times n$ matrix satisfying $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$), one has that $\boldsymbol{U}\boldsymbol{x}$ and $\boldsymbol{x}$ are identically distributed.

b) Now consider the sphere, i.e. the set $\mathbb{S}^{n-1} = \{\boldsymbol{q} \in \mathbb{R}^n : \|\boldsymbol{q}\|_2 = 1\}$. Argue that if $\boldsymbol{x} \in \mathbb{R}^n$ is a Gaussian random vector with zero mean and identity covariance, the distribution of

$$\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}$$

is invariant to rotation, i.e., $\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} \equiv_d \frac{\boldsymbol{U}\boldsymbol{x}}{\|\boldsymbol{U}\boldsymbol{x}\|_2}$ for any orthogonal matrix $\boldsymbol{U}$.

This can be interpreted as saying that $\boldsymbol{x}$ is distributed according to the (rotationally) invariant probability measure μ on the sphere (the so-called *Haar measure*). Equivalently, $\boldsymbol{x}$ is uniformly distributed on the sphere.

c) Perhaps surprisingly, we can generalize the algorithm from part (b) to produce uniform samples from the set of subspaces of $\mathbb{R}^n$ with dimension k, for any $2 \leq k \leq n$. In particular, define the *Stiefel manifold with parameter* p to be the set

$$\mathrm{St}(n, p) = \{\boldsymbol{U} \in \mathbb{R}^{n \times p} : \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_p\}$$

where p is an integer satisfying $1 \leq p \leq n$. For each such p, one can define a Haar measure μ_p on $\mathrm{St}(n, p)$ which is invariant under action of the set's symmetry group (i.e. left multiplication by $n \times n$ orthogonal matrices, and right multiplication by $p \times p$ orthogonal matrices). Prove that, if $\boldsymbol{N} \in \mathbb{R}^{n \times p}$ is a matrix with i.i.d. zero-mean, unit-variance Gaussian entries, then

$$\mathtt{orth}(\boldsymbol{N}) \sim \mu_p,$$

where $\mathtt{orth} : \mathbb{R}^{n \times p} \to \mathbb{R}^{n \times p}$ denotes the Gram-Schmidt orthonormalization process.

d) Conclude an algorithm for sampling uniformly random subspaces of dimension k in $\mathbb{R}^n$ for each integer $1 \leq k \leq n$.

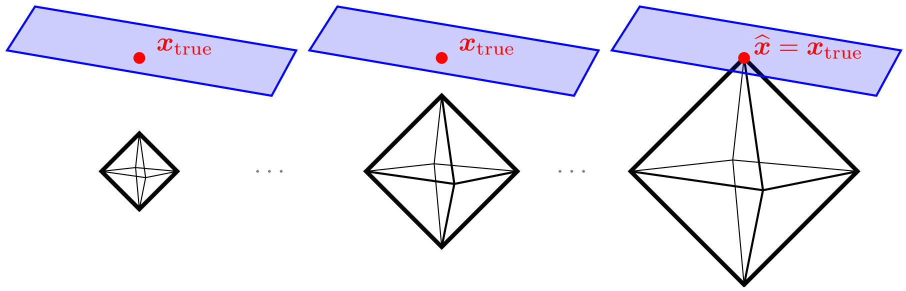

Figure 1: Sketch of the ℓ^1 norm ball being enlarged until it becomes tangent to an affine set in $\mathbb{R}^3$.

Exercise 2. (Coefficient space picture for ℓ^1 minimization.) In this exercise, we will explore the "first picture" we drew to build intuition for ℓ^1 minimization (see figure 1) and implement an alternate algorithm for ℓ^1 minimization based on the idea behind it.

a) In MATLAB (or your favorite alternate programming language), reproduce figure 1 by plotting the cross polytope B_1 and an affine set in $\mathbb{R}^3$. Write your code so that you can painlessly vary the diameter of the cross polytope, and display, roughly, the instant at which the cross polytope first touches the affine set.

b) Next, we will develop an algorithm to find a point in the intersection of the c-scaled cross polytope cB_1 and an affine set $A \subseteq \mathbb{R}^n$, for any $c > 0$. For this, we will use an algorithm due to von Neumann, which is sometimes called the "projections onto convex sets" (POCS) algorithm: see Algorithm 1. The algorithm alternates between projecting onto cB_1 and projecting onto the affine set A. This algorithm can be shown to converge linearly to a point in the intersection $cB_1 \cap A$, whenever the intersection is nonempty.

   i) We already learned how to project onto an affine set A in lecture. We can apply the POCS algorithm once we can project onto the cross polytope. Devise a reasonably-efficient algorithm to compute the projection

$$p_{cB_1}(\mathbf{z}) = \arg\min_{\|\mathbf{x}\|_1 \le c} \|\mathbf{x} - \mathbf{z}\|_2^2.$$

   onto any scaling cB_1 of the cross polytope $\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_1 \le c\}$, with $c \ge 0$.

   ii) Assuming a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a measurement $\boldsymbol{y} \in \mathbb{R}^m$ are given, devise an ℓ^1 minimization algorithm based on algorithm 1 and the sketch in figure 1. Experiment with algorithm 1 in order to fill in the missing stopping criterion.

   iii) Compare the performance of your algorithm to that of the projected subgradient algorithm developed in the lecture and given in the demo folder on the Courseworks. When $\boldsymbol{y}$ is generated as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, with $\boldsymbol{x}$ truly sparse, which algorithm returns a more accurate solution? What about time to converge and computational costs?

c) (Optional) Again specializing to $\mathbb{R}^3$, use the algorithm from part (b.ii) to create an animation of the expanding process shown in figure 1.

Algorithm 1 Projections onto convex sets: $S, C \subseteq \mathbb{R}^n$ closed and convex

function POCS(S, C) ▷ Returns a point in $S \cap C$ when $S \cap C \neq \varnothing$

  $p_S \leftarrow \boldsymbol{x} \mapsto \arg\min_{\boldsymbol{x}' \in S} \|\boldsymbol{x}' - \boldsymbol{x}\|_2$ ▷ Projection map for set S

  $p_C \leftarrow \boldsymbol{y} \mapsto \arg\min_{\boldsymbol{y}' \in C} \|\boldsymbol{y}' - \boldsymbol{y}\|_2$ ▷ Projection map for set C

  $\boldsymbol{z} \leftarrow \boldsymbol{0}$

  while a suitable test for convergence fails **do**

    $\boldsymbol{z} \leftarrow p_S(\boldsymbol{z})$

    $\boldsymbol{z} \leftarrow p_C(\boldsymbol{z})$

  end while

end function

Exercise 3. (Various sampling patterns in MR reconstruction.) In this problem, we will specialize our projected subgradient method for MR image reconstruction, and investigate the effect of the sampling pattern on reconstruction.

a) Recall that MRI systems acquire a discretized version of the 2D discrete Fourier transform (DFT) of the underlying continuous spatial profile, and exploiting sparsity amounts to subsampling the DFT in a particular way and using a reconstruction procedure such as ℓ^1 minimization to recover the original spatial profile.

   In particular, a reasonable measurement model for the MRI problem can be given as

$$\boldsymbol{y} = \boldsymbol{S}_U \circ \mathcal{F}(\boldsymbol{Z}),$$

   where $\boldsymbol{Z} \in \mathbb{R}^{N \times N}$ is the underlying spatial profile, $\mathcal{F} : \mathbb{C}^{N \times N} \to \mathbb{C}^{N \times N}$ is the two-dimensional discrete Fourier transform operator, U is a subset of $[n] \times [n]$ that can be either random or deterministic, and $\boldsymbol{S}_U : \mathbb{C}^{N \times N} \to \mathbb{C}^{|U|}$ denotes projection onto the elements of U, with concatenation performed in increasing lexicographic order.

   For this model, assuming you are given a draw of the subset U and a corresponding measurement $\boldsymbol{y}$, give a mathematical description of the form the projected subgradient algorithm for ℓ^1 minimization takes for recovery of $\boldsymbol{Z}$. (Hint: show that $\boldsymbol{S}_U \circ \mathcal{F}$ is linear, and compute its adjoint map; then compute the projection operator for $\ker(\boldsymbol{S}_U \circ \mathcal{F})$ as in the lecture.)

b) Implement your algorithm for the following forms of U:

   i) Bernoulli-distributed entries: each index pair $(i, j) \in [n] \times [n]$ is included in U independently with probability $\theta \in (0, 1)$.

   ii) Variable density random sampling: each index pair $(i, j) \in [n] \times [n]$ is included in U independently with probability proportional to its distance from the "center" of the grid $[n] \times [n]$. (Hint: this is a special case of (i), with probabilities θ that depend on spatial position in a rotationally-symmetric manner.)

   iii) Radial sampling: a deterministic sampling method in which only index pairs (i, j) lying on a discretization of a set of lines through the origin are included (see Figure 5.7 in the notes on MRI).

c) Compare the results for the different types of U you have implemented. Which algorithm performs the best as the measurement density $|U|$ is varied? How do the algorithms compare to the one from the demo, where the top-magnitude wavelet coefficients are preserved? Can you interpret performance in terms of the coherence of the measurement map, given that $\boldsymbol{Z}$ is sparse in a wavelet basis? Which of the sampling patterns do you expect to be most easily implementable in hardware?