Convex Optimization 10-725/36-725
Homework 5, due Nov 26

Instructions:

- You must complete Problems 1–3 and **either Problem 4 or Problem 5** (your choice between the two).
- When you submit the homework, upload a single PDF (e.g., produced by LaTeX, or scanned handwritten exercises) for the solution of each problem separately, to blackboard. You should your name at the top of each file, except for the first problem. **Your solution to Problem 1 (mastery set) should appear completely anonymous to a reader.**

1 Mastery set [25 points]

A. [3+3+4] What are the subgradients of the function

$$f(x) = \max_{i=1,\dots m} a_i^T x + b_i?$$

What about, for f_i, $i = 1, \dots m$, convex,

$$f(x) = \max_{i=1,\dots m} f_i(A_i x)?$$

And finally

$$f(x) = \max_{i=1,\dots m} \|A_i x\|_{p_i},$$

where each $p_i \geq 1$?

B. [2+2] Let C be a convex set. Consider the projection of x onto C; prove that x projects to a unique element of C. (Hint: write as an optimization problem.) What happens when C is nonconvex? (Hint: draw a picture.)

D. [1] Can a linear program ever be nonconvex? Why or why not?

E. [2+2+2+4] Derive the conjugates of $f(x) = ax + b$, $f(x) = e^x$, and $f(x) = x \log x$. Let $\|\cdot\|$ be an arbitrary norm. Derive the conjugate of $f(x) = \|x\|^2/2$.

Bonus. [5] Is the projection operator onto a convex set differentiable? Why or why not?

2 PSD Matrices (Adona) [25 points]

Part A [15 points]: Basic properties of PSD matrices

Assume $X \in \mathbb{R}^{n \times n}$ is a symmetric PSD matrix.

1. **[3 points]** Let $I \subseteq \{1...n\}$ be an index set. Prove that X_I is also PSD for all I, where X_I is the submatrix formed by choosing all rows and columns from index-set I.
2. **[3 points]** Using the above, prove that for any i, j, we have $X_{ii}X_{jj} \geq X_{ij}^2$. As a corollary, prove the property (from class) that $X_{ii} = 0$ implies that the entire i-th row and column must be zero.
3. **[3 points]** Using the eigenvalue decomposition, prove that the determinant of X is the product of its eigenvalues.
4. **[6 points]** Prove that if $X := \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ then $X \succ 0$ is psd iff $A \succ 0$ and $C - B^T A^{-1} B \succ 0$.

   HINT: Show that if $X = Y^T Z Y$ for some invertible matrix Y, then $X > 0$ iff $Z > 0$.

   Now, can you decompose X as

   $$X = Y^T \begin{pmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{pmatrix} Y$$

   for some matrix Y?

Part B [10 points]: Formulating problems as SDPs

Using the above property (and others), formulate the following problems as SDP.

1. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph. Assume $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$. Let $w_e > 0$ denote the weight of edge e and $T = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ be a set of node pairs. Consider the

following problem:

$$\begin{aligned}
&\min_x \frac{1}{4} \sum_{(u,v)\in\mathcal{E}} w_e \|x_u - x_v\|^2 \\
&\|x_u - x_v\|^2 + \|x_v - x_w\|^2 \geq \|x_u - x_w\|^2 \ \forall u, v, w \in \mathcal{V} \\
&\|x_{s_i} - x_{t_i}\|^2 = 4 \ \forall (s_i, t_i) \in T \\
&\|x_u\|^2 = 1 \ \forall u \in \mathcal{V}
\end{aligned}$$

where $x_u \in \mathbb{R}^n$.

2. Consider the following set

$$\mathcal{P} = \{P \in \mathbb{R}^{n\times m} : \|p_i - c_i\| \leq d_i\}.$$

where $c_i \in \mathbb{R}^n$ denotes the i^{th} column of C and $d_i \in \mathbb{R}^+$. Let B be a given $n \times m$ matrix with full column rank. Assume $P^\top B + B^\top P$ is positive definite for all $P \in \mathcal{P}$. Consider the following problem:

$$\begin{aligned}
&\min_P \ \operatorname{tr}\left(P(I_m + P^\top B + B^\top P)^{-1} P^\top\right) \\
&\text{subject to } P \in \mathcal{P}.
\end{aligned}$$

Here I_m denotes the $m \times m$ identity matrix.

3 Mixed Images (25 Points)

Due to a terrible mistake four of my images got mixed on my hardrive. You can see them in Figure 1, and they can be downloaded from `http://www.stat.cmu.edu/~ryantibs/convexopt/homeworks/mixedimages.zip`. The images are black and white, and their size is 600×1000 pixels. I know that the mixing process was linear, i.e. the pixel values of the i^{th} mixed image at location (x, y) were generated by the following equation

$$MixedImage_i(x, y) = \sum_{j=1}^{4} \alpha_{ij} OriginalImage_j(x, y), \quad (i = 1, \ldots, 4).$$

I do not remember α_{ij}, and I did not keep the original images either.

Your task is to implement the FastICA algorithm and estimate the original images. You need to submit your implementation and the estimated images. More information about the FastICA can be found here: `http://www.cs.helsinki.fi/u/ahyvarin/papers/NN00new.pdf`. You can choose its parameters as you wish, I just need my images.

In Matlab you can use the "imread" and "imagesc" commands to read and display images.

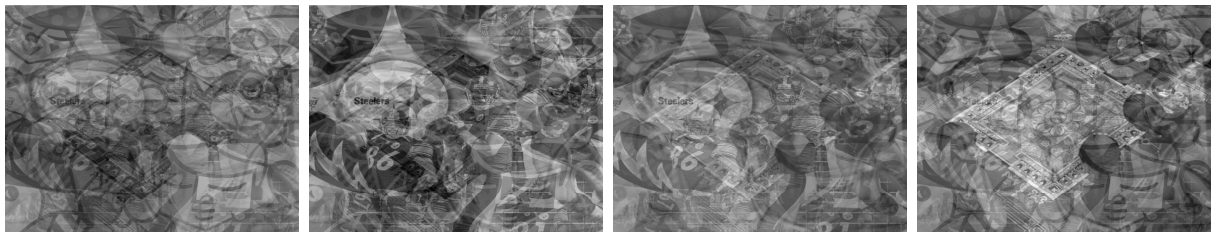

Figure 1: Mixed images.

4 Advanced Theory

Part A [15 points] Maximum Likelihood Estimation for Multivariate Gaussians

A multivariate Gaussian distribution parametrized by the mean $\mu \in \mathbb{R}^d$ and the covariance matrix $\Sigma \in \mathcal{S}^d_{++}$ is:

$$\mathcal{N}(\mathrm{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det \Sigma}} \exp\left\{-\frac{1}{2}(\mathrm{x}-\mu)^\top \Sigma^{-1}(\mathrm{x}-\mu)\right\}$$

1. [3 points] Given n independent draws $\mathrm{x}_i \in \mathbb{R}^d, i = 1, \ldots, n$, derive an explicit form for the log likelihood:

$$\ell(\mu, \Sigma) = \sum_{i=1}^{n} \log \mathcal{N}(\mathrm{x}_i; \mu, \Sigma).$$

Notice the dependency on observations is omitted.

2. [5 points] Using matrix differential operators, find out the derivative of $\ell(\mu, \Sigma)$ in terms of μ and Σ.

3. [1+3+1 points] Define the following:

$$g(\Sigma) = \log\det(\Sigma) \quad , \quad h_i(\mu, \Sigma) = (\mathrm{x}_i - \mu)^\top \Sigma^{-1}(\mathrm{x}_i - \mu).$$

Show that $g(\Sigma)$ is a concave function. Use a similar technique to show that $h_i(\mu, \Sigma)$ is convex on the concatenation of joint $(\mu_1, \ldots, \mu_d, \Sigma_{11}, \ldots, \Sigma_{d1}, \ldots, \Sigma_{dd})^\top$ on its feasible region. What can you conclude about the Gaussian log likelihood, as a function μ and Σ? Is it concave, convex, neither?

[Hint: check out the proof for log concavity of the determinant of a positive definite matrix in Chapter 3.1.5, Boyd & Vandenberghe. Convex Optimization.]

4. [2 points] What is the maximum likelihood estimator for the multivariate Gaussian distribution?

Part B [10 points]

Recall that a zero-mean, P-dimensional Gaussian distribution is defined by its covariance matrix Σ. So 'learning' such a distribution from some iid samples $X = \{X^{(1)}, \ldots, X^{(N)}\}$ amounts to an estimate $\widehat{\Sigma}$ of Σ. Since we have an i.i.d. sample from a known family of distributions, a natural approach is to pick the $\widehat{\Sigma}$ which maximizes the likelihood of the sample. However, if $N < P$, the estimate is not well-defined. Further, even if $N \geq P$ but N and P are of comparable size, the maximum likelihood estimate can have high variability, leading to bad predictive performance. Since P is big, we are inclined to restrict attention to sparse distributions, for some appropriate notion of sparsity. A tempting, but unrealistic, kind of sparsity is independence of a large number of the features i and j: $\Sigma_{ij} = E_{X \sim N(0,\Sigma)}(X_i X_j) = 0$. A more realistic kind of sparsity is conditional independence of features i and j given the other features. Since the distribution is Gaussian, this happens when $\Sigma^{-1}_{ij} = 0$. Since our sparsity belief/assumption concerns Σ^{-1}, let's orient our notation around that, starting with the log-likelihood function, which can be shown to be:

$$\ell(K) = \log \det(K) - \operatorname{tr}(SK)$$

where K is a symmetric positive semidefinite $P \times P$ matrix meant to estimate Σ^{-1}, S is the empirical covariance matrix $S = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})(X_i - \bar{X})^T$ and the sample mean $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$. The maximizer of $\ell(K)$ is generally ill-defined and non-sparse, and so one considers the following ℓ_1 penalized estimator (and the problem is lovingly called the graphical lasso):

$$\min_{K \succ 0} - \log \det(K) + Tr(SK) + \lambda \sum_{i \neq j} |K_{ij}|$$

3 points Write down the subgradient method's update equations.

4 points Write down the proximal gradient update equations.

3 points Can you accelerate this? If yes, write down the update equations. If no, why not?

5 Graph Lasso (25 points)

Please read Q4(b) for an introduction to the graph lasso problem. There, you will prove that if you want to learn a sparse inverse covariance matrix K from data $X \in \mathbb{R}^{N \times P}$, then maximizing the likelihood corresponded to solving the optimization problem

$$\min_{K \succ 0} - \log \det(K) + Tr(SK) + \lambda \sum_{i \neq j} |K_{ij}|$$

where $S \in \mathbb{R}^{P \times P}$ is the sample covariance matrix (assuming that you don't penalize diagonal elements). In the previous question, you could use your mastery of matrix differentials to derive a good primal algorithm to solve this problem quickly. If you didn't do it, think about it for a second now.

First, you will be the TA and generate data for your problem (to know how to run your own simulations, and to know what it's like to TA this class). We need to generate random draws from a gaussian distribution that has a sparse inverse covariance matrix. How do we come up with a sparse PSD matrix that is not just any PSD matrix, but also its inverse is consistent with a (probably dense) covariance matrix that encodes correlations? We'll do some hacky stuff.

Choose a reasonable value for P (and N) that you think your algorithm will scale to without making your life miserable (for eg: N=250, P=25 is probably conservative). Generate N points from a P-dimensional standard gaussian, and calculate the empirical inverse covariance matrix. Threshold this entrywise at some reasonable value γ, to get a sparse matrix iT. This will be the unknown (to your algorithm) truth. Invert this to get a (probably dense) covariance matrix T. Now, generate N points from a P-dimensional gaussian with covariance matrix T. Call its sample inverse covariance iS, and its sample covariance S (either the datapoints or S is the input to your algorithm). Caution: Thresholding at γ is not guaranteed to preserve PSD-ness, ie iT may not be PSD. In this case, you can try again, or you can add a small constant times the identity matrix to make it PSD.

3 points Submit your data-generating code as text in your answer. What N, P, γ did you use?

2 points Plot the sparsity pattern of zero elements vs non-zero elements in iT and iS using *imagesc*. This are respectively where the data came from, and your sample estimator from the data.

In this question, we will derive a dual for this problem, and perform dual ascent. Remember that a lot of the matrices are symmetric by definition.

1 points Introduce a new constraint (name the new variable Z), as we have often done and write down the Lagrangian (name the dual variable W).

4 points Show clearly (not lengthily, but convincingly) that the dual is

$$\max_{W \in \mathbb{R}^{p \times p}} \log\det(S+W) \quad \text{s.t.} \quad W_{ii} = 0 \text{ and } |W_{ij}| \leq \lambda \forall i \neq j$$

We will use projected gradient ascent with backtracking line search to perform dual ascent. ALTERNATELY, you can run any other algorithm of your choice on the dual, but submit a solution that describes your algorithm and has the same amount of detail as below.

You will run the algorithm while the duality gap is smaller than ϵ or till the iteration count crosses $MAXITER$ (choose a reasonable value for both! For eg: for the gap, 10^{-8} is aiming too high but 10^{-1} is trivial).

4 points Show that the duality gap at any dual feasible point W is $\eta = Tr(SK) + \lambda \sum_{i \neq j} |K_{ij}| - n$ for an appropriate K. What did you choose for ϵ or $MAXITER$?

1 points Why is the suggested algorithm better suited to the dual than the primal?

Run a simple backtracking line search with a reasonable initial stepsize t_0 to find *any* point that is better than the current point, and cutting your stepsize by $\delta < 1$ at every step till you do so (remember that you are *ascending*). Whenever you evaluate the function while backtracking, it needs to be at a feasible point, hence you need to take a (possibly large) step and then project and then evaluate. Again $t_0 = 20$ or $\delta = 0.001$ are probably not great choices, just be reasonable.

4 points Implement your algorithm! Submit your code as text in the answer. Avoid using unnecessary functions. What did you choose for t_0, δ?

3 points Plot the sparsity pattern of zero elements vs non-zero elements in your final inverse covariance matrix using *imagesc* for a *good* value of λ. (Warning: MATLAB often faces rounding issues - many of the elements may be zero upto very high precision but appear nonzero in your plot - threshold them at some tiny value to avoid this problem)

3 points Choose any two other sensible and revealing values of λ and provide the sparsity pattern - does it make sense to you? How long did your algorithm take to run, in terms of time and number of iterations?

This problem is intentionally left open ended, and hence there is no single correct answer, but plenty of reasonable answers. If your simulated example and parameter choices don't demonstrate that the algorithm works, then you should go back and think of what you did wrong. Too few samples and hence S is hardly informative? Algorithm hitting MAXITER and hence too conservative steps? Too low γ and too high λ simultaneously?

It is the last advanced implementation question and while the implementation itself is not hard at all, but we wanted you to think about what kind of choices you would make, because in real-life there won't be a TA around to hand you tuning parameters, and you must be your own TA!