COMS 4772 Fall 2016 Homework 2
Due Friday, October 28

Instructions:

- The usual homework policies (`http://www.cs.columbia.edu/~djhsu/coms4772-f16/about.html`) are, of course, in effect.
- Using this LaTeX template will be helpful for grading purposes.

Problem 1 (25 points)**.** Let $\boldsymbol{X}$ be a random vector in $\mathbb{R}^d$ whose distribution is a mixture of k spherical Gaussians:

$$\boldsymbol{X} \sim \pi_1 \mathrm{N}(\boldsymbol{\mu}_1, \sigma_1^2 \boldsymbol{I}) + \pi_2 \mathrm{N}(\boldsymbol{\mu}_2, \sigma_2^2 \boldsymbol{I}) + \cdots + \pi_k \mathrm{N}(\boldsymbol{\mu}_k, \sigma_k^2 \boldsymbol{I}) .$$

For any set $C \subset \mathbb{R}^d$, define

$$\mathrm{cost}(C) := \mathbb{E}\left[\min_{\boldsymbol{y}\in C} \|\boldsymbol{X}-\boldsymbol{y}\|_2^2\right] .$$

Let $M := \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k\}$. Prove that if $k < e^{d/2}$, then

$$\mathrm{cost}(M) \le \frac{1}{1-\frac{2\ln(k)}{d}} \cdot \min_{\substack{C\subset\mathbb{R}^d:\\ |C|\le k}} \mathrm{cost}(C) .$$

Solution.
We start by finding the upper bound of $\mathrm{cost}(M)$:

$$\begin{aligned}
\mathbb{E}\left[\min_{\boldsymbol{y}\in M} \|\boldsymbol{X}-\boldsymbol{y}\|_2^2\right] &= \mathbb{E}\left[\min_{\boldsymbol{y}\in M} (\boldsymbol{X}-\boldsymbol{y})^T(\boldsymbol{X}-\boldsymbol{y})\right] \\
&= \mathbb{E}\left[\min_{\boldsymbol{y}\in M} \boldsymbol{X}^T\boldsymbol{X}_i - 2\boldsymbol{X}^T y + \boldsymbol{y}^T\boldsymbol{y}\right] \\
&= \mathbb{E}\left[\boldsymbol{X}^T\boldsymbol{X}\right] - \mathbb{E}\left[\min_{\boldsymbol{y}\in M} -2\boldsymbol{X}^T y + \boldsymbol{y}^T\boldsymbol{y}\right] \\
&= \mathbb{E}\left[\boldsymbol{X}^T\boldsymbol{X}\right] + \mathbb{E}\left[\max_{\boldsymbol{y}\in M} (2\boldsymbol{X}-\boldsymbol{y})^T\boldsymbol{y}\right] \\
&= \mathbb{E}\left[\boldsymbol{X}^T\boldsymbol{X}\right] + \frac{1}{\lambda}\ln\exp\mathbb{E}\left[\max_{\boldsymbol{y}\in M} \lambda(2\boldsymbol{X}-\boldsymbol{y})^T\boldsymbol{y}\right] \\
&\le \mathbb{E}\left[\boldsymbol{X}^T\boldsymbol{X}\right] + \frac{1}{\lambda}\ln\mathbb{E}\left[\max_{\boldsymbol{y}\in M} \exp\lambda(2\boldsymbol{X}-\boldsymbol{y})^T\boldsymbol{y}\right] \\
&\le \mathbb{E}\left[\boldsymbol{X}^T\boldsymbol{X}\right] + \frac{1}{\lambda}\ln\sum_{i=1}^{k}\mathbb{E}\left[\exp\lambda(2\boldsymbol{X}-\boldsymbol{y})^T\boldsymbol{y}_i\right] \\
&\le E\left[\boldsymbol{X}^T\boldsymbol{X}\right] + \frac{1}{\lambda}\ln\sum_{i=1}^{k}\mathbb{E}\left[\exp\lambda(2\boldsymbol{X}_k-\boldsymbol{y})^T\boldsymbol{y}_i\right] \text{ where } \boldsymbol{X}_k = \max_{\boldsymbol{y}\in M}\boldsymbol{y} \\
&\le E\left[\boldsymbol{X}^T\boldsymbol{X}\right] + \frac{1}{\lambda}\ln\sum_{i=1}^{k}\mathbb{E}\left[\exp\lambda(2\boldsymbol{X}_k^T\boldsymbol{y})\right] \\
&\le E\left[\boldsymbol{X}^T\boldsymbol{X}\right] + \frac{1}{\lambda}\ln\sum_{i=1}^{k}\mathbb{E}\left[\exp 2\lambda(\boldsymbol{X}_k^T\boldsymbol{y})\right] \\
&\le E\left[\boldsymbol{X}^T\boldsymbol{X}\right] + \frac{1}{\lambda}\ln\sum_{i=1}^{k}\exp(\sigma_k\lambda^2)tr(\boldsymbol{y})
\end{aligned}$$

With $\lambda = \sqrt{\frac{\ln k}{\sigma_k tr(\boldsymbol{y})}}$, $\mathbb{E}\left[\min_{\boldsymbol{y}\in M}\|\boldsymbol{X}-\boldsymbol{y}\|_2^2\right] \le E\left[\boldsymbol{X}^T\boldsymbol{X}\right] + 2\sqrt{\sigma_k tr(\boldsymbol{y})\ln k}$.
The lower bound of the $\mathrm{cost}(C)$ occurs when $\boldsymbol{y} = \mathbb{E}[\boldsymbol{X}]$ and hence $\mathbb{E}[\boldsymbol{X}^T\boldsymbol{X}] - \mathbb{E}[\boldsymbol{X}^T]\mathbb{E}[\boldsymbol{X}] \le \mathrm{cost}(c)$.
□

Problem 2 (25 points)**.** Suppose $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times d}$ each have rank d. Give unambiguous pseudocode for an algorithm that, when given $\boldsymbol{A}$ and $\boldsymbol{B}$ as inputs, finds all solutions $\boldsymbol{v} \in S^{d-1}$ satisfying

$$\exists \lambda \in \mathbb{R} \setminus \{0\} \text{ s.t. } \boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{v} \;=\; \lambda \boldsymbol{B}^\top \boldsymbol{B} \boldsymbol{v}\,.$$

If there is an entire subspace of solutions, the algorithm just needs to return an orthonormal basis for this subspace. Your pseudocode can use things like SVD, Gram-Schmidt, etc. as black-box subroutines. Prove that the algorithm is correct.

Solution.
Input: $\boldsymbol{A}$, $\boldsymbol{B}$
Output: All solutions $\boldsymbol{v} \in S^{d-1}$ satisfying $\exists \lambda \in \mathbb{R} \setminus \{0\}$ s.t. $\boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{v} \;=\; \lambda \boldsymbol{B}^\top \boldsymbol{B} \boldsymbol{v}$

Begin
 Calculate $\boldsymbol{A}^T \boldsymbol{A}$
 Calculate $\boldsymbol{B}^T \boldsymbol{B}$
 Invert $\boldsymbol{B}^T \boldsymbol{B}$ and set it as $\boldsymbol{C}$
 Take the product of $\boldsymbol{C}$ and $\boldsymbol{A}^T \boldsymbol{A}$ and set it as $\boldsymbol{D}$
 Perform eigendecompoisiton on $\boldsymbol{D}$ to obtain $\boldsymbol{D} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{U}^T$
 Normalise and return $\boldsymbol{U}$
End

Steps 1 and 2 will be always be valid for matrices of any size. As $\boldsymbol{B}$ is of rank d, $\boldsymbol{B}^T \boldsymbol{B}$ is also of rank d and a $d \times d$ matrix, hence $\boldsymbol{B}^T \boldsymbol{B}$ is not rank deficient and is invertible so step 3 is valid. Since both $\boldsymbol{C}$ and $\boldsymbol{A}^T \boldsymbol{A}$ are $d \times d$ matrices, their product can be computed which verifies step 4. $\boldsymbol{D}$ is a square $d \times d$ matrix of rank d, we can perform eigendecomposition to obtain $\boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{U}^T$ where $U^T = U^{-1}$. Hence U is orthogonal and forms a basis for the subspace of the solutions. After normalising U, we obtain a orthonormal basis for the subspace.

□

Problem 3 (25 points)**.** Let $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ be a data matrix whose rows are $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n \in \mathbb{R}^d$. Let $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ be the matrix whose (i,j)-th entry is the squared Euclidean distance $D_{i,j} = \left\|\boldsymbol{a}_i - \boldsymbol{a}_j\right\|_2^2$. Suppose you are given the squared Euclidean distance matrix $\boldsymbol{D}$ as input, and you are asked to recover the set of original points $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$ up to some translation. *You do not have access to the original data matrix $\boldsymbol{A}$.*

(a) Let $\boldsymbol{s} \in \mathbb{R}^n$ be the vector whose i-th entry is $\|\boldsymbol{a}_i\|_2^2$. Prove that $\boldsymbol{D} = \boldsymbol{s}\boldsymbol{1}_n^\top - 2\boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{1}_n\boldsymbol{s}^\top$, where $\boldsymbol{1}_n \in \mathbb{R}^n$ is the all-ones vector.

(b) Let $\boldsymbol{\Pi} \in \mathbb{R}^{n\times n}$ be the orthogonal projector for the $(n-1)$-dimensional subspace

$$\left\{\boldsymbol{x} \in \mathbb{R}^n : \langle \boldsymbol{1}_n, \boldsymbol{x}\rangle = 0\right\} .$$

Prove that $-(1/2)\boldsymbol{\Pi}\boldsymbol{D}\boldsymbol{\Pi} = \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{\Pi}$.

(c) Explain how to determine points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ from $\boldsymbol{D}$ such that:

- $D_{i,j} = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2$ for all $i, j \in [n]$; and
- $\sum_{i=1}^n \boldsymbol{x}_i = \boldsymbol{0}$.

(You may assume that you are told the original dimension d.)

(d) *Optional.* Suppose the matrix $\boldsymbol{D}$ is corrupted (say, because your distance measuring device is imperfect), so the entries no longer correspond to the squared Euclidean distances between the $\boldsymbol{a}_i$. Explain how to determine points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^n$ (yes, n and not d) from $\boldsymbol{D}$ such that:

- $\sum_{i=1}^n \boldsymbol{x}_i = \boldsymbol{0}$;
- $\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2 \geq D_{i,j}$ for all $i \neq j$; and
- $\max\left\{\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2}{D_{i,j}} : 1 \leq i < j \leq n\right\}$ is as small as possible.

Hint: use semidefinite programming.

Solution.

a

$\boldsymbol{D}_{i,j} = ||\boldsymbol{a}_i - \boldsymbol{a}_j||_2^2 = (\boldsymbol{a}_i - \boldsymbol{a}_j)^T(\boldsymbol{a}_i - \boldsymbol{a}_j) = \boldsymbol{a}_i^T\boldsymbol{a}_i - 2\boldsymbol{a}_i^T\boldsymbol{a}_j + \boldsymbol{a}_j^T\boldsymbol{a}_j$. We have $\boldsymbol{s} = ||\boldsymbol{a}_i||_2^2 = \boldsymbol{a}_i^T\boldsymbol{a}_i$ and so

$$\boldsymbol{s}\boldsymbol{1}_n^\top = \begin{bmatrix} \boldsymbol{a}_1^T\boldsymbol{a}_1 & \boldsymbol{a}_1^T\boldsymbol{a}_1 & \boldsymbol{a}_1^T\boldsymbol{a}_1 & \ldots & \boldsymbol{a}_1^T\boldsymbol{a}_1 \\ \boldsymbol{a}_2^T\boldsymbol{a}_2 & \boldsymbol{a}_2^T\boldsymbol{a}_2 & \boldsymbol{a}_2^T\boldsymbol{a}_2 & \ldots & \boldsymbol{a}_2^T\boldsymbol{a}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_n^T\boldsymbol{a}_n & \boldsymbol{a}_n^T\boldsymbol{a}_n & \boldsymbol{a}_n^T\boldsymbol{a}_n & \ldots & \boldsymbol{a}_n^T\boldsymbol{a}_n \end{bmatrix}$$

and

$$\boldsymbol{1}_n\boldsymbol{s}^\top \begin{bmatrix} \boldsymbol{a}_1^T\boldsymbol{a}_1 & \boldsymbol{a}_2^T\boldsymbol{a}_2 & \boldsymbol{a}_3^T\boldsymbol{a}_3 & \ldots & \boldsymbol{a}_n^T\boldsymbol{a}_n \\ \boldsymbol{a}_1^T\boldsymbol{a}_1 & \boldsymbol{a}_2^T\boldsymbol{a}_2 & \boldsymbol{a}_3^T\boldsymbol{a}_3 & \ldots & \boldsymbol{a}_n^T\boldsymbol{a}_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_1^T\boldsymbol{a}_1 & \boldsymbol{a}_2^T\boldsymbol{a}_2 & \boldsymbol{a}_3^T\boldsymbol{a}_3 & \ldots & \boldsymbol{a}_n^T\boldsymbol{a}_n \end{bmatrix}$$

while

$$
\boldsymbol{A}\boldsymbol{A}^T = \begin{bmatrix} \boldsymbol{a}_1^T\boldsymbol{a}_1 & \boldsymbol{a}_1^T\boldsymbol{a}_2 & \boldsymbol{a}_1^T\boldsymbol{a}_2 & \dots & \boldsymbol{a}_1^T\boldsymbol{a}_n \\ \boldsymbol{a}_2^T\boldsymbol{a}_1 & \boldsymbol{a}_2^T\boldsymbol{a}_2 & \boldsymbol{a}_2^T\boldsymbol{a}_3 & \dots & \boldsymbol{a}_2^T\boldsymbol{a}_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_n^T\boldsymbol{a}_1 & \boldsymbol{a}_n^T\boldsymbol{a}_2 & \boldsymbol{a}_n^T\boldsymbol{a}_3 & \dots & \boldsymbol{a}_n^T\boldsymbol{a}_n \end{bmatrix}
$$

The i, j^{th} component of $\boldsymbol{s}\mathbf{1}_n^\top - 2\boldsymbol{A}\boldsymbol{A}^T + \boldsymbol{s}\mathbf{1}_n^\top = \boldsymbol{a}_i^T\boldsymbol{a}_i - 2\boldsymbol{a}_i^T\boldsymbol{a}_j + \boldsymbol{a}_j^T\boldsymbol{a}_j = \boldsymbol{D}_{i,j}$, hence $\boldsymbol{D} = \boldsymbol{s}\mathbf{1}_n^\top - 2\boldsymbol{A}\boldsymbol{A}^\top + \mathbf{1}_n\boldsymbol{s}^\top$.

b

$$
\begin{aligned}
-(1/2)\mathbf{\Pi}\boldsymbol{D}\mathbf{\Pi} &= -(1/2)\mathbf{\Pi}(\boldsymbol{s}\mathbf{1}_n^\top - 2\boldsymbol{A}\boldsymbol{A}^\top + \mathbf{1}_n\boldsymbol{s}^\top)\mathbf{\Pi} \\
&= -(1/2)(\mathbf{\Pi}\boldsymbol{s}\mathbf{1}_n^\top\mathbf{\Pi} - 2\mathbf{\Pi}\boldsymbol{A}\boldsymbol{A}^\top\mathbf{\Pi} + \mathbf{\Pi}\mathbf{1}_n\boldsymbol{s}^\top\mathbf{\Pi}) \text{ where } \mathbf{1}_n^\top\mathbf{\Pi} = \mathbf{\Pi}1_n = 0 \\
&= -(1/2)(-2\mathbf{\Pi}\boldsymbol{A}\boldsymbol{A}^\top\mathbf{\Pi}) \\
&= \mathbf{\Pi}\boldsymbol{A}\boldsymbol{A}^\top\mathbf{\Pi}
\end{aligned}
$$

c

We know that any vector $\boldsymbol{x}$ can be split into components consisting of $\mathbf{1}_n$ and its orthogonal projection in the form of $\boldsymbol{x} = \langle \boldsymbol{x}, \mathbf{1}_n \rangle \mathbf{1}_n + \mathbf{\Pi}\boldsymbol{x}$. Next, we rewrite the above in terms of $\mathbf{\Pi}\boldsymbol{x}$ which is $\mathbf{\Pi}\boldsymbol{x} = \boldsymbol{x} - \langle \boldsymbol{x}, \mathbf{1}_n \rangle \mathbf{1}_n$. To find out what $\mathbf{\Pi}$ is, we express the RHS in terms of $\boldsymbol{x}$. $\langle \boldsymbol{x}, \mathbf{1}_n \rangle = \boldsymbol{x}^T\mathbf{1}_n = \sum_{i=1}^n \boldsymbol{x}_i$, so $\langle \boldsymbol{x}, \mathbf{1}_n \rangle \mathbf{1}_n$ is an n dimension vector of $\sum_{i=1}^n \boldsymbol{x}_i$. This is also equivalent to the product of a $n \times n$ matrix of 1 ($= \mathbf{1}_n\mathbf{1}_n^T$) and $\boldsymbol{x}$. Hence $\mathbf{\Pi}\boldsymbol{x} = (I - \mathbf{1}_n\mathbf{1}_n^T)\boldsymbol{x}$ and $\mathbf{\Pi} = (I - \mathbf{1}_n\mathbf{1}_n^T)$ which is invertible. Using the results in b), $-(1/2)\mathbf{\Pi}\boldsymbol{D}\mathbf{\Pi} = \mathbf{\Pi}\boldsymbol{X}\boldsymbol{X}^\top\mathbf{\Pi}$ where $\boldsymbol{X}$ is the concatenation of $\boldsymbol{x}_i$s. With SVD, we can break $\mathbf{\Pi}\boldsymbol{X}\boldsymbol{X}^\top\mathbf{\Pi}$ into $\boldsymbol{V}\mathbf{\Sigma}\boldsymbol{V}^T = \boldsymbol{V}\mathbf{\Sigma}^{1/2}\mathbf{\Sigma}^{1/2}\boldsymbol{V}^T$ where $\boldsymbol{V}\mathbf{\Sigma}^{1/2} = \mathbf{\Pi}\boldsymbol{X}$ and $\mathbf{\Sigma}^{1/2}\boldsymbol{V}^T = \boldsymbol{X}^\top\mathbf{\Pi}$. By taking the product of $\mathbf{\Pi}^{-1}\boldsymbol{V}\mathbf{\Sigma}^{1/2} = \mathbf{\Pi}^{-1}\mathbf{\Pi}\boldsymbol{X}$, we recover $\boldsymbol{x}_1, ..\boldsymbol{x}_n$ in the form of $\boldsymbol{X}$.

$\square$

Problem 4 (25 points)**.** Exercise 3.25 from BHK.

Solution.

a

$$\arg\max ||Ax||_2^2 = \arg\max x^T A^T A x$$

The above is equivalent to solving for the eigenvalues and eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$ and choosing the eigenvector $\boldsymbol{v}_1$ with the largest eigenvalue λ_1. The synthetic document is then $\boldsymbol{A}\boldsymbol{v}_1$.

b

The synthetic document does not represent the centre of gravity which is the document with averaged term counts. Instead, it represents the document with the largest average dot-product with the set of documents represented by the matrix.

c

Perform eigendecomposition on $\boldsymbol{A}$ like in a) but choose the k eigenvectors with the largest eigenvalues. Obtain the synthetic documents by calculating the product $\boldsymbol{A}$ and $\boldsymbol{v}_i$ where $i \in [1, k]$ and are hence the document-term matrix multiplied by singular vectors.

d

Assuming that we can arrange $\boldsymbol{A}$ as a block-diagonal matrix, $\boldsymbol{A}^T\boldsymbol{A}$ (which could also be loosely called the 'term-correlation matrix') would be block-diagonal as well. To see this, the i, j^{th} entry of the matrix is $\langle \boldsymbol{a}_i^T, \boldsymbol{a}_j \rangle$ and would only be non-zero when the vectors belong to the same block since each block is characterised by an exclusive set of terms. Breaking the so-called 'term-correlation matrix' into parts via eigendecomposition, $\boldsymbol{A}^T\boldsymbol{A}$ can be represented as $\boldsymbol{V}\boldsymbol{\lambda}\boldsymbol{V}^T$ where $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$. As each entry of $\boldsymbol{A}^T\boldsymbol{A}$ can be expressed as $\lambda_i v_{ij}^2$, the eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$ or the right-singular vectors of $\boldsymbol{A}$ must also reflect the block-diagonal structure of $\boldsymbol{A}^T\boldsymbol{A}$ as we only see a 0 when $v_{ij} = 0$. Hence each right-singular vector can be divided into distinct blocks where the existence of non-zero numbers in the blocks gives rise to the corresponding block in $\boldsymbol{A}^T\boldsymbol{A}$ if it multiplied by another right-singular vector containing non-zero numbers in the same block.

e

Pick a $\boldsymbol{v}_I$ of the right singular matrix of $\boldsymbol{A}^T\boldsymbol{A}$ and obtain the $m \times 1$ vector obtained by $\boldsymbol{A}\boldsymbol{v}_i$. The corresponding documents with non-zero entries in the resulting vector all belong to the same cluster. Repeat till all documents have an assignment. □