Andy Liu
Math189R SU19
Homework 5
Thursday, June 25, 2020

Feel free to work with other students, but make sure you write up the homework and code on your own (no copying homework *or* code; no pair programming). Feel free to ask students or instructors for help debugging code or whatever else, though.

1 (Murphy 12.5 - Deriving the Residual Error for PCA) It may be helpful to reference section 12.2.2 of Murphy.

(a) Prove that

$$\left\| \mathbf{x}_i - \sum_{j=1}^{k} z_{ij}\mathbf{v}_j \right\|^2 = \mathbf{x}_i^\top \mathbf{x}_i - \sum_{j=1}^{k} \mathbf{v}_j^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{v}_j.$$

Hint: first consider the case when $k = 2$. Use the fact that $\mathbf{v}_i^\top \mathbf{v}_j$ is 1 if $i = j$ and 0 otherwise. Recall that $z_{ij} = \mathbf{x}_i^\top \mathbf{v}_j$.

(b) Now show that

$$J_k = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^\top \mathbf{x}_i - \sum_{j=1}^{k} \mathbf{v}_j^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{v}_j \right) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^\top \mathbf{x}_i - \sum_{j=1}^{k} \lambda_j.$$

Hint: recall that $\mathbf{v}_j^\top \mathbf{\Sigma} \mathbf{v}_j = \lambda_j \mathbf{v}_j^\top \mathbf{v}_j = \lambda_j$.

(c) If $k = d$ there is no truncation, so $J_d = 0$. Use this to show that the error from only using $k < d$ terms is given by

$$J_k = \sum_{j=k+1}^{d} \lambda_j.$$

Hint: partition the sum $\sum_{j=1}^{d} \lambda_j$ into $\sum_{j=1}^{k} \lambda_j$ and $\sum_{j=k+1}^{d} \lambda_j$.

(a) Note that we can expand $\left\| \mathbf{x}_i - \sum_{j=1}^{k} z_{ij}\mathbf{v}_j \right\|^2$ to show that it is equal to $(\mathbf{x}_i - \sum_{j=1}^{k} z_{ij}\mathbf{v}_j)^T \cdot (\mathbf{x}_i - \sum_{j=1}^{k} z_{ij}\mathbf{v}_j) = \mathbf{x}_i^T \mathbf{x}_i - \mathbf{x}_i^T(\sum_{j=1}^{k} z_{ij}\mathbf{v}_j) - (\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)^T \mathbf{x}_i + (\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)^T(\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)$.

Bringing the $\mathbf{x}_i$ terms into the summations, this expression becomes $\mathbf{x}_i^T \mathbf{x}_i - 2\sum_{j=1}^{k} z_{ij}\mathbf{v}_j^T \mathbf{x}_i + (\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)^T(\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)$. We can then make the substitution $z_{ij} = \mathbf{x}_i^T \mathbf{v}_j$, resulting in the expression $\mathbf{x}_i^T\mathbf{x}_i - 2\sum_{j=1}^{k}\mathbf{v}_j^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}_j + (\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)^T(\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)$.

We then consider the term $(\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)^T(\sum_{j=1}^{k} z_{ij}\mathbf{v}_j)$. Note that, combining the summations and using the fact that $\mathbf{v}_i^T\mathbf{v}_j$ is 0 if $i \neq j$, as well as our $z_{ij} = \mathbf{x}_i^T\mathbf{v}_j$ substitution, this is just equal to $\sum_{j=1}^{k}\mathbf{v}_j^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}_j$. Thus, our expression as a whole is equal to $\mathbf{x}_i^T\mathbf{x}_i - 2\sum_{j=1}^{k}\mathbf{v}_j^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}_j + \sum_{j=1}^{k}\mathbf{v}_j^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}_j$. Combining like terms, this expression becomes $\mathbf{x}_i^T\mathbf{x}_i - \sum_{j=1}^{k}\mathbf{v}_j^T\mathbf{x}_i\mathbf{x}_j\mathbf{v}_j$, as desired.

(b) It suffices to show that $\frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{k}\mathbf{v}_j^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{v}_j) = \sum_{j=1}^{k}\lambda_j$, as both sides contain an identical $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i$ term.

Note that we can rewrite this double summation as $\frac{1}{n}\sum_{j=1}^{k}\mathbf{v}_j^T\mathbf{v}_j(\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T)$. Note that this inner summation is now just equal to $n\mathbf{\Sigma}$, so our double summation is equivalent to the summation $\sum_{j=1}^{k}\mathbf{v}_j^T\mathbf{\Sigma}\mathbf{v}_j$. We are given that this is equal to $\sum_{j=1}^{k}\lambda_j$, as $\mathbf{v}_j^T\mathbf{\Sigma}\mathbf{v}_j = \lambda_j$ by definition, so we can conclude that $\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^\top\mathbf{x}_i - \sum_{j=1}^{k}\mathbf{v}_j^\top\mathbf{x}_i\mathbf{x}_i^\top\mathbf{v}_j\right) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i - \sum_{j=1}^{k}\lambda_j$, as desired.

(c) Note that $\sum_{j=1}^{d}\lambda_j = \sum_{j=1}^{k}\lambda_j + \sum_{j=k+1}^{d}\lambda_j$. Since we are given that $J_d = 0$, from part (b), we can conclude that $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i - \sum_{j=1}^{d}\lambda_j = 0 \Rightarrow \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^\top\mathbf{x}_i = \sum_{j=1}^{d}\lambda_j$. Thus, $J_k = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i - \sum_{j=1}^{d}\lambda_j + \sum_{j=k+1}^{d}\lambda_j = \sum_{j=k+1}^{d}\lambda_j$, as desired.

$\blacksquare$

2 (ℓ_1-Regularization) Consider the ℓ_1 norm of a vector $\mathbf{x} \in \mathbb{R}^n$:

$$\|\mathbf{x}\|_1 = \sum_i |\mathbf{x}_i|.$$

Draw the norm-ball $B_k = \{\mathbf{x} : \|\mathbf{x}\|_1 \leq k\}$ for $k = 1$. On the same graph, draw the Euclidean norm-ball $A_k = \{\mathbf{x} : \|\mathbf{x}\|_2 \leq k\}$ for $k = 1$ behind the first plot. (Do not need to write any code, draw the graph by hand).

Show that the optimization problem

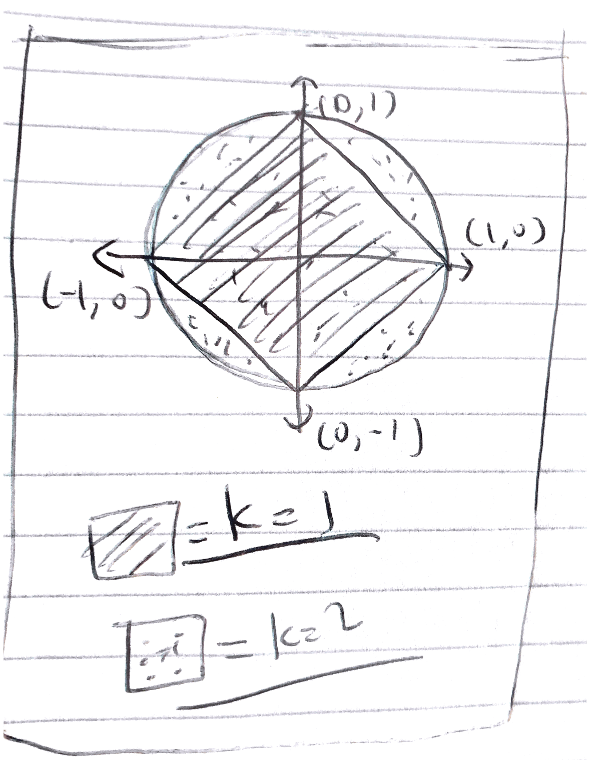
$$\begin{aligned} \text{minimize: } & f(\mathbf{x}) \\ \text{subj. to: } & \|\mathbf{x}\|_p \leq k \end{aligned}$$

is equivalent to

$$\text{minimize: } f(\mathbf{x}) + \lambda \|\mathbf{x}\|_p$$

(hint: create the Lagrangian). With this knowledge, and the plots given above, argue why using ℓ_1 regularization (adding a $\lambda\|\mathbf{x}\|_1$ term to the objective) will give sparser solutions than using ℓ_2 regularization for suitably large λ.

Below are the norm-balls B_k (the rhombus) and A_k (the circle) for $k = 1$.



To show that the given optimization problem is equivalent to minimizing $f(\mathbf{x}) + \lambda||\mathbf{x}||_p$, we begin by creating the Lagrangian, which is equal to $f(\mathbf{x}) + \lambda(||\mathbf{x}||_p - k)$. Since we seek to minimize this, note that the value of $\mathbf{x}$ that minimizes the Lagrangian is equal to the value of $\mathbf{x}$ that minimizes $f(\mathbf{x}) + \lambda(||\mathbf{x}||_p)$, since we can ignore the constant $-\lambda k$. Thus, the two optimization problems are equivalent.

Note that if we are trying to minimize $f(x)$ with the constraint $||\mathbf{x}||_p \leq k$, solutions when $p = 1$ are more likely to land on an "edge" where more variables are equal to zero since a B_k norm-ball has sharp edges, while the A_k Euclidean norm-ball does not. Thus, solutions for $p = 1$ are likely to be more sparse than solutions for $p = 2$.

$\blacksquare$