Math189R SU18
Homework 5
Monday, May 21, 2018

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)

$$\|x_i - \Sigma_{j=1}^{k} z_{ij} v_j\|^2 = (x_i - \Sigma_{j=1}^{k} z_{ij} v_j)^\top (x_i - \Sigma_{j=1}^{k} z_{ij} v_j)$$

$$= x_i^\top x_i - \Sigma_{j=1}^{k} z_{ij} x_i^\top - x_i^\top (\Sigma_{j=1}^{k} z_{ij} v_j) + (\Sigma_{j=1}^{k} z_{ij} v_j)^\top (\Sigma_{j=1}^{k} z_{ij} v_j)$$

$$= x_i^\top x_i - 2\Sigma_{j=1}^{k} z_{ij} v_j^\top x_i + (\Sigma_{j=1}^{k} z_{ij} v_j)^\top (\Sigma_{j=1}^{k} z_{ij} v_j)$$

$$= x_i^\top x_i - 2\Sigma_{j=1}^{k} z_{ij} v_j^\top x_i + \Sigma_{j=1}^{k} v_j^\top z_{ij}^\top z_{ij} v_j$$

$$= x_i^\top x_j - 2\Sigma_{j=1}^{k} v_j^\top x_i x_i^\top v_j + \Sigma v_j^\top x_i v_j^\top v_j x_i^\top v_j$$

Since $v_i^\top v_j = 1$ when $i = j$ and 0 other wise, then:

$$= x_i^\top x_j - 2\Sigma_{j=1}^k v_j^\top x_i x_i^\top v_j + \Sigma v_j^\top x_i x_i^\top v_j$$

$$= x_i^\top x_i - \Sigma v_j^\top x_i x_i^\top v_j$$

b)

$$J_k = \frac{1}{n}\Sigma_{i=1}^n (x_i^\top x_i - \Sigma_{j=1}^k v_j^\top x_i x_i^\top v_j)$$

$$= \frac{1}{n}\Sigma_{i=1}^n x_i^\top x_i - \frac{1}{n}\Sigma_{i=1}^n \Sigma_{j=1}^k v_j^\top x_i x_i^\top v_j$$

$$= \frac{1}{n}\Sigma_{i=1}^n x_i^\top x_i - \Sigma_{j=1}^k v_j^\top \frac{1}{n}\Sigma_{i=1}^n (x_i x_i^\top) v_j$$

$$= \frac{1}{n}\Sigma_{i=1}^n x_i^\top x_i - \Sigma_{j=1}^k v_j^\top \Sigma v_j$$

$$= \frac{1}{n}\Sigma_{i=1}^n x_i^\top x_i - \Sigma_{j=1}^k \lambda_j$$

c)

$$J_k = \frac{1}{n}\Sigma_{i=1}^n x_i^\top x_i - (\Sigma_{j=1}^d \lambda_j - \Sigma_{j=k+1}^d \lambda_j$$

Since $J_d = 0$, thus:

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

■

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

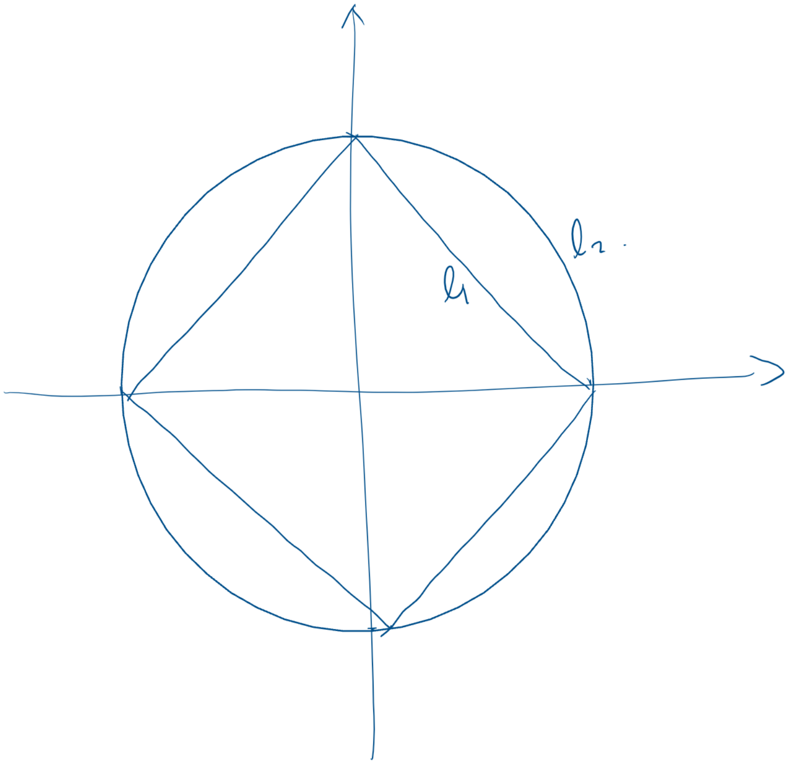
$$\begin{aligned} \text{minimize: } & f(\mathbf{x}) \\ \text{subj. to: } & \|\mathbf{x}\|_p \leq k \end{aligned}$$

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.

The norm balls graph is attached below:



To find the minimal value for f(x), we create the Lagrangian:

$$L(x,\lambda) = f(x) + \lambda(||x||_p - k) = f(x) + \lambda||x||_p - \lambda k$$

However, since λk does not depend on x, thus, the optimization function $L(x,\lambda)$ can be simplified as:

$$L(x,\lambda) = f(x) + \lambda(||x||_p - k) = f(x) + \lambda||x||_p$$

Since the norm ball l1 is a diamond shape and l2 is a circle, when we are regularizing, the probability of landing on the corner of the l1 norm ball, where at least one of the variables is zero, is much larger than the corner of l2. As a result, l1 penalty will encourage to drop more variables. Thus, l1 regularization will give sparser solutions than using l2 regularization for suitably large λ.

■

Extra Credit (Lasso) Show that placing an equal zero-mean Laplace prior on each element of the weights $\boldsymbol{\theta}$ of a model is equivelent to ℓ_1 regularization in the Maximum-a-Posteriori estimate

$$\text{maximize: } \mathbb{P}(\boldsymbol{\theta}|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}(\mathcal{D})}.$$

Note the form of the Laplace distribution is

$$\text{Lap}(x|\mu, b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$$

where μ is the location parameter and $b > 0$ controls the variance. Draw (by hand) and compare the density $\text{Lap}(x|0,1)$ and the standard normal $\mathcal{N}(x|0,1)$ and suggest why this would lead to sparser solutions than a Gaussian prior on each elements of the weights (which correspond to ℓ_2 regularization).

$$\text{We want to maximize:} \mathbb{P}(\boldsymbol{\theta}|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}(\mathcal{D})}$$

However, as we can see, $\mathbb{P}(\mathcal{D})$ does not depend on θ, then we can ignore this term. Then, we take the log of this likelihood function:

$$log(\mathbb{P}(\boldsymbol{\theta}|\mathcal{D})) = log(\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta}))$$

$$= log(\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})) + log(\mathbb{P}(\boldsymbol{\theta})))$$

Thus, it is the same as minimizing the negative log likelihood function:

$$min : -log(\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})) - log(\mathbb{P}(\boldsymbol{\theta})))$$

Since θ is distributed in the form of Laplace distribution, then we can rewrite $log(\mathbb{P}(\boldsymbol{\theta}))$ into:

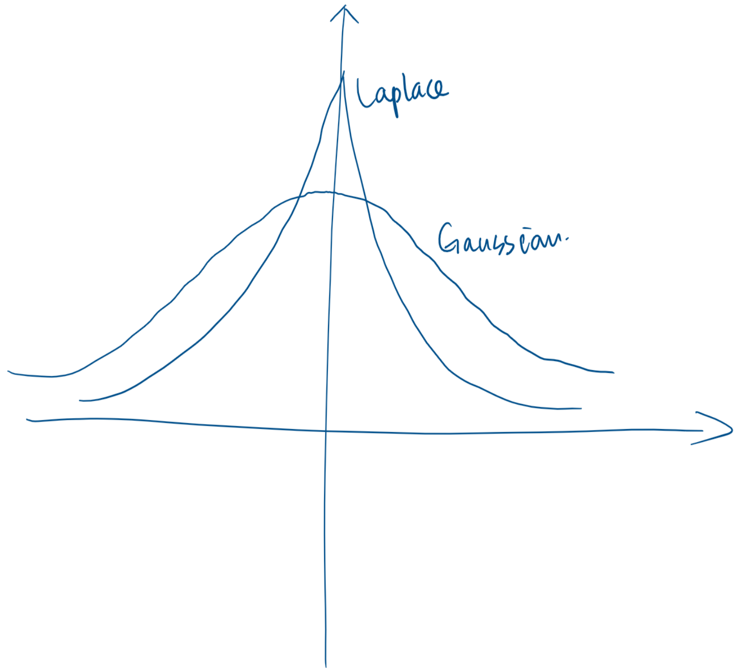
$$log(\mathbb{P}(\boldsymbol{\theta})) = -log(\frac{1}{2b}exp(-\frac{|\boldsymbol{\theta}|}{b}))$$

$$= -log(\frac{1}{2b}) + \frac{1}{b}|\boldsymbol{\theta}|$$

Since the first term is a constant, it won't have an influence on the minimizing problem, we can rewrite it in the form of:

$$\text{min:} - log(\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})) + \frac{1}{b}|\boldsymbol{\theta}|$$

This is the same as l1 regularized maximum likelihood estimate, where $\lambda = \frac{1}{b}$



The laplace distribution is sharper than the Gaussian when x = 0. As a result of this, it is similar to l1 penalty that would encourage zero weights, leading to sparser solutions than a Gaussian prior. ■