Eli Weissler

Math189R SU17
Homework 5
Monday, June 5, 2017

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 $\mathbf{v}_j^\top \Sigma \mathbf{v}_j = \lambda_j \mathbf{v}_j^\top \mathbf{v}_j = \lambda_j$.

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

For this problem, we have

- $v_j \in \mathbb{R}^D$ denotes the jth principal direction
- $x_i \in \mathbb{R}^D$ denotes the ith high-d representation
- $z_i \in \mathbb{R}^K$ denotes the ith low-d reprentation

To begin (a), just expand out the expression

■

a) $\| x_i - \sum_{j=1}^{k} z_{ij} v_j \|^2$ (Recall $z_{ij} = x_i^T v_j = v_j^T x_i$)

$$= x_i^T x_i - \underbrace{x_i^T \sum_{j=1}^{k} z_{ij} v_j - \sum_{j=1}^{k} v_j^T z_{ij}^T x_i}_{\text{These two evaluate to scalers and are the transpose of each other, so are equal}} + \underbrace{\left[\sum_{j'=1}^{k} z_{ij'} v_{j'}\right]^T \sum_{j=1}^{k} z_{ij} v_j}_{[z_{ij'} v_{j'}]^T = v_{j'}^T z_{ij'}^T}$$

$$= x_i^T x_i - 2 \sum_{j=1}^{k} \underbrace{z_{ij}}_{\text{This is a number}} \underbrace{v_j^T x_i}_{\text{so this must be also. Take transpose}} + \sum_{j'=1}^{k} \sum_{j=1}^{k} \underbrace{v_{j'}^T z_{ij'}^T z_{ij} v_j}_{= z_{ij'}^T v_{j'}^T v_j z_{ij} = z_{ij'}^T \delta_{jj'} z_{ij} = z_{ij} \delta_{jj'} z_{ij'}}$$

$$= x_i^T x_i - 2 \sum_{j=1}^{k} v_j^T x_i x_i^T v_j + \sum_{j=1}^{k} v_j^T x_i x_i^T v_j$$

$$= \boxed{x_i^T x_i - \sum_{j=1}^{k} v_j^T x_i x_i^T v_j}$$

This term is really $\sum z_{ij}^2$

b) Recall $\Sigma = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T$ is the empiracal covariance matrix

$$\rightarrow J_k = \frac{1}{n} \sum_{i=1}^{n} \left(x_i^T x_i - \sum_{j=1}^{k} v_j^T x_i x_i^T v_j \right) = \frac{1}{n} \sum_{i=1}^{n} x_i^T x_i - \sum_{j=1}^{k} v_j^T \Sigma v_j$$

v_j are eigenvectors of Σ w/ $\lambda = \lambda_j$

$\rightarrow v_j^T \Sigma v_j = \lambda_j v_j^T v_j = \lambda_j$

(assuming $\|v_j\| = 1$)

$$= \boxed{\frac{1}{n} \sum_{i=1}^{n} x^T x_i - \sum_{j=1}^{k} \lambda_j}$$

c) $J_{k=d} = \frac{1}{n} \sum_{i=1}^{n} x_i^T x_i - \sum_{j=1}^{k'} \lambda_j - \sum_{j=k'+1}^{d} \lambda_j = 0$ because we include all dimensions

So $J_{k=k'} = J_{k=k'} - J_{k=d} = \boxed{\sum_{j=k'+1}^{d} \lambda_j}$

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).

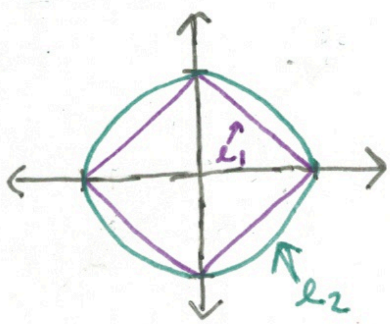
Show that the optimization problem

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

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.



It is clearest to see in two dimensions.

$\|\vec{x}\|_1 = x + y = 1$

$\|\vec{x}\|_2 = \sqrt{x^2 + y^2} = 1$

This extends into n-dimensions with n-dimensional hypercubes and hyperspheres.

Consider representing the minimization problem using Lagrange multipliers:

$$L(\vec{x}, \lambda) = f(\vec{x}) + \lambda\big(\overbrace{\|\vec{x}\|_p - 1}^{\text{This is zero}}\big)$$

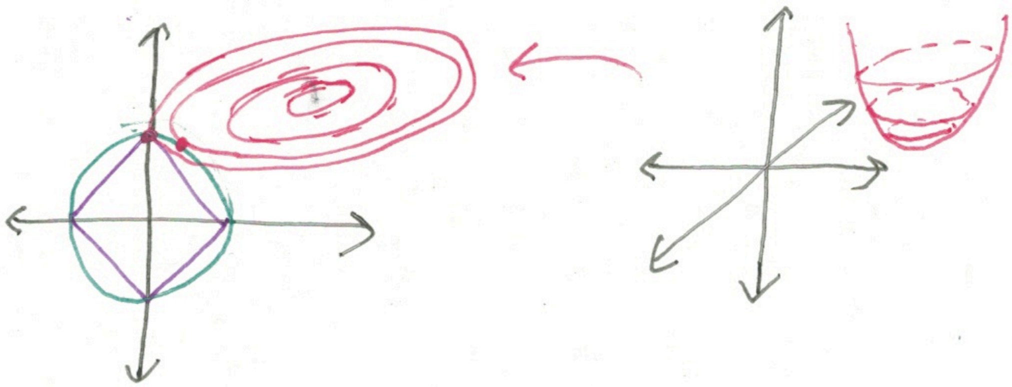
Taking partials of L yields

$$\frac{\partial L}{\partial x_i} = \frac{\partial f(\vec{x})}{\partial x_i} + \frac{\partial}{\partial x_i}\left[\lambda \|\vec{x}\|_p\right]$$

Therefore minimizing the constrained cost function is equivalent to minimizing

$$f(\vec{x}) + \lambda \|\vec{x}\|_p \quad \blacksquare$$

Now looking at the plot, ℓ_1 will give sparser solutions because



We are looking for the minimum value on our convex cost function that interacts with our p-norm ball. The ℓ_1 has sharper corners, so we are more likely to hit at the corner vs. at exactly the top of a circle. These corner hits correspond to zero values for one of our fit parameters, i.e. sparser solutions.