Optimization and Computational Linear Algebra – Brett Bernstein

Recitation 9

It may be helpful if after recitation you try to re-solve these problems by yourself, and use them as additional study problems for the class.

1. Define $x_1 = (4, 1)$, $x_2 = (-3, 1)$, and $x_3 = (1, 1)$.
   (a) Give a one-dimensional affine subspace of $\mathbb{R}^2$ that best approximates these three points.
   (b) Use this to represent each point using a single number (i.e., reduce the dimension from 2 to 1).

   Solution.

   (a) $\{(a, 1) : a \in \mathbb{R}\} = e_2 + \text{Span}(e_1)$
   (b) Using the representation above we can write

   $$x_1 = e_2 + 4e_1, \quad x_2 = e_2 - 3e_1, \quad x_3 = e_2 + e_1$$

   giving $\hat{x}_1 = 4$, $\hat{x}_2 = -3$, $\hat{x}_3 = 1$. Note that these do not add to 0. We could instead use the sample mean $\mu = (2/3, 1)^T$ and write the affine subspace as $\mu + \text{Span}(e_1)$ (note this is the same affine subspace). The resulting coefficients would then sum to zero. In PCA we always use the sample mean.

2. Suppose $x_1, \ldots, x_n \in \mathbb{R}^p$ are datapoints you want to represent in $k < p$ dimensions.
   (a) Explain how to do this using PCA.
   (b) How do you determine a value for k?
   (c) How can you implement PCA using the SVD?
   (d) Why should we perform dimensionality reduction?

   Solution.

   (a) Here are the steps:
      i. Let X be the matrix with x_i as its ith column, and then subtract the sample mean μ from each column giving $\tilde{X}$.
      ii. Form the sample covariance matrix $\Sigma = \frac{1}{n-1}\tilde{X}\tilde{X}^T$.
      iii. Choose the first k eigenvectors $v_1, \ldots, v_k$ from the spectral decomposition of Σ corresponding to the largest k eigenvalues. These are called the first k principal directions or loading vectors.

iv. Compute the coefficients of the data when you project onto the first k eigenvectors:

$$\hat{X} = Q^T \tilde{X}$$

where $Q \in \mathbb{R}^{p \times k}$ is the matrix whose ith column is v_i. Then $\hat{X} \in \mathbb{R}^{k \times n}$ contains the lower dimensional representation.

(b) Use a scree plot (look for elbow).

(c) Write $\tilde{X} = V \Sigma U^T$. Then

$$(n-1)\Sigma = \tilde{X}\tilde{X}^T = V \Sigma \Sigma^T V^T.$$

Thus we can use the first k left singular vectors to perform the projection.

(d) Here are some reasons:

i. Allows us to visualize high dimensional data.

ii. Can reduce the size of the data that we input into other learning algorithms downstream in our ML pipeline.

iii. Can act as a form of regularization (e.g., discrete version of ridge regression called principal component regression).

3. Suppose there are two eigenvectors of the covariance matrix that correspond to large eigenvalues, and the rest of the eigenvalues are small. How do we interpret this?

Solution.

(a) There is a good two-dimensional affine subspace approximating the data. We can also say that two directions carry most of the variance of the data. Note that these directions are linear combinations of features (i.e., baskets of features), so it may be hard to interpret what they mean.

4. Let $x_1, \ldots, x_n \in \mathbb{R}^p$, and fix a direction $w \in \mathbb{R}^p$ with $\|w\| = 1$. We define the variance along the direction w by

$$\frac{1}{n-1} \sum_{i=1}^{n} (w^T x_i - w^T \mu)^2$$

where $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$. On the homework we will see that the first eigenvector of the covariance matrix gives the direction with maximum variance. Why is this desirable?

Solution. If we project onto a direction with low variance, then we are destroying information present in the data. The hope is that the variability present in the data is informative, and will be useful for our ML techniques (prediction, clustering, etc.).

5. Someone suggests that you should standardize each feature before running PCA (i.e., subtract the mean of each feature, and then divide by the standard deviation). Does this have any effect?

Solution. If we compute the spectral decomposition XX^T instead of $\tilde{X}\tilde{X}^T$ (i.e., if we don't center the data first) then the principal components will be influenced by the location of the mean (e.g., the first principal direction may point toward the mean of the data). Stated differently, we will try to approximate the data with a low dimensional linear subspace through the origin instead of an affine subspace, leading to a biased less accurate approximation.

If we do not standardize the data then the units used to measure each feature can play a large role in the principal directions. For example, changing from meters to centimeters will multiply the feature by 100, which will scale the variance in that direction by 10000. PCA always favors directions with larger variance, so this will skew the results. This is why we often use the correlation matrix instead of the covariance matrix for PCA. You may not want to standardize the data if you think the differences in scale are informative (maybe the data is already measured in the same units).

6. Suppose $A \in \mathbb{R}^{n\times n}$ (not necessarily symmetric) has a linearly independent list of n eigenvectors $v_1, \ldots, v_n$ with eigenvalues $\lambda_1, \ldots, \lambda_n$. Can we factor A in a way similar to the spectral decomposition?

   Solution. Yes, we can write A as

   $$A = V\Lambda V^{-1}.$$

   Note that V need not be orthogonal here.

7. The Fibonacci sequence is defined by $F_0 = 0$, $F_1 = 1$, and $F_{k+2} = F_{k+1} + F_k$ for $k \geq 0$. How quickly does F_k grow (linearly, polynomially, exponentially)?

   Solution. Note that

   $$\begin{bmatrix} F_{k+2} \\ F_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix}$$

   so that

   $$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}.$$

   This matrix has 2 linearly independent eigenvectors

   $$v_1 = \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix} \quad v_2 = \begin{bmatrix} \lambda_2 \\ 1 \end{bmatrix}$$

   with eigenvalues

   $$\lambda_1 = \frac{1+\sqrt{5}}{2}, \lambda_2 = \frac{1-\sqrt{5}}{2}.$$

   While these vectors are orthogonal, they are not normalized to keep the expressions simple. If you know about characteristic polynomials or some other technique you can

compute these by hand. Otherwise, you can use numpy.linalg.eig. Note that

$$\begin{aligned}\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= (V\Lambda V^{-1})^n \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= (V\Lambda^n V^{-1}) \tfrac{v_1 - v_2}{\sqrt{5}} \\ &= \lambda_1^n \tfrac{v_1}{\sqrt{5}} - \lambda_2^n \tfrac{v_2}{\sqrt{5}}.\end{aligned}$$

Thus F_n is given by

$$F_n = \frac{\lambda_1^n - \lambda_2^n}{\sqrt{5}}.$$

Since $|\lambda_1| > |\lambda_2|$ we see the λ_1^n term dominates giving

$$F_n \sim \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n.$$