CSCC11 Week 9 Notes

Unsupervised Learning:

- With supervised learning algos you want to produce the desired outputs for the given inputs. Also, you're given both the inputs and outputs during training.

- With unsupervised learning algos, only the inputs are given during training. The labels/outputs are unknown.

- Types of unsupervised learning:
  1. Dimension Reduction
  2. Clustering
  3. Data Density Modelling

Dimension Reduction:

- Increasing the num of features will not always improve performance. It may even lead to worse performance.

- Generally, the num of training data required increases exponentially with dimensionality to avoid overfitting.

- The goal is to choose an optimum set of features to lower dimensionality.

- Feature extraction: Finds a set of new features through some mapping f() from existing features

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \xrightarrow{f(x)} Y = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix}$$

Mapping could be linear or non-linear.

$K << N$

- Feature Selection: Chooses a subset of the original features.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \longrightarrow y = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ik} \end{bmatrix}$$

$k << N$

Intro to Principal Component Analysis (PCA):

- Is a technique for dimensionality reduction. It aims to find a low-dimensional representation of high dimensional data.

- Uses and motivations of PCA:

1. Visualization: High-dim data are extremely hard to visualize (I.e. To see how disjoint 2 diff categories of feature vectors might be or to see how noisy some measurements are.) PCA provides a way to project high-dim data onto 2^d or 3^d for purposes of easy visualization.

2. Pre-processing: Learning regression and classification models from high-dim data is often very slow and prone to overfitting. This is called the curse of dimensionality.

3. Compression: One of PCA's earliest uses was data compression. If one can find a low-dim representation of a high-dim image, for example, then one can use such a representation to store and transmit data more efficiently.

PCA Intuition and Steps:

- Assume each data point y_i has dimension p, so $y_i \in R^p$.
  The aim is to reduce the dimension.
  There are 2 ways to do so.

- **Maximum Variance Formulation:**
  - Find the orthogonal projection of the data into a lower dim linear space s.t. the variance of the projected data is maximised.

  - Consider the projection onto a 1-D space.
  - The linear projection is $u_1^T y_i$, $u_1 \in R^p$
  - For convenience, we choose the unit vector. I.e. $u_1^T u_1 = 1$
  - Assume y_i is standardized.
    ↳ $\frac{y-\bar{y}}{SD(y)}$ ← The mean of y.
  - Then, the sample variance of the projected data is

$$\frac{\sum_{i=1}^{N} (u_1^T y_i)^2}{N} = u_1^T S u_1$$

where $S = \frac{1}{N} \sum_{i=1}^{N} y_i \cdot y_i^T$

  - We want to max $u_1^T S u_1$ w.r.t u_1, given the constraint $u_1^T u_1 = 1$.

- We have $\arg\max_{u_1} \left(u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1) \right)$

  The soln is: $S u_1 = \lambda_1 u_1$

  $\rightarrow u_1^T S u_1 = \lambda_1$ (Multiply both sides by u_1^T)

- The variance is max when u_1 = the eigenvector having the largest eigenvalue λ_1.
  Such $u_1^T y_i$ is called the first principal component.

- We can extend this to k-Dim.
  Let $U = [u_1, u_2, \dots, u_k] \rightarrow U \in \mathbb{R}^{p \times k}$
  The k^{th} principal component for the i^{th} sample is $u_k^T y_i$. We can find/get this by ranking the eigenvalues.

  The new (dimension reduced) data is:

$$x_i = \begin{bmatrix} u_1^T y_i \\ u_2^T y_i \\ \vdots \\ u_k^T y_i \end{bmatrix} = U^T y_i \in \mathbb{R}^k$$

  Principal components are uncorrelated:

$$\begin{aligned} E(U^T y y^T U) &= U^T E(y y^T) U \\ &= U^T (U D U^T) U \\ &= (U^T U) D (U^T U) \\ &= D \leftarrow \text{A diagonal matrix} \end{aligned}$$

- Minimum Error Formulation:
  - Find a linear projection that min the error btwn the data points and their projection
    $y = Wx + b$ where W is a $p \times k$ matrix and x is a k-dim vector.

    $W = [w_1, \ldots, w_k]$

  - One way to learn the model is to solve the following problem:

    $$\arg\min_{W, b, \{x_i\}} \sum_i \| y_i - (Wx_i + b) \|^2$$

    Subject to $W^T W = \underbrace{I_{k \times k}}_{\text{Identity matrix of size } k.}$

    This constraint requires that we obtain an orthonormal mapping W.
    I.e.
    $$w_i^T w_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

    Without this, the problem would be underconstrained.
    E.g.
    Assume $\alpha \neq 0$
    $$y_i = W^* x_i$$
    $$= (\alpha W^*)\left(\tfrac{1}{\alpha} x_i\right)$$
    I can change W^* and x_i but still get the same error.

Steps to solve the problem:

1. Let $B = \frac{\sum_i y_i}{N}$

2. Compute the data co-variance matrix
   $C = \frac{\sum_i (y_i - b)(y_i - b)^T}{N}$

3. Let $VDV^T = C$ be the eigenvector decomposition of C. D is a diagonal matrix of eigenvalues (I.e. $D = \text{diag}(\lambda_1, \dots, \lambda_d)$) and $V = [v_1, \dots, v_d]$ contains the orthonormal eigenvectors.
   $V^T V = I_d$

4. Assume the eigenvalues are sorted from largest to smallest. If this is not the case, sort them along with their corresponding eigenvectors.

5. Let W be a matrix containing the first k eigenvectors.
   I.e. $W = [v_1, \dots, v_k]$

6. Let $x_i = W^T (y_i - b) \; \forall i$

Representation and Reconstruction of New Data:

- Suppose we have learned a PCA model and are given a new y_{new} value. To estimate its corresponding x_{new} value, do:
  1. Minimize $\|y_{new} - (Wx_{new} + b)\|^2$.
     However, since W is orthonormal, the solution simplifies to
  2. $x^*_{new} = W^T(y_{new} - b)$

Properties of PCA:

1. **Mean Zero Coefficients:** The PCA coefficients/latent coordinates, $\{x_i\}_{i=1}^n$, have a mean of 0.

   Proof:

$$\text{Mean}(x) = \frac{\sum_{i=1}^{N} x_i}{N}$$

$$= \frac{\sum_{i=1}^{N} W^T(y_i - b)}{N} \leftarrow \text{By step 6 on pg 6}$$

$$= \frac{W^T}{N}\left(\sum_{i=1}^{N}(y_i - b)\right)$$

$$= \frac{W^T}{N}\left(\underbrace{\sum_{i=1}^{N} y_i - Nb}_{\text{Equals } 0}\right)$$

$$= 0 \leftarrow \text{By step 1 on pg 6.}$$

We set $b = \frac{\sum_i y_i}{N}$

$$Nb = \sum_i y_i$$

2. Max Var Formulation and Min Error Formulation are equivalent.

3. Out of subspace error: The total variance in the data is given by the sum of the eigenvalues of the sample co-variance matrix. The variance captured by PCA is the sum of the first k eigenvalues. The total amount of variance lost is given by the sum of the remaining eigenvalues.

   One can show that the least-squares error in the approx to the original data provided by the opt model params ω^*, $\{x_i^*\}$, and b^* is:

$$\sum_i \|y_i - (\omega^* x_i^* + b^*)\|^2$$

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

   When learning a PCA model, it is common to use the ratio of the total LS error and total variance in the training data (I.e. The sum of all eigenvalues). One needs to choose a k large enough s.t. this ratio is small (often 0.1 or less).

4. Proportion of Variance:

$$\text{Proportion of variance} = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j}$$

$$\sum_{j=1}^{p} \lambda_j = \sum_{j=1}^{p} var(x_j) = p$$

PCA Final Comments:

1. Generally, it's critical to perform standardization prior to PCA.

   PCA is very sensitive to the variances of the inital variables.
   I.e. If there are large diff btwn the range of the initial variables, the variables with the larger ranges will dominate over those with small ranges.

2. Pros:
   - Removes correlated features. Used a lot in the multi-collinearity issue.
   - Can speed up algos with fewer features.
   - Reduces overfitting with fewer features.

3. Cons:
   - Less interpretable since it transforms the original data.
   - Data standardization is a must.
   - Info loss w/o proper number of components