Machine Learning (CS60050) – Weekly Report

Kaushal Banthia (19CS10039)

Week 9: 6th – 8th October, 2021

Topics Covered:

- Why to reduce dimensions?
- Subset Selection
- Principal component analysis (PCA)
- Linear Discriminant Analysis (LDA)

Summary (Topic Wise):

- Why to reduce dimensions?

  - For reducing complexity of inference, memory and computation.
  - In most learning algorithms, the complexity depends on the number of input dimensions (d) and the size of the data sample (N),
  - Saves cost of extraction of features. Also, simpler models are more robust in small datasets. They are convenient to plot and visualize.
  - Explanation is convenient with fewer features for knowledge extraction.
  - There are two major approaches towards reducing dimensions.
    - Feature selection: To find k of the d dimensions that give us the most information discarding the other (d – k) dimensions. This can be done by subset selection method.
    - Feature extraction: A new set of k dimensions that are a combination of the original d dimensions. This can be done via supervised and unsupervised techniques like PCA and LDA.

- Subset Selection

  - F is a feature set of input dimensions $x_i \; for \; i = 1,2,..d$
  - E(F) is the error in the validation set if F is used as input.
  - This is a supervised method. It has two greedy methods:
    - Sequential forward selection

```
1. F = NULL
2. Select $x_i$ which provides least $E(F \cup x_i)$
3. Add $x_i$ if $E(F \cup x_i) < E(F)$
4. Repeat above 2 steps, till no more addition is possible.
```

    - Sequential backward selection

```
1. F = Set of all the features
2. Select $x_i$ which provides least $E(F - x_i)$
3. Remove $x_i$ if $E(F - x_i) < E(F)$
4. Repeat above 2 steps, till no more removal is possible.
```

Both of them are local search methods and don't guarantee optimal feature combination. Also, the cost of training and testing is $O(d^2)$. The more the features, the costlier the training.

- <u>Principal component analysis (PCA)</u>

  - To find a mapping from the inputs in the original d – dimensional space to a new $(k < d)$ – dimensional space, with minimum loss of information.
  - x is the input feature vector of dimension d
  - w is a direction (unit vector) of dimension d.
  - Projection of x along w: $w^T x$ (Make data centered around origin of the space).
  - Principal component is the component along the direction w_1 such that its variance is maximum among all possible projections. This is also known as the 1st principal component. The 2nd component is the component along a direction w_2 orthogonal to w_1 having the maximum variance. Similarly other principal components are defined (For a d – dimensional space there are maximum d principal components).
  - Computation of 1st component is done as follows:

    - $z_1 = w_1^T x$ (The corresponding random variable is denoted as Z_1)
    - X is the random variable with instance x, mean m & covariance matrix Σ.
    - $Mean(z_1)$: $w_1^T m$, $Variance(z_1)$: $w_1^T \Sigma w_1$
    - The optimization problem is maximizing the variance, while keeping w_1 as a unit vector.
    - $w_1 = argmax\{w^T \Sigma \mathrm{w} - \mathrm{l} * (w^T \mathrm{w} - 1)\}$, where l is the Lagrange coefficient.
    - Taking the derivative of the argument with respect to w and setting it to 0.
    - $2\Sigma w_1 - 2lw_1 = 0 \rightarrow \Sigma w_1 = lw_1 \rightarrow w_1^T \Sigma w_1 = w_1^T l w_1 = l w_1^T w_1 = l\ (variance)$
    - w_1 is the eigen vector of Σ corresponding to the maximum eigen value.

  - Computation of 2nd component:

    - $z_2 = w_2^T x$
    - Optimization problem: $w_2 = argmax_w\{w^T \Sigma w - l_1(w^T w - 1) - l_2(w_1^T w - 0)\}$
    - l_1 and l_2 are Lagrange coefficients and w_2 is orthogonal to w_1.
    - Taking the derivative of the argument with respect to w and setting it to 0.
    - $2\Sigma w_2 - 2l_1 w_2 - l_2 w_1 = 0$
    - Pre-multiplying with w_1 we get
    - $2w_1^T \Sigma\, w_2 - 2l_1 w_1^T w_2 - l_2 w_1^T w_1 = 0 \rightarrow w_1^T \Sigma\, w_2 - l_2 = 0$
    - As $w_1^T \Sigma w_2$ is scalar, $w_2^T \Sigma w_1$ is also scalar.
    - Replacing $\Sigma w_1\ by\ l w_1\ (w_2^T \Sigma w_1 = 0. Hence\ l_2 = 0)$
    - $2\Sigma w_2 - 2l_1 w_2 = 0 \rightarrow \Sigma w_2 = l_1 w_2 \rightarrow w_2$ is eigen vector and l_1 is variance.
    - w_2 is the eigen vector of Σ to the 2nd maximum eigen value.
    - Other components can be calculated similarly.

  - PCA-Algorithm:

    - Input: A set of data points: $S = \{x_j = (x_{1j}, x_{2j}, \dots x_{dj}) \big| x_j\ in\ R^d\}$
    - Output: A set of k eigen vectors providing tx. matrix: $W = [w_1, w_2, \dots, w_k]$

```
1. Compute mean of data points & translate data points to their mean.
2. Compute covariance matrix of the set.
3. Compute eigen vectors and eigen values (in increasing order).
4. Choose k such that the fraction of variance accounted for is more
   than a threshold.
5. Use those k-components for representing any data point.
```

  - Properties:

    - PCA diagonalizes the data covariance matrix Σ.
    - $\Sigma = CDC^T$ where D: Diagonal matrix and C: Columns are unit eigen vectors of $\Sigma \rightarrow CC^T = C^TC = I$
    - Components are uncorrelated as covariance among components is zero.
    - By normalizing components with their variances (eigen values), Euclidean distance could be used for classification.
    - Reconstruction error from lower dimensional space minimum among all linear transforms of the data.

  - Application of PCA:

    - Data Compression
    - Decorrelating Components
    - Factor Analysis
    - Classification / High Level Processing

- Linear Discriminant Analysis (LDA)

  - For the purpose of classification, dimensional reduction using PCA may not work as it captures the direction of maximum variance for a data set. For labelled data sets, it does not capture the direction of maximum separation between the groups of data points of differing labels.
  - Fisher linear discriminant

    - Consider a set of data points $S = \{x_i | x_i \ in \ R^d\}$ with N_1 points in class w_1 and N_2 points in class w_2. Say, $N_1 + N_2 = N$ (total data points).
    - Consider a line with direction u.
    - Projection of data x_i on u: $y_i = x_i^T u$
    - One dimensional subspace representing data.

  - Separation between projected data of different classes

    - m_1= mean of data points in w_1 and m_2= mean of data points in w_2
    - Projection of means: $m_{y1} = m_1^T u \ and \ m_{y2} = m_2^T u$
    - A measure of separation: $D = |m_{y1} - m_{y2}|$. It doesn't consider variance.
    - Thus, a better measure of separation would be by normalizing by a factor which is proportional to the class variances.
    - Scatter of data belonging to class C: $s^2 = \sum_{y \in C}(y - m_c)^2$
    - Measure of separation: $J(u) = \frac{D^2}{s_1^2 + s_2^2}$, where s_1 is the scatter of class w_1 and s_2 is the scatter of class w_2.
    - We need to obtain u by maximizing J(u).
    - The scatter matrix for samples of class C in original space:
      $$S_C = \sum_{x \in C}(x - m_c)(x - m_c)^T$$
    - Within the class scatter matrix: $S_W = S_1 + S_2$ & $S_1^2 + S_2^2 = u^T S_W u$
    - Between the class scatter matrix: $S_B = (m_1 - m_2)(m_1 - m_2)^T$ & $D^2 = u^T S_B u$

- Rewriting optimization function, to maximize J(u).

$$J(U) = \frac{D^2}{s_1^2 + s_2^2} = \frac{u^T S_B u}{u^T S_W u}$$

- To maximize J(u), u should be such that $S_W^{-1} S_B u = \lambda u$

$$u = S_W^{-1}(m_1 - m_2) \qquad (Only\ direction\ matters)$$

Concepts Challenging to Comprehend: None yet.

Interesting and Exciting Concepts: Principal Component Analysis (PCA)

Concepts not understood: None yet.

A novel idea: We should do an analysis on the dataset before applying PCA, to understand, whether PCA is required or not. We could check the correlation between the columns and if they are strongly correlated, then PCA would work well. On the other hand, if the columns are weakly correlated, PCA does not work well to reduce data. This is because, each column carries valuable data that cannot be represented from other columns (since they are not related).

Also, we can use PCA for K – Means Clustering, after prediction, for visualization purposes. We can bring the dataset down to the 2 most important variables (first 2 components) and then visualize the predictions made (on the complete dataset, without the application of PCA) on the reduced data, using a scatter plot.