Weekly report of lessons

Name: Mayank Kumar
Roll No: 19CS30029
The week: 04-10-2021 to 10-10-2021

The topics covered:

Dimensionality Reduction: Reasons, Approaches: Subset Selection, PCA: Computation of 1st component, Computation of 2nd component, PCA Algorithm, PCA Properties, Applications of PCA; Linear Discriminant Analysis: Fisher linear discriminant, Separation between projected data of different classes, A better measure of separation; Scatter Matrix: Within the class Scatter matrix, Between the class Scatter matrix, Maximizing J(u)

Summary topic wise:

Dimensionality Reduction:

Reasons: To reduce the complexity of inference, memory and computation; save the cost of extraction of features; build a simpler and more robust model; easily plot and visualize the features

Approaches:

- Feature selection: Subset selection method which finds k dimensions that give most information discarding other (d - k) dimensions
- Feature extraction: Use supervised and unsupervised techniques to get a new set of k dimensions that are a combination of original d dimensions. Ex- PCA, LDA

Subset Selection: For a feature set F of input dimensions x_i (i = 1 to d), error in validation set E(F), the supervised method requires training and testing. Greedy methods:

- Sequential Forward Selection: The features are sequentially added to empty set of features until the addition of extra features is possible. Cost = $O(d^2)$
- Sequential Backward Selection: The features are picked from input data in a set and are sequentially removed until the removal is possible. Cost = $O(d^2)$

Principal Component Analysis (PCA): Accounts for minimum loss of information while mapping the inputs from d-dimensional space to k dimensions (k < d). For input vector x and unit vector w of dimension d, projection of x along w = w^Tx. The component along direction w_1 such that its variance is maximum among all possible projections is called 1st or principal component. The component along direction w_2 orthogonal to w_1 having maximum variance is called 2nd component.

Computation of 1st component: $z_1 = w_1^T x$, for an instance x of random variable X, if the corresponding random variable is denoted as Z_1, we have: Mean of $Z_1 = w_1^T m$ and Variance of $Z_1 = w_1^T \Sigma w_1$

To maximize variance keeping w_1 as a unit vector: $w_1 = argmax_w\{w^T \Sigma w - l\cdot(w^Tw - 1)\}$, l is Lagrange coefficient. Taking derivative w.r.t. w and equating to 0, we get:

$\Sigma w = l\, w_1 \Rightarrow w_1^T \Sigma w_1 = l\, w_1^T w_1 = l$(variance), where w_1 is the eigenvector of Σ for maximum eigenvalue.

Computation of 2nd component: $z_2 = w_2^T x$

$w_2 = argmax_w\{w^T \Sigma w - l_1\cdot(w^Tw - 1) - l_2\cdot(w_1^Tw - 0)\}$, l_1 and l_2 are Lagrange coefficients. Taking derivative w.r.t. w and equating to 0, we get: $l_2 = 0$ and $\Sigma w_2 = l_1 w_2$, where w_2 is the eigenvector of Σ for 2nd maximum eigenvalue and l_1 is the variance

PCA Algorithm: For a set of data points S having instances of dimension d, it outputs the mapping as a set of k eigenvectors. The algorithm involves computing the mean of data points, translating them to their mean, computing the covariance matrix of the set, eigenvectors and eigenvalues, and finally choosing k such that the fraction of variance accounted for is more than a threshold.

PCA Properties:

- It diagonalizes data covariance matrix Σ (=CDC^T), D -> diagonal matrix, C -> unit eigenvectors
- The components are uncorrelated
- Euclidean distance can be used for classifying if components are normalized with their variances
- Minimum reconstruction error from lower dimensional space

Applications of PCA:

- Data Compression: an optimum set of orthonormal basis is provided for a set of data points
- Decorrelating Components: PCA on highly correlated RGB space color images gives transformed matrix, useful for segmentation. Also used in multispectral, hyperspectral and ultraspectral remote
- Factor Analysis: Decorrelated factors are highlighted which is useful for classification. Ex: eigen faces for representing human faces

- Classification / High-Level Processing: Representation derived by PCA is used

Linear Discriminant Analysis: The PCA algorithm for dimensionality reduction might not work for classification as it uses the maximum variance direction for a data set. No direction of maximum separation is captured between groups of data points having different labels.

Fisher linear discriminant: Consider S = $\{x_i \mid x_i \text{ in } R^d\}$, total data points N = N1+N2 (points in class w1 and w2 respectively), and a line with direction u. The projection of one dimensional subspace x_i representing data on u is $y_i = x_i^T u$.

Separation between projected data of different classes: Consider m_1 and m_2 as the mean of data points in w_1 and w_2 respectively. Projection of means: $m_{y1} = m_1^T u$ and $m_{y2} = m_2^T u$. The measure of separation is given as D = $|m_{y1} - m_{y2}|$ which does not consider the variance of data.

A better measure of separation: Normalized by a factor proportional to class variances.

Scatter of data belonging to class C: $s^2 = \sum_{y \in C} (y - m_c)^2$

Measure of separation: J(u) = $\frac{D^2}{(s_1^2 + s_2^2)}$

Scatter Matrix: Scatter matrix for samples of class C in original space is $S_C = \sum_{x \in C} (x - m_c)(x - m_c)^T$

Within the class Scatter matrix: $S_w = S_1 + S_2$. On solving, we get $s_1^2 + s_2^2 = u^T S_w u$

Between the class Scatter matrix: $S_B = (m_1 - m_2)(m_1 - m_2)^T$ and $D^2 = u^T S_B u$

Maximizing 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), such u should be taken that makes $S_w^{-1} S_B u$ invertible. $\Rightarrow$ $u = S_w^{-1}(m_1 - m_2)$

Concepts challenging to comprehend:
None

Interesting and exciting concepts:
None

Concepts not understood:
None

Any novel idea of yours out of the lessons:
A point to be observed is that both Laplacian eigenmaps and Kernel machines use the same idea of feature embedding through preserving the given pairwise similarities or computing it by a kernel function. On the same line, we can think of using nonlinear basis functions for nonlinear dimensionality reduction. As of now, we use Euclidean distance or dot product for similarity calculation. Here, the kernel methods would definitely allow us to use other apt metrics for similarity calculation.
