12 a

Dimensionality Reduction

* Motivation

- Complexity of most estimators depends on the number of inputs.
- Affects time and space complexity.
- Reducing the dimensionality of the input lowers the complexity.

Goal: Prune part of the feature space that contribute little to the solution.

* Arguments for Dimensionality Reduction for Classifications

- Reduces complexity of the classifier.
- Irrelevant dimensions add to the variance.
- If data is explained with fewer features, we can get a better idea about the underlying process.
- What is relevant for computing a solution to our problem?

* Feature Selection vs Feature Extraction

Feature Selection → We try to find the K out of D dimensions that contain most of the information. We discard the other (D-K) dimensions.

Feature Extraction → We try to find a new set of K dimensions that are combinations of the D dimensions and yield most of the information.

* Popular Dimensionality Reduction Technique for Feature Extraction

- PCA : Principal Component Analysis
- Fisher-LDA : Fisher's Linear Discriminant Analysis.
- LLE: Locally Linear Embedding

★ Principal Component Analysis (PCA)

Idea of PCA

→ Find the mapping from the original D to K dimensional space with minimum loss of information ($K < D$).

→ PCA analysis the spread of the data and tries to find new dimensions that cover the spread best.

Data Matrix

→ Matrix of N vectors with D dimensions

$$\underset{N \times D}{X} = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix}$$

→ D-dimensionals vectors.

→ Goal: Find K-dimensions that represents the data as good as possible.

$$\underset{N \times D}{X} = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} \Rightarrow \underset{N \times K}{W} = \begin{bmatrix} w_1^T \\ \vdots \\ w_N^T \end{bmatrix}$$

Scanned by CamScanner

* Mean Reduced Features

$$\mu_x = \frac{1}{N} \sum_{n=1}^{N} x_n = \frac{1}{N} X^T 1_N$$

⇒ Mean reduced features

$$\bar{x}_n = x_n - \mu_x$$

→ In matrix form

$$\bar{X} = X - 1_N \mu_x^T$$

* Covariance Matrix of Mean-Reduced Features

$$\Sigma_{xx} = \frac{1}{N-1} \bar{X}^T \bar{X}$$

⇒ We describe our data using first and second central moment in the D-dimensional space. (i.e. μ, Σ_{xx})

↓

D-dimensional

Question: What are the best $K \ll D$ dimensions to approximate the data?

★ Eigenvector and Eigenvalue

→ The Eigenvector u_1 corresponding to the largest the largest Eigenvalue of Σ_{xx} is the direction of maximum spread.

→ Make this Eigenvector u_1 the first principal component.

→ All other Eigenvectors are orthogonal to u_1.

→ Repeat the process K times for the remaining Eigen values/vectors.

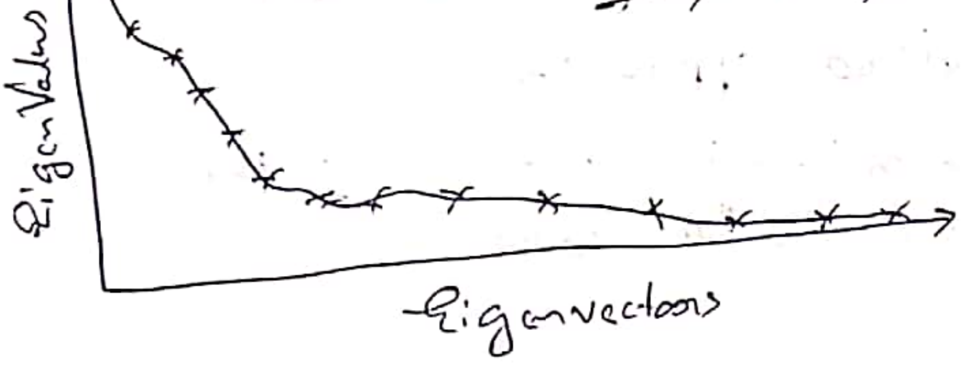
★ Eigenvalue Decomposition

⇒ Eigenvalue decomposition yields.

$$\Sigma_{xx} = \underbrace{R}_{\text{rotation matrix}}\ \underbrace{S^2}_{\text{diagonal matrix}} R^T$$

⇒ Eigenvalues/Vectors are computed via SVD - Single value decomposition (or special variants).

→ Useful for deciding K.



* Mapping to the Reduced Space

$$x \rightarrow [u_1^T(x-\mu), \cdots u_K^T(x-\mu)]$$

$$= [\omega_1, \cdots \omega_K]$$

K dimensional data points

D-dimensional data point

* Mapping to the Original Space

⇒ We can also map from the reduced K-dimensional space to the original one.

$$\hat{x} = \mu + \sum_{i=1}^{K} \omega_i u_i$$

⇒ which yields the squared reconstruction error:

$$e(x) = \|\hat{x} - x\|^2$$

* PCA Summary

→ Linear dimensionality reduction

→ Unsupervised approach

→ Goal is to minimize the sum of the squared reconstruction error.

Scanned by CamScanner

→ Principal components are the directions of maximum spread.

→ Computed via Eigenvalues/vectors of the Covariance matrix of the training data Points.

Fisher's Linear Discriminant Analysis

Limitation of PCA

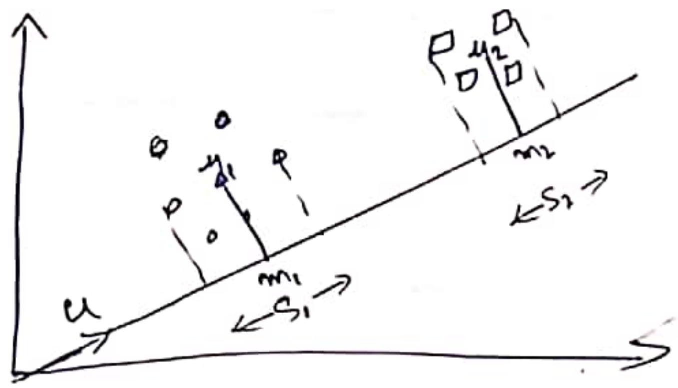
→ The direction of maximum variance is not always good for classification.

⇒ Fisher's LDA tries to find the best direction to separate classes.

* Idea of Fisher-LDA

⇒ Find the u that

- maximizes $\|m_1 - m_2\|$
- minimizes s_1 and s_2



* LDA for 2 Classes and K=1

⇒ Compute the mean of the classes

$$m_1 = \frac{\sum_t u^T x^t c^t}{\sum_t c^t} \qquad m_2 = \frac{\sum_t u^T x^t (1-c^t)}{\sum_t (1-c^t)}$$

$$c^t = 1 \rightarrow m_1 \qquad c^t = 0 \rightarrow m_2$$

⇒ and the Scatter of Samples in the 1-dim Space ($s_1^2 = \sigma_1^2 (\Sigma_t c^t - 1)$)

$$s_1^2 = \sum_t (u^T x^t - m_1)^2 c^t$$

$$s_2^2 = \sum_t (u^T x^t - m_2)^2 (1 - c^t)$$

* <u>Objective Function</u>

$$J(u) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

→ maximize for LDA

$$(m_1 - m_2)^2 = (u^T \mu_1 - u^T \mu_2)^2$$

$$= u^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T u$$

$$= u^T S_B u$$

→ between-class scatter matrix

$$s_1^2 = \sum_t (u^T x^t - m_1)^2 c^t$$

$$= u^T \sum (x^t - \mu_1)(x^t - \mu_1)^T c^t \, u$$

$$= u^T S_1 u$$

→ within-class scatter matrix of class 1.

⇒ Similarly for s_2^2.

$$s_1^2 + s_2^2 = u^T S_1 u + u^T S_2 u$$
$$= u^T (S_1 + S_2) u$$
$$= u^T (S_w) u$$

total within-class Scatter matrix

$$J(u) = \frac{u^T S_B u}{u^T S_w u}$$

$$u^* = \underset{u}{\arg\max}\, J(u) = S_w^{-1} (\mu_2 - \mu_1)$$

→ Fisher-LDA is the optimal solution if both features are normally distributed.

→ Also applicable for non-normally distributed features.

→ Can be easily generalized to N Classes and $k > 1$.

→ Linear dimensionality reduction.

★ Fischer's LDA vs PCA

- PCA minimizes the reconstruction error
- PCA is the standard choice for unsupervised Problem (no labels)

* Fisher-LDA exploits class labels to find a subspace so that separates the classes as good as possible.

★ Locally Linear Embedding (LLE)

* Technique for unsupervised non-linear dimensionality reduction
* Compute for each input data point a coordinate on a low-dimensional manifold.

★ LLE Key Steps.

1. Step 1: Find neighbours for each input point.
2. Step: Compute for each input point a weight vector that best recovers the point itself from its neighbours.
3. Step: For each point, find latent coordinates such that the same weights can be used for re-construction.