Lecture 17: Dimensionality Reduction

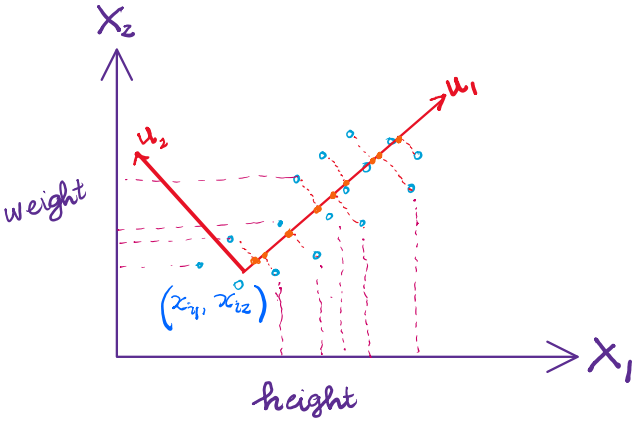
20 March 2024 11:03

Q: Can we reduce the data without compromising the information?

Two methods for dimension reduction:

- Unsupervised: $\{x_i\}$ → Principal Component Analysis
- Supervised: $\{(x_i, y_i)\}$ → LDA.

PCA:



Note: u_1 is a better projection direction than u_2 for this data.

PCA in action:

i) Data visualization

ii) Feature extraction.

Setup:

Consider a Dataset, $D = \{x_1, x_2, \dots x_n\}$, $x_i \in \mathbb{R}^d$

Goal: To map these data on a dimension $m < d$.

The Question becomes: Which u_1 to choose (see fig above)?

Objective: maximize the variance of projected data.

For a point x_i, $\|x_i\|$ is the length of $\vec{x_i} = (x_{i1}, x_{i2})$ from origin.

∴ Projection of x on u_1 direction:

$$\text{proj}_{u_1}(x_i) = (x_i^T u_1) u_1$$

Now to calculate the variance of projected data we first need mean:

$$\text{Mean of projected data} = \frac{1}{n} \sum_{i=1}^{n} \text{proj}_{u_1}(x_i) = (u_1^T \bar{x}) u_1$$

(due to linearity)

Mean of projected data $= \frac{1}{n}\sum_{i=1}^{n} \text{proj}_{u_1}(x_i) = (u_1^T x)\, u_1$

(due to linearity)

Now, Variance of the projections $= \frac{1}{n}\sum_{i=1}^{n} \left(u_1 x_i - u_1 \bar{x}\right)^2$

$$= \frac{1}{n}\sum_{i=1}^{n} \left(u_1^T (x_i - \bar{x})\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} u_1^T (x_i - \bar{x})(x_i - \bar{x})^T u_1$$

$$= u_1^T \left(\frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T\right) u_1$$

$$= u_1^T S u_1$$

(S: covariance matrix)

Aside: (Covariance Matrix)

Let $x_1 = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix}$, $x_2 = \begin{pmatrix} x_{21} \\ x_{22} \end{pmatrix}$ and $\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$

$$X = \begin{bmatrix} x_1 - \mu & x_2 - \mu & \cdots & x_n - \mu \end{bmatrix}$$

$\frac{1}{n} XX^T$ = covariance matrix of data points = S

PCA as optimization Problem:

maximize $u_1^T S u_1$ subject to constraint $u_1^T u_1 = 1$

Notes:

1. $\nabla^2 f(u_1)$ is Positive Semi-definite.
2. S is a real symmetric matrix
3. $y^T X X^T y = \|X^T y\|^2 \geq 0$

Though $f(u_1) = u_1^T S u_1$ is convex, we try to maximize it, instead of minimizing! So this not a convex optimization problem.

Result: For a real symmetric matrix A, A has d-orthonormal eigen vectors.

S is real symmetric. Let $v_1, v_2, \ldots v_d$ be its orthonormal eigenvectors with eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_d$. We know $v_i^T v_j = \mathbb{1}\{i=j\}$

We've: $Sv_i = \lambda_i v_i$ $\qquad V = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_d \\ | & & | \end{bmatrix}_{d \times d}$

$$SV = \begin{bmatrix} Sv_1 & Sv_2 \cdots & Sv_d \end{bmatrix}$$

$$= \begin{bmatrix} \lambda_1 v_1 & \lambda_2 v_2 & \cdots & \lambda_d v_d \end{bmatrix}$$

$$= V \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \cdots \\ \cdots & \cdots & \ddots & \cdots \\ \cdots & \cdots & 0 & \lambda_d \end{bmatrix}_{d \times d}$$

Let $\Sigma = \text{diag}(\lambda_1, \ldots \lambda_d)$ $\therefore S = V\Sigma$ — (1)

Observe that $V^T V = I$ as $\left[V^T V\right]_{ij} = v_i^T v_j$
(unitary matrix)

Now, $SVV^T = V\Sigma V^T$

But $VV^T = I \Rightarrow S = V\Sigma V^T$ → digonalization!
(eigen value decomposition)

Since $\{v_1, \ldots v_d\}$ form an orthonormal basis for $\mathbb{R}^d$, we can write

$$u_1 = \sum_{j=1}^{d} x_j v_j = Vx \qquad \text{where} \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

Returning to our objective fn:

$$f(u_1) = u_1^T S u_1 = x^T \underbrace{V^T V}_{I} \Sigma \underbrace{V^T V}_{I} x$$

$$= x^T \Sigma x = \sum_{j=1}^{d} \lambda_j x_j^2$$

Thus our new optimization problem:

$$\max \sum_{j=1}^{d} \lambda_i x_j^2 \quad \text{s.t.} \quad \sum_{j=1}^{d} x_j^2 = 1$$
(convert to linear programm)

This can be easily solved: Put $x_k = 1$ where $k = \text{argmax}\, \lambda_i$ and $x_j = 0 \;\forall j \neq k$

This can be easily solved: Put $x_k = 1$ where $k = \underset{i}{\text{argmax}}\ \lambda_i$ and $x_j = 0\ \forall j \neq k$

$\therefore$ First principal component, $u_1 = v_k$ (largest eigen vector) {usually we arrange in decreasing λ's then $v_k = v_1 = u_1$}

Generalization for $d \to m$ reduction:

After we've found first Principal component, u_1, we can find u_2

$$\max\ u_2^T S u_2$$
$$\text{s.t.}\quad u_2^T u_2 = 1, \quad u_1^T u_2 = 0$$

$\Rightarrow u_2$ = second largest eigenvectors.

For m Principal components, choose the largest m eigenvectors.

PCA algorithm:

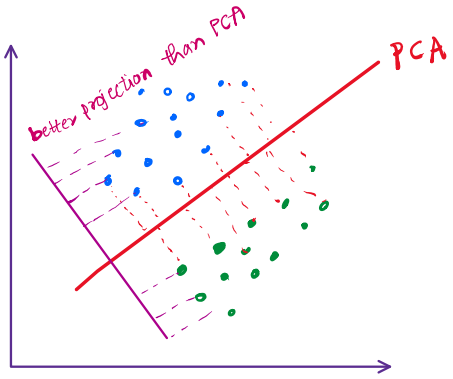
1. Compute mean of data points, $\bar{x}$
2. Mean centre the data, i.e. compute $x_i - \bar{x}$, $\forall i$
3. Compute the covariance matrix, $S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$
4. Do eigenvalue decomposition of $S = V \Sigma V^T$
5. Pick m top (largest) eigenvectors: $u_1, \ldots u_m \to$ Principal Components
6. Create a projection matrix, $U = [u_1, \ldots\ u_m]$
7. For each $x_i \in D$, $U^T x_i$ will give the projection on m-dimensional space

$$U^T x = \begin{bmatrix} u_1^T x \\ u_2^T x \\ \vdots \\ u_m^T x \end{bmatrix}$$

Supervised Learning:

If we use PCA, it fails to retain class information.

We need a projection that separates the classes well



We need a good measure to evaluate these projections (variance not enough)

$$J(u) = \left| \mu_1^T u - \mu_2^T x \right| \qquad \mu_i : \text{Mean of class } i.$$

One possible measure,
but this alone is not a good measure.

Objective: i) The different class means are well separated

ii) The data in the same class are not well separated.