Unsupervised learning (1)

Fraida Fund

Contents

Unsupervised learning

The basic supervised learning problem

Given a **sample** with a vector of **features**

$$\mathbf{x} = (x_1, x_2, ...)$$

There is some (unknown) relationship between $\mathbf{x}$ and a **target** variable, y, whose value is unknown.

We want to find $\hat{y}$, our **prediction** for the value of y.

The basic unsupervised learning problem

Given a **sample** with a vector of **features**

$$\mathbf{x} = (x_1, x_2, ...)$$

We want to learn something about the underlying *structure* of the data.

No labels!

Unsupervised learning examples

- dimensionality reduction
- clustering
- anomaly detection
- feature learning
- density estimation

Dimensionality reduction

Why?

- Supervised ML on small feature set
- Visualize data
- Compress data

Goal of dimensionality reduction

Previous feature selection:

- Choose subset of existing features
- Many features are somewhat correlated; redundant information

Now: minimum number of features, maximum information.

Dimensionality reduction problem

- Given $N \times p$ data matrix **X** where each row is a sample x_n
- **Problem**: Map data to $N \times p'$ where $p' \ll p$
- subject to ???

PCA intution (1)

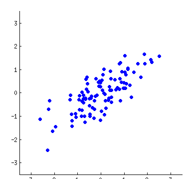

Figure 1: Data with two features, on two axes. Data is centered.

PCA intuition (2)

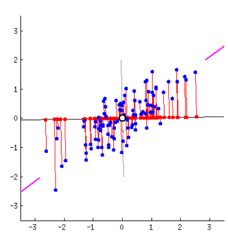

Figure 2: Construct a new feature by drawing a line $w_1 x_1 + w_2 x_2$, and projecting data onto that line (red dots are projections). Animation source

PCA intuition (3)

Which line?

- Maximize average squared distance from the center to each red dot; variation of new feature

- Minimize average squared length of the corresponding red connecting lines; total reconstruction error

Projections

Given vectors z and v, θ is the angle between them. projection of z onto v is:

$$\hat{z} = \text{Proj}_v(z) = \alpha v, \quad \alpha = \frac{v^T z}{v^T v} = \frac{||z||}{||v||} \cos\theta$$

$V = \{\alpha v | \alpha \in R\}$ are the vectors on the line spanned by v, then $\text{Proj}_v(z)$ is the closest point in V to z: $\hat{z} = \text{argmin}_{w \in V} ||z - w||^2$.

Sample covariance matrix (1)

- sample variance $s_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$
- sample covariance $s_{xy} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$
- covariance matrix is $p \times p$: $\text{Cov}(x, y)$ is a matrix Q with components:

$$Q_{k,l} = \frac{1}{N} \sum_{i=1}^N (x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l)$$

Sample covariance matrix (2)

Let $\widetilde{X}$ be the data matrix with sample mean removed, row $\tilde{x}_i = x_i - \bar{x}$
Sample covariance matrix is:

$$Q = \frac{1}{N} \widetilde{X}^T \widetilde{X}$$

(compute covariance matrix by matrix product!)

Directional variance

Projection onto v: $z_i = v^T \tilde{x}_i$

- Sample mean: $\bar{z} = v^T \bar{x}$
- Sample variance: $s_z^2 = v^T Q v$

Maximizing directional variance (1)

Given data $\tilde{x}_i$, what directions of unit vector v ($||v|| = 1$) maximizes the variance of projection $z_i = v^T \tilde{x}_i$ along direction of v?

$$\max_v v^T Q v$$

s.t $||v|| = 1$

Maximizing directional variance (2)

Let $v_1, \dots, v_p$ be *eigenvectors* of Q (there are p):

$$Qv_j = \lambda_j v_j$$

- Any local maxima of the directional variance is an eigenvector of Q.
- Sort them in descending order: $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_p$. The largest one is the maximizing vector.

Projections onto eigenvectors: uncorrelated features

- Eigenvectors are orthogonal: $v_j^T v_k = 0$ if $j \neq k$
- So the projections of the data onto eigenvectors are uncorrelated
- These are called the *principal components*
- In practice, computed using singular value decomposition (SVD): numerically more stable

PCA intuition (5)

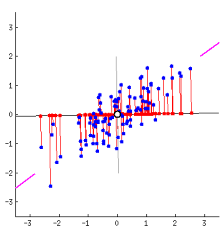

Figure 3: Grey and black lines form rotating coordinate frame. When variance of projection is maximized, the black line points in direction of first eigenvector of covariance matrix, and grey line points toward second eigenvector.

Approximating data

- Given data $\tilde{x}_i$, $i = 1, \dots, N$, and PCs $v_1, \dots, v_p$

$$\tilde{x}_i = \sum_{j=1}^{p} \alpha_{i,j} v_j, \quad \alpha_{i,j} = v_j^Y \tilde{x}_i$$

Consider approximation with d coefficients:

$$\hat{x}_i = \sum_{i=1}^{d} \alpha_{i,j} v_j$$

Average approximation error

For sample i, error is:

$\tilde{x}_i - \hat{x}_i = \sum_{j=d+1}^{p} \alpha_{i,j} v_j$$

which is sum of smallest $p - d$ eigenvalues:

$$\frac{1}{N}\sum_{i=1}^{N}||\tilde{x}_i - \hat{x}_i||^2 = \sum_{j=d+1}^{p}\lambda_j$$

Proportion of variance

- Variance of data set: $\frac{1}{N}\sum_{i=1}^{N}||\tilde{x}_i||^2 = \sum_{j=1}^{p}\lambda_j$
- Average approximation error: $\frac{1}{N}\sum_{i=1}^{N}||\tilde{x}_i - \hat{x}_i||^2 = \sum_{j=d+1}^{p}\lambda_j$
- The *proportion of variance* explained by d PCs is:

$$PoV(d) = \frac{\sum_{j=1}^{d}\lambda_j}{\sum_{j=1}^{p}\lambda_j}$$

PCA demo

Notebook link

PCA reference

Excellent set of notes on the topic: Link

Clustering

Clustering problem

- Given $N \times d$ data matrix **X** where each row is a sample x_n
- **Problem**: Group data into K clusters
- More formally: Assign $\sigma_n = \{1, \dots, K\}$ cluster label for each sample
- Samples in same cluster should be close: $||x_n - x_m||$ is small when $\sigma_n = \sigma_m$

K-means clustering

We want to minimize

$$J = \sum_{i=1}^{K} \sum_{n \in C_i} ||x_n - \mu_j||^2$$

- u_i is the mean of each cluster
- $\sigma_n \in \{1, \dots, K\}$ is the cluster that x_n belongs to

K-means algorithm

Start with random (?) guesses for each μ_i. Then, iteratively:

- Update cluster membership (nearest neighbor rule): For every n,

$$\sigma_n = \underset{i}{\text{argmin}} \, ||x_n - \mu_i||^2$$

- Update mean of each cluster (centroid rule): for every i, u_i is average of x_n in C_i

(Sensitive to initial conditions!)

K-means visualization

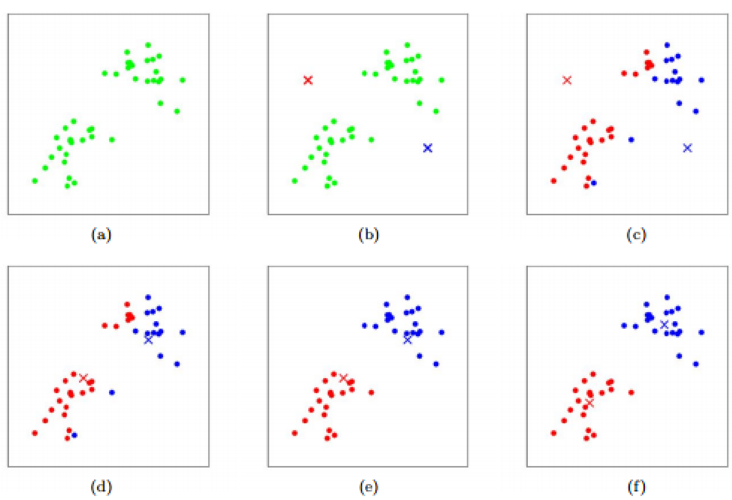

Figure 4: Visualization of k-means clustering.

K-means demo

Notebook link