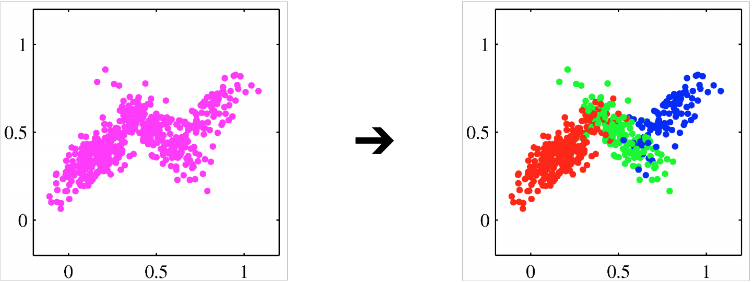
Clustering

ORIE 4741

November 13, 2020

Clustering: data segmentation

Discover groups such that samples within a group are more similar to each other than samples across groups.



Clustering: unsupervised methods

A clustering algorithm groups data points into clusters:

example:

- medical diagnosis. cluster patients with similar medical histories
- topic model. cluster documents with similar patterns of word usage
- market segmentation. cluster customers with similar purchase patterns

Table of Contents

Clustering ingredients

- A dissimilarity function between samples.
  - Quantitative? Ordinal? Categorical?

Clustering ingredients

- A dissimilarity function between samples.
  - Quantitative? Ordinal? Categorical?
- A loss function to evaluate clusters.
  - Usually weighted average of pairwise dissimilarity

Clustering ingredients

- A dissimilarity function between samples.
  - Quantitative? Ordinal? Categorical?
- A loss function to evaluate clusters.
  - Usually weighted average of pairwise dissimilarity
- Algorithm that optimizes this loss function.

K-means clustering: Objective Function

- All variables are of the quantitative type.
- Use squared Euclidean distance as dissimilarity metric.

$$d\left(x_i, x_{i'}\right) = \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 = \|x_i - x_{i'}\|^2$$

K-means clustering: Objective Function

- All variables are of the quantitative type.
- Use squared Euclidean distance as dissimilarity metric.

$$d\left(x_i, x_{i'}\right) = \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 = \left\|x_i - x_{i'}\right\|^2$$

We can then write the within-cluster dissimilarity:

$$\begin{aligned} W(C) &= \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} \left\|x_i - x_{i'}\right\|^2 \\ &= \sum_{k=1}^{K} N_k \sum_{C(i)=k} \left\|x_i - \bar{x}_k\right\|^2 \end{aligned}$$

where $\bar{x}_k = (\bar{x}_{1k}, \ldots, \bar{x}_{pk})$ is the mean vector associated with the kth cluster, and $N_k = \sum_{i=1}^{N} \mathrm{I}(C(i) = k)$ is the size of kth cluster.

K-means clustering: Algorithm

Optimization problem:

$$C^* = \min_C \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

K-means clustering: Algorithm

Optimization problem:

$$C^* = \min_C \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

For any set of observations S,

$$\bar{x}_S = \underset{m}{\operatorname{argmin}} \sum_{i \in S} \|x_i - m\|^2$$

K-means clustering: Algorithm

Optimization problem:

$$C^* = \min_{C} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

For any set of observations S,

$$\bar{x}_S = \operatorname*{argmin}_{m} \sum_{i \in S} \|x_i - m\|^2$$

Consider the enlarged optimization problem

$$\min_{C, \{m_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - m_k\|^2$$

K-means clustering: Alternative Minimization

Consider the enlarged optimization problem

$$\min_{C,\{m_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - m_k\|^2$$

- Fix clusters $C_1, \ldots, C_K$, the minimizers of $\{m_k\}_{k=1}^K$ are the mean of points in each cluster.
- Fix cluster centers $\{m_k\}_{k=1}^K$, for each point i the best cluster would be

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} \|x_i - m_k\|^2$$

- Iterate until convergence.
- Multiple random starting points should be used to find the global optimal solution.

Gaussian mixture

- Each cluster is described in terms of a Gaussian density, which has a centroid and a covariance matrix.

Consider "observations" $Y_1 \sim \mathcal{N}(\mu_1, \Sigma_1), \ldots, Y_k \sim \mathcal{N}(\mu_k, \Sigma_k)$, the true observation is X a weighted average:

$$X = \sum_{i=1}^{k} p_i Y_i, \quad \text{with } \sum_{i=1}^{k} p_i = 1 \text{ and } 0 < p_i < 1$$

Gaussian mixture

- Each cluster is described in terms of a Gaussian density, which has a centroid and a covariance matrix.

Consider "observations" $Y_1 \sim \mathcal{N}(\mu_1, \Sigma_1), \ldots, Y_k \sim \mathcal{N}(\mu_k, \Sigma_k)$, the true observation is X a weighted average:

$$X = \sum_{i=1}^{k} p_i Y_i, \quad \text{with } \sum_{i=1}^{k} p_i = 1 \text{ and } 0 < p_i < 1$$

- Generate X from cluster $\mathcal{N}(\mu_i, \Sigma_i)$ with probability p_i.

Gaussian mixture: Model parameters

Consider "observations" $Y_1 \sim \mathcal{N}(\mu_1, \Sigma_1), \ldots, Y_k \sim \mathcal{N}(\mu_k, \Sigma_k)$, the true observation is X a weighted average:

$$X = \sum_{j=1}^{k} p_j Y_j, \quad \text{with } \sum_{j=1}^{k} p_j = 1 \text{ and } 0 < p_j < 1$$

- Likelihood function:

$$\ell(\{\mu_j, \Sigma_j, p_j\}_{j=1}^{k} | x) = f_X(x | \{\mu_j, \Sigma_j, p_j\}_{j=1}^{k}) = \sum_{j=1}^{k} p_j \phi(x; \mu_j, \Sigma_j)$$

- Goal: find the mean and covariance $\{\mu_j, \Sigma_j\}_{j=1}^{k}$ and the weights $\{p_j\}_{j=1}^{k}$ that maximize the likelihood function.

Gaussian mixture: Model estimation

$$\ell(\{\mu_j, \Sigma_j, p_j\}_{j=1}^k | x) = f_X(x | \{\mu_j, \Sigma_j, p_j\}_{j=1}^k) = \sum_{j=1}^k p_j \phi(x; \mu_j, \Sigma_j)$$

- Fixing the cluster mean and covariance $\{\mu_j, \Sigma_j\}_{j=1}^k$ and weights $\{p_j\}_{j=1}^k$, find the contribution of clusters on each point:

$$\hat{p}_{ij} = \frac{p_j \phi(x_i; \mu_j, \Sigma_j)}{\sum_{j=1}^k p_j \phi(x_i; \mu_j, \Sigma_j)}, \qquad \text{for each point } x_i$$

- Update the mean and covariance matrix and weights, using the weighted average of samples:

$$\mu_j = \frac{\sum_{i=1}^N \hat{p}_{ij} x_i}{\sum_{i=1}^N \hat{p}_{ij}}, \quad \Sigma_j = \frac{\sum_{i=1}^N \hat{p}_{ij} (x_i - \mu_j)(x_i - \mu_j)^\top}{\sum_{i=1}^N \hat{p}_{ij}}, \quad p_j = \frac{\sum_{i=1}^N \hat{p}_{ij}}{N}$$

Gaussian mixture: EM algorithm

- Fixing the cluster mean and covariance $\{\mu_j, \Sigma_j\}_{j=1}^k$ and weights $\{p_j\}_{j=1}^k$, find the contribution of clusters on each point:

$$\hat{p}_{ij} = \frac{p_j \phi(x_i; \mu_j, \Sigma_j)}{\sum_{j=1}^k p_j \phi(x_i; \mu_j, \Sigma_j)}, \qquad \text{for each point } x_i$$

Expectation step: the assignment of cluster on each point is random. Compute its expectation given old model parameters.

Gaussian mixture: EM algorithm

- Update the mean and covariance matrix and weights, using the weighted average of samples:

$$\mu_j = \frac{\sum_{i=1}^N \hat{p}_{ij} x_i}{\sum_{i=1}^N \hat{p}_{ij}}, \quad \Sigma_j = \frac{\sum_{i=1}^N \hat{p}_{ij}(x_i - \mu_j)(x_i - \mu_j)^\top}{\sum_{i=1}^N \hat{p}_{ij}}, \quad p_j = \frac{\sum_{i=1}^N \hat{p}_{ij}}{N}$$

  Maximization step: maximize the likelihood function ruling out the randomness of cluster membership assignment with its expectation.

- EM algorithm is guaranteed to increase the objective function value at each iteration.
- Not guaranteed to find the global optimal solution. Multiple starting points needed.

Gaussian mixture vs K-means: cluster description

For each data point x_i and clusters $C_1, ..., C_k$

- K-means: center point for each cluster
- Gaussian mixture: mean and covariance matrix for each cluster

Gaussian mixture vs K-means: soft vs hard assignment

For each data point x_i and clusters $C_1, ..., C_k$

- K-means: assign 1 to one cluster and 0 to all other clusters.
- Gaussian mixture: assign probability $p1, ..., p_k$ to all clusters.

Gaussian mixture vs K-means: soft vs hard assignment

For each data point x_i and clusters $C_1, ..., C_k$

- K-means: assign 1 to one cluster and 0 to all other clusters.
- Gaussian mixture: assign probability $p1, ..., p_k$ to all clusters.
- When Gaussian mixture has covariance matrix all as $\sigma^2 \mathrm{I}$, the assigned probability is a monotone function of the distance between data point and the center.

Gaussian mixture vs K-means: soft vs hard assignment

For each data point x_i and clusters $C_1, ..., C_k$

- K-means: assign 1 to one cluster and 0 to all other clusters.
- Gaussian mixture: assign probability $p1, ..., p_k$ to all clusters.
- When Gaussian mixture has covariance matrix all as $\sigma^2\mathrm{I}$, the assigned probability is a monotone function of the distance between data point and the center.
- When σ^2 approaches 0, Gaussian mixture is identical to K-means.

Gaussian mixture vs K-means: optimization procedure

- K-means: iterate between finding cluster center points and assigning cluster membership to each point
- Gaussian mixture: iterate between assigning cluster weights and estimate model parameters

Gaussian mixture vs K-means: optimization procedure

- K-means: iterate between finding cluster center points and assigning cluster membership to each point
- Gaussian mixture: iterate between assigning cluster weights and estimate model parameters
- K-means: alternative maximization, each step explicitly solve an optimization problem
- Gaussian mixture: EM algorithm, E-step computes expectation, M-step maximizes a likelihood function different from the original objective function

K-means and Gaussian mixture for classification

For a new coming point,

- K-means: assign the cluster membership by its distance to the cluster center.
- Gaussian mixture: assign the cluster membership weights, and select the cluster with largest weight.
- Majority vote in the selected cluster.

Reference

Elements of Statistical Learning: Section 8.5, Section 13.2 and Section 14.3.