Lecture 24

§1 Unsupervised Learning

1. Unsupervised Learning (非监督学习)

1° 适用于数据没有 structured 或 objective answers (i.e. labels) 时.
换言之，对于任一 sample (x^i, y^i)，$i = 1, 2, \cdots, N$，你只能看到 x^i，看不到 y^i.

2° Algorithm 将理解 input 并 form 一个合适的 decision.
其目的为 examine the information and identify structure within it.

- For example, the algorithm can identify customer segments who possess similar attributes. Customers within these segments can then be targeted by similar marketing campaigns.

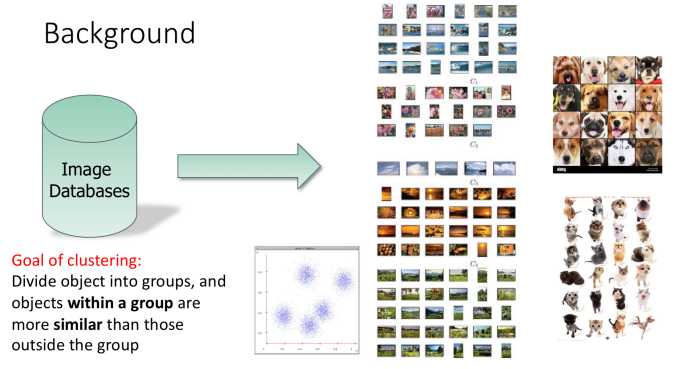
3° Popular techniques 包括 nearest-neighbor mapping, self-organizing maps, singular value decomposition and k-means clustering (k均值聚类)

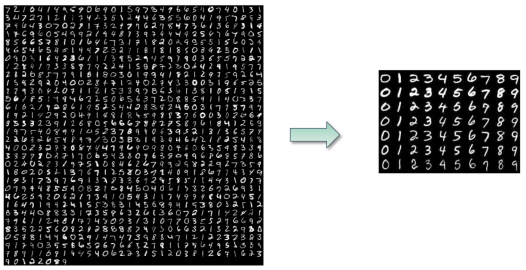
4° Algorithms 将后续用于 segment topics, identify outliers 和 recommend items.

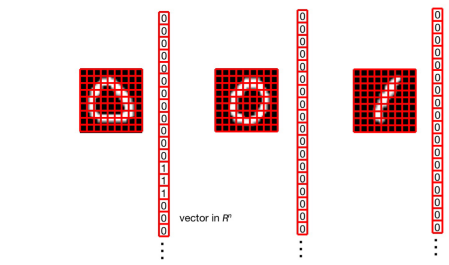
§2 Unsupervised learning: Clustering: Clustering (K-means)

1. Clustering (分类)

将物体分为不同的组，同一组内的物体有相似性.





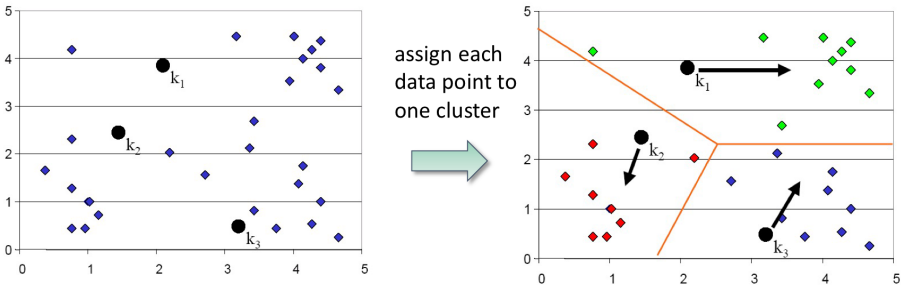


2. Formal statement of clustering problem (一般的处理方法)

- 给出 m 个 data points, $\{x^1, x^2, \cdots\cdots, x^m\}$
- 找出 k 个 cluster centers (聚类中心) $\{c^1, c^2, \cdots\cdots, c^n\}$
- assign 每个 data point i 至一个 cluster, $\pi(i) \in \{1, \cdots\cdots, k\}$ (π是一种数据点与聚类中心间的 mapping)

  使得从每个 data point 到其 cluster center 的 averaged square distances 最小

  $$\min_{c,\pi} \frac{1}{m} \sum_{i=1}^{m} |x^i - c^{\pi(i)}|^2$$ (此处的 distance 为欧氏距离)



* 对比 KNN: KNN 可以理解为 pre-define 了聚类中心, 且不会改变

3. K-means algorithm (k均值算法)

1° 随机选取 k 个 cluster centers $\{c^1, c^2, \cdots\cdots, c^k\}$

2° 分类并调整 cluster centers

① cluster assignment

将每个数据点 x^i 归入最近的 cluster centers

$$\pi(i) = \arg\min_{j=1,\cdots\cdots,k} \|x^i - c^j\|^2$$

② center adjustment

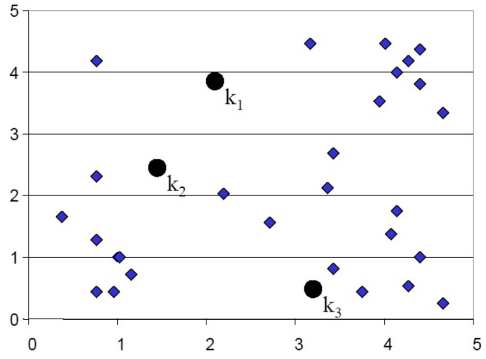
根据每一类别内的数据点, 调整每一类别的 cluster centers

(将 cluster centers 移至该类数据点的平均位置)

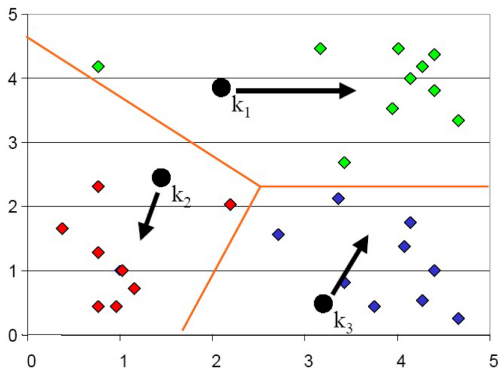
$$c^j = \underbrace{\frac{1}{|\{i: \pi(i)=j\}|}}_{\text{在 } c^j \text{ 类别内的点的个数}} \cdot \sum_{i:\pi(i)=j} x^i$$

3° 重复上一步直到 cluster centers 收敛或 π(i) 不改变

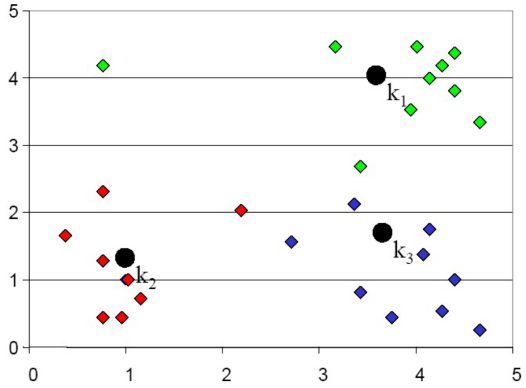
K-means: step 1



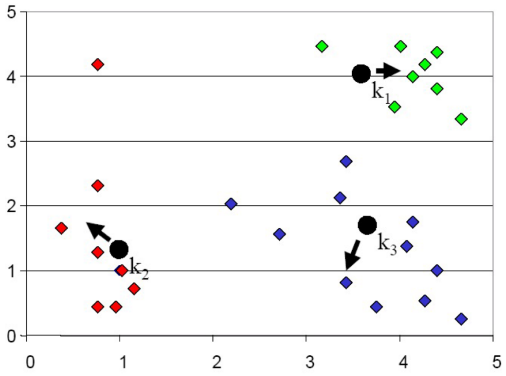
K-means: step 2



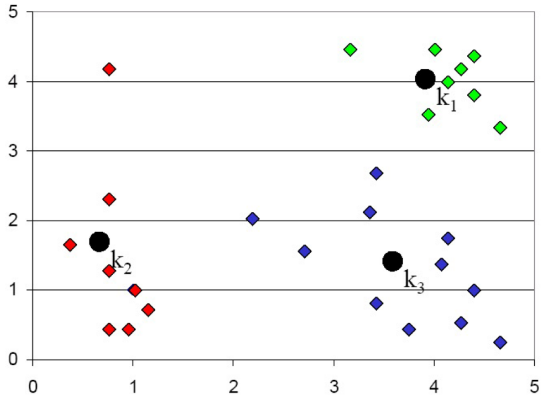
K-means: step 3



K-means: step 4



K-means: step 5



注：1° 不同的 initialization 可能会导致不同的结果

2° algorithm 永远会在 some iteration 之后停下来

* 通常 algorithm 会有 tie breaking rule，例如：若一个 data point 落在两个 center points 的中点处，该点会被 assign 给 index 较小的 center point.

4. K-means 的推广

- 给出 m 个 data points，$\{x^1, x^2, \cdots\cdots, x^m\}$
- 找出 k 个 cluster centers $\{c^1, c^2, \cdots\cdots, c^n\}$
- assign 每个 data point i 至一个 cluster，$\pi(i) \in \{1, \cdots\cdots, k\}$

使得从每个 data point 到其 cluster center 的 sum of the distances 最小

$$\min_{c,\pi} \sum_{i=1}^{m} d(x^i, c^{\pi(i)})$$ （此处的 distance 可以为任一种距离测定方式）

§3 Clustering in general

1. Clustering in general

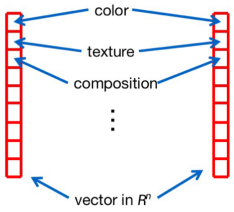
1° 选出 similarity / dissimilarity function

2° 算法根据 similarity / dissimilarity function 分组

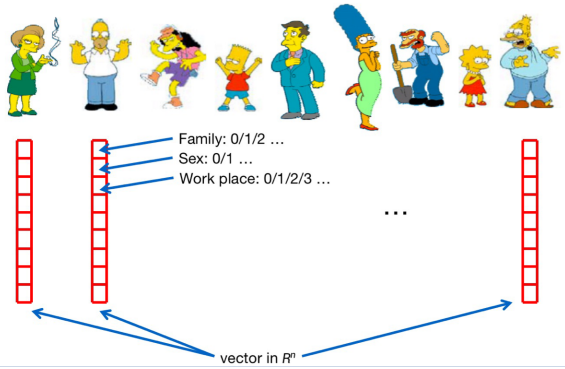
- 同一组内的点相似
- 不同组内的点不相似

2、How to represent objects

Images of different sizes



Objects in real life



3、similarity / dissimilarity function

similarity / dissimilarity function 有以下性质

1° Symmetry (对称性)

$d(x,y) = d(y,x)$

2° Positive separability

$d(x,y)=0$ 当且仅当 $x=y$

3° Triangular inequality (三角不等式)

$d(x,y) \le d(x,z) + d(z,y)$

- Desired properties of dissimilarity function
  - Symmetry: $d(x,y) = d(y,x)$
    - *Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"*
  - Positive separability: $d(x,y) = 0$, if and only if $x = y$
    - *Otherwise there are objects that are different, but you cannot tell apart*
  - Triangular inequality: $d(x,y) \le d(x,z) + d(z,y)$
    - *Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"*

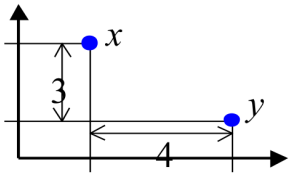
4、Distance functions for vectors

- 假定 R^n 内有两点：
  - $x = (x_1, x_2, \cdots\cdots, x_n)^T$
  - $y = (y_1, y_2, \cdots\cdots, y_n)^T$
- Euclidian distance (欧氏距离)：

  $d(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
- Minkowski distance (闵氏距离)：

  $d(x,y) = \sqrt[p]{\sum_{i=1}^n (x_i - y_i)^p}$
  - Euclidian distance：$p=2$
  - Manhattan distance：$p=1$，$d(x,y) = \sum_{i=1}^n |x_i - y_i|$ (曼哈顿距离)
  - "inf"-distance：$p=\infty$，$d(x,y) = \max_{i=1}^n |x_i - y_i|$

Distance example



- Euclidian distance: $\sqrt{4^2 + 3^2} = 5$
- Manhattan distance: $4 + 3 = 7$
- “inf”-distance: $max\{4,3\} = 4$