# K-Means Clustering

## APMA4903 Talk

Anji Zhao

9 November 2015

# Outline

# REFERENCES


- H. Daumé, *A Course in Machine Learning*, 2015.
- T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer, 2013.
- D. Hsu, Lecture Slides, *COMS 4771 - Elementary Machine Learning*, Columbia University, 2015.
- D. Arthur, S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding", *SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027-1035, 2007.
- S. D. Roy, G. Lotan, "Detecting geo-spatial weather clusters using dynamic heuristic subspaces", *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on*, 811-818, 2014.
- Wikipedia, "k-means clustering", https://en.wikipedia.org/wiki/K-means_clustering.

# Unsupervised Learning

goal: find hidden structure behind unlabeled data

examples:

- partitioning data into clusters
- dimensionality reduction

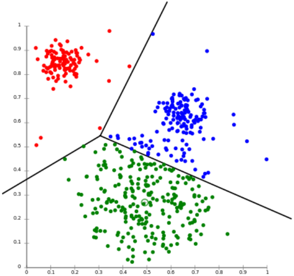
no direct measure of success

# Clustering

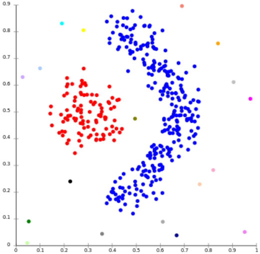
partition a dataset into groups of “similar” data points

some types of clustering:

- centroid-based (k-means)
- connectivity-based (hierarchical clustering)
- distribution-based



k-means



hierarchical

Images by Chire on Wikipedia

# K-Means Clustering

The Problem:

- input: $n$ points $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n \in \mathbb{R}^d$, $k \in \mathbb{N}$
- output: $k$ centers $\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_k \in \mathbb{R}^d$ and (optionally) $n$ cluster assignments $z_1, z_2, ..., z_n \in \{1, 2, ..., k\}$
- objective: choose $\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_k \in \mathbb{R}^d$ to minimize the within-cluster sum of squares:

$$SSE(\boldsymbol{x}, \boldsymbol{c}) = \sum_{i=1}^{n} \min_{j \in \{1,2,...,k\}} ||\boldsymbol{x}_i - \boldsymbol{c}_j||^2$$

- the k-means problem is NP-hard

# Lloyd's Algorithm

- also known as the k-means algorithm or Lloyd-Forgy algorithm
- iterative greedy algorithm which converges to a local optimum
- pseudocode:

for $k$ in range($K$):
    $\boldsymbol{c}_k$ = random vector in $\mathbb{R}^d$
repeat until convergence:
    for $i$ in range($n$):
        $z_i = argmin_k ||\boldsymbol{c}_k - \boldsymbol{x}_i||$
    for $k$ in range($K$):
        $S_k = \{\boldsymbol{x}_i : z_i = k\}$
        $\boldsymbol{c}_k = mean(S_k)$
return $\boldsymbol{c}, \boldsymbol{z}$

# Convergence

For any set of $n$ points $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n \in \mathbb{R}^d$ and number of clusters $k \in \mathbb{N}$, Lloyd's algorithm converges in a finite number of iterations.

$$
\begin{aligned}
SSE(\boldsymbol{x}, \boldsymbol{c}) &= \sum_{i=1}^{n} \min_{j \in \{1,2,...,k\}} ||\boldsymbol{x}_i - \boldsymbol{c}_j||^2 \\
&= \sum_{j=1}^{k} \sum_{i:z_i=k} ||\boldsymbol{x}_i - \boldsymbol{c}_j||^2 \\
&\text{where } z_i = argmin_k ||\boldsymbol{c}_k - \boldsymbol{x}_i||
\end{aligned}
$$

"Convergence" = SSE stops changing

quick proof on board

# Drawbacks of Lloyd's Algorithm

- converges to a local, not global minimum
- arbitrarily bad clusters depending on initialization
- yields poor results when $k$ is chosen incorrectly

# K-MEANS++ ALGORITHM

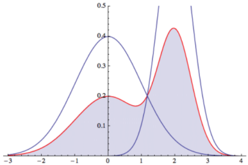
- proposed in 2007 by David Arthur and Sergei Vassilvitskii
- a method of choosing initial points to obtain more accurate clusterings than standard k-means
- algorithm:
  - choose one center $c_1$ uniformly at random from $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n$
  - until we have $k$ centers, choose center $c_i$ from the dataset, picking $x$ with probability $\frac{D(x)^2}{\sum_x D(x)^2}$, where $D(x)$ is the distance from $x$ to the closest center we have already chosen.
  - using these initial centers, proceed with the standard k-means algorithm
- guarantees $E[\phi] \leq 8(\ln k + 2)\phi_{OPT}$

# Gaussian Mixture Models

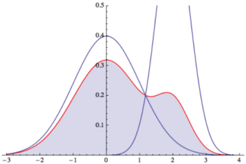
- Model dataset by a mixture of $k$ probability distributions, where the $j$th component is a Gaussian distribution with $\mu_j$ and $\Sigma_j$, $j = 1, 2, \ldots, k$.
- Formally: $(\boldsymbol{X}, Y) \sim P_\theta$, a distribution over $\mathbb{R}^d \times [k]$, where:
  - $Y \sim \boldsymbol{\pi}$
  - $\boldsymbol{X} | Y = j \sim N(\boldsymbol{\mu}_j, \Sigma_j)$
- $P_\theta$ has parameters $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \Sigma_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \Sigma_k)$
- Modeling assumption: our data $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times [k]$ is an iid sample from $P$, but we only know $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$.
- Gaussian mixture model:

$$\boldsymbol{X} \sim \sum_{j=1}^{k} \pi_j N(\boldsymbol{\mu}_j, \Sigma_j)$$

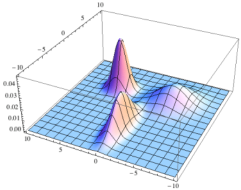
# Gaussian Mixtures in $\mathbb{R}^1$ and $\mathbb{R}^2$



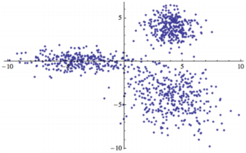

$\frac{1}{2}N(0,1) + \frac{1}{2}N(2,\frac{1}{4})$




$\frac{4}{5}N(0,1) + \frac{1}{5}N(2,\frac{1}{4})$




Mixture density




Observed sample points

Images from COMS4771 Lecture Slides

# Gaussian Probability Densities

- in one dimension ($x \in \mathbb{R}^1$):

$$p(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- in $d$ dimensions ($x \in \mathbb{R}^d$):

$$p(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = \frac{1}{\sqrt{(2\pi)^d det(\Sigma)}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- for a mixture of $k$ Gaussians in $d$ dimensions ($x \in \mathbb{R}^d$):

$$p(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = \sum_{j=1}^{k} \pi_j \frac{1}{\sqrt{(2\pi)^d det(\Sigma_j)}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^T\Sigma_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right)$$

# Soft Clustering

- instead of assigning each point to one component (as in k-means), we find the probability that the point belongs to each component
- suppose we are given the parameters of a Gaussian mixture model, and $(\boldsymbol{X}, Y) \sim P_\theta$
- let $\Phi \in \{0,1\}^k$ be the vector of assignment variables, where $\Phi_j = \mathbb{1}\{Y = j\}$
- soft assignment of a data point $\boldsymbol{x}$ to component $j$:

$$
\begin{aligned}
E_\theta[\Phi_j | \boldsymbol{X} = \boldsymbol{x}] &= Pr_\theta[Y = j | \boldsymbol{X} = \boldsymbol{x}] \\
&= \frac{Pr_\theta[Y = j] \cdot Pr_\theta[\boldsymbol{X} = \boldsymbol{x} | Y = j]}{Pr_\theta[\boldsymbol{X} = \boldsymbol{x}]} \\
&= \frac{\pi_j \cdot \sqrt{det(\Sigma_j^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right)}{\sum_{i=1}^k \left(\pi_i \cdot \sqrt{det(\Sigma_i^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)\right)\right)}
\end{aligned}
$$

# Maximum Likelihood Estimator

Maxmimum Likelihood Estimation is a method used to estimate the parameters of a model, given a set of data points.

The MLE for a model $P$ is

$$\boldsymbol{\theta}_{ML} = argmax_{\theta} \prod_{i=1}^{n} p(\boldsymbol{x}_i; \boldsymbol{\theta})$$

We can try to use the MLE to estimate the parameters of a Gaussian mixture, but we get...

$$\begin{aligned} \boldsymbol{\theta}_{ML} &= argmax_{\theta} \sum_{i=1}^{n} \ln p(\boldsymbol{x}_i, \boldsymbol{\theta}) \\ &= argmax_{\theta} \sum_{i=1}^{n} \ln (\dots) \end{aligned}$$

# Expectation-Maximization Idea

Iterative local optimization method for estimating parameters.
Given labeled data $(\boldsymbol{x}_1, \phi_1), (\boldsymbol{x}_2, \phi_2), \ldots, (\boldsymbol{x}_n, \phi_n) \in \mathbb{R}^d \times \{0,1\}^k$, the "complete log-likelihood" of $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \Sigma_1, \ldots, \pi_k, \boldsymbol{\mu}_k, \Sigma_k)$ is

$$\sum_{i=1}^{n}\sum_{j=1}^{k} \phi_{i,j} \ln\left(\pi_j \cdot \sqrt{det(\Sigma_j^{-1})} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right)\right)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{k} \phi_{i,j} \left(\ln \pi_j + \frac{1}{2}\ln det(\Sigma_j^{-1}) - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right)$$

(can also use soft assignments $w_{i,j} = E_\theta[\phi_{i,j} | \boldsymbol{X} = \boldsymbol{x}_i]$ instead of $\phi_{i,j}$)

# E-M Algorithm

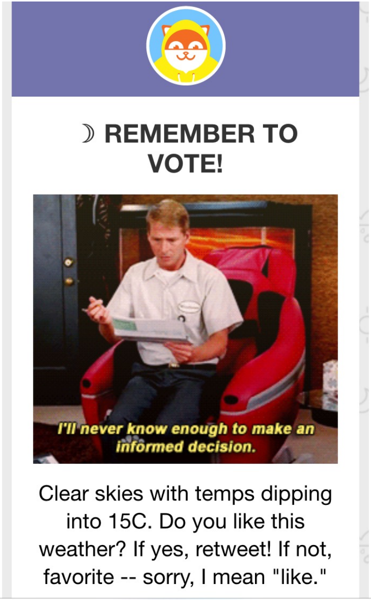
Algorithm:

- initialize $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\mu}_1, \Sigma_1, ..., \pi_k, \boldsymbol{\mu}_k, \Sigma_k)$
- E step: calculate the expectation of the unknown labels / soft assignments given the parameters $\boldsymbol{\theta}$:
  - $w_{i,j} = E_\theta[\phi_{i,j} | \boldsymbol{X} = \boldsymbol{x}_i], \ \forall i \in \{1, 2, ..., n\}, j \in \{1, 2, ..., k\}$
- M step: maximize the expected complete log-likelihood w.r.t. each parameter
  - $\pi_j = \frac{1}{n} \sum_{i=1}^n w_{i,j}$
  - $\boldsymbol{\mu}_j = \frac{1}{n\pi_j} \sum_{i=1}^n w_{i,j} \boldsymbol{x}_i$
  - $\Sigma_j = \frac{1}{n\pi_j} \sum_{i=1}^n w_{i,j} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T$

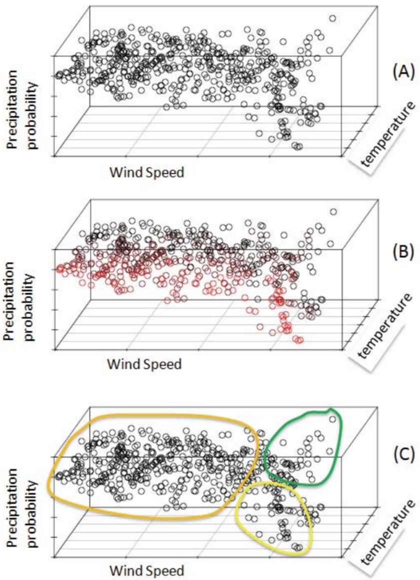
# Gaussian Mixture Models vs. K-Means

- k-means is a special case of GMM where we restrict $\Sigma_i = I \ \forall i \in [k]$ and $\pi_i = \pi_j \ \forall i, j \in [k]$
- in k-means we use hard assignment in the E-step
- k-means converges faster, but GMM is more flexible

# Application: Clustering Weather Data



# Application: Clustering Weather Data




Weather Data distribution at 12PM, Feb 16, 2011, in Stamford, CT.

A. The original data
B. Two clusters using typical k-means
C. Three clusters after heuristic splits and rounding

Image by Roy, Lotan

# Future & Questions

Current Areas of Research:

- improving performance of existing clustering algorithms
- handling high-dimensional data
- specific applications to different fields