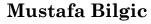
CS 584 - MACHINE LEARNING

TOPIC: CLUSTERING





♦ http://www.cs.iit.edu/~mbilgic



https://twitter.com/bilgicm

MOTIVATION

- There is no supervision i.e., there is no target variable
- Find groups / clusters in the data

VARIOUS APPROACHES

- K-Means
- Expectation Maximization (EM)
 - EM is a general approach that can be used for many other tasks, one of which is clustering
- Hierarchical clustering
- DBScan
- Spectral clustering
- Mean-shift
- o ... and many more

WE'LL COVER

- K-Means
- EM for mixture of Gaussians

4

K-MEANS

- Basic idea: given data, find k prototype vectors, and assign each data point to its closest prototype vector, such that the sum of the distances to the closest prototype vectors is minimized
- Let $D = \{x_j\}_{j=1}^N$ be the data and let $\{m_i\}_{i=1}^k$ be the prototype vectors. Objective:
 - minimize $\sum_{j=1}^{N} \left(\min_{i} distance(x_{j}, m_{i}) \right)$
 - A typical distance metric is Euclidan distance

K-MEANS

- Initialize k centers $\{m_i\}_{i=1}^k$
- Repeat until convergence
 - Assign each data point x_i to the closest cluster center
 - Re-compute each center $\{m_i\}_{i=1}^k$ as the mean of the points that are assigned to that center
- See OneNote for an illustration

EM FOR MIXTURE OF GAUSSIANS

- Basic idea: very much like the Bayes classifier where we have P(C) and p(x | C) and we use that for computing P(C | x), except, in this case, classes are not given to us. So, we take an iterative approach
- Repeat until convergence
 - Assign data points to classes/centers/components,
 <u>probabilistically</u> (i.e., compute P(C|x) using P(C) and p(x|C)) **Expectation**
 - Re-estimate class/center/component parameters based on the <u>probabilistic</u> assignments to classes/centers (i.e., recompute P(C) and p(x | C) using the probabilistic assignments) – **Maximization**



NOTATION

- \circ $P(C_i)$: prior probability of component C_i
 - A multinomial distribution
- \circ $P(x_j|C_i)$: probability that component C_i generated x_j
 - A Gaussian distribution with mean μ_i and variance σ_i^2
 - This can be easily generalized to multivariate Gaussian distribution with mean μ_i and co-variance Σ_i
- \circ $P(C_i|x_j)$: probability that x_j belongs to component C_i
 - Via Bayes rule, $P(C_i|x_j) \propto P(x_j|C_i)P(C_i)$
- o n_i : Expected number of instances that belong to component C_i
 - $n_i = \sum_{j=1}^N P(C_i|x_j)$

8

EM

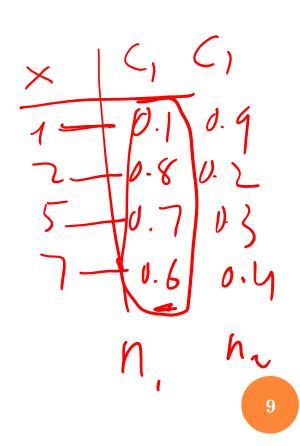
- E step: compute cluster/component assignments
 - $P(C_i|x_j) \propto P(x_j|C_i)P(C_i)$
- M step: recompute parameters
 - $n_i = \sum_{j=1}^N P(C_i|x_j)$
 - Weighted counts, where weights are $P(C_i|x_i)$
 - $P(C_i) = \frac{n_i}{N}$
 - $P(x_i|C_i) \sim N(\mu_i, \sigma_i^2)$

$$\bullet \ \mu_i = \frac{\sum_{j=1}^{N} \left(P(C_i | x_j) x_j \right)}{n_i}$$

• Weighted mean, where weights are $P(C_i|x_j)$

$$\sigma_i^2 = \frac{\sum_{j=1}^N \left(P\left(C_i \middle| x_j\right) \left(x_j - m_i\right)^2\right)}{n_i}$$

 \circ Weighted variance, where weights are $P(C_i|x_j)$



SCIKIT-LEARN

<u>http://scikit-</u>
 <u>learn.org/stable/modules/clustering.html</u>