Data Clustering: A Very Brief Overview

Serhan Cosar
INRIA-STARS

Outline

- Introduction

- Five Ws of Clustering

  Who, What, When, Where, Why?

- One H of Clustering

  How?

- Algorithms

- Conclusion

Introduction

- Unsupervised Learning: a very important problem in machine learning
  - Big amount of data
  - Unlabeled data
    - Time and effort to label
    - Not enough information to label
- Data Mining: an interdisciplinary field in computer science
  - A very large set of data in a database
  - Intersection of
    - Machine learning
    - Database systems

Introduction

- Some examples
  - Classification of plants given their features
  - Finding patterns in a DNA sequence
  - Recognizing objects, actions from images
  - Image segmentation
  - Document classification
  - Customer shopping patterns
  - Analyzing web searching patterns

5Ws of Clustering

- *Who, What, When, Where, Why?*
- As a researcher, you are given a (large) set of points without labels
- Grouping unlabeled data
  - Points within each cluster should be similar (close) to each other
  - Points from different clusters should be dissimilar (far)

5Ws of Clustering

- Given points are usually in a high-dimensional space
- Similarity is defined using a distance measure
  - Euclidean Distance,
  - Mahalanobis Distance,
  - Minkowski Distance,
  - ...

1H of Clustering

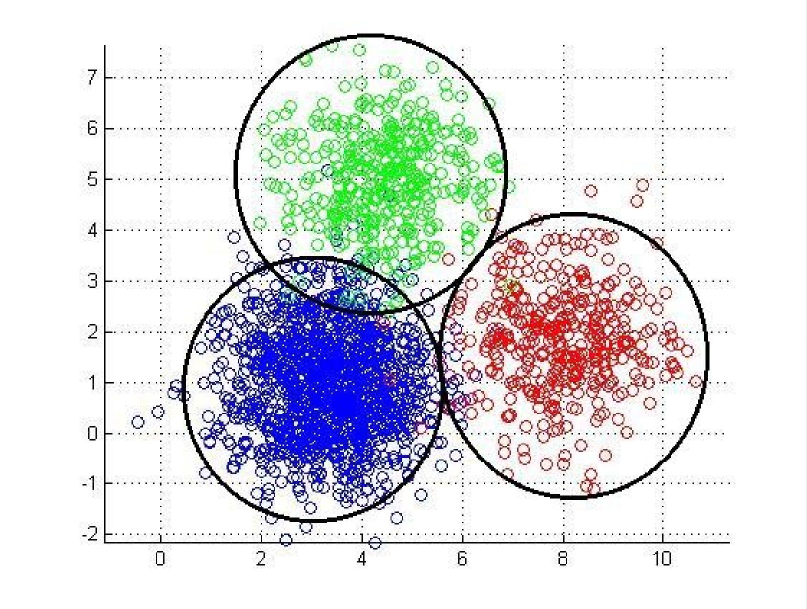
- *How do we cluster?*
- In general two types of algorithms:
  - Partition Algorithms
    - Obtain a single level of *partition*
  - Hierarchical Algorithms
    - Obtain a *hierarchy* of clusters

Partition Algorithms

- K-Means
  - Set the number of clusters (*k*)
    - Initialize *k* centroids
    - Group points close to centroid

$$\sum_{i=0}^{N} \min_{\mu_j \in C} \left(\left\| x_i - m_j \right\|^2 \right)$$

    - Re-calculate centroids
  - Always converges (may be to local minimum)
    - Kmeans++
  - Not highly scalable, Computation
    - Minibatch K-means



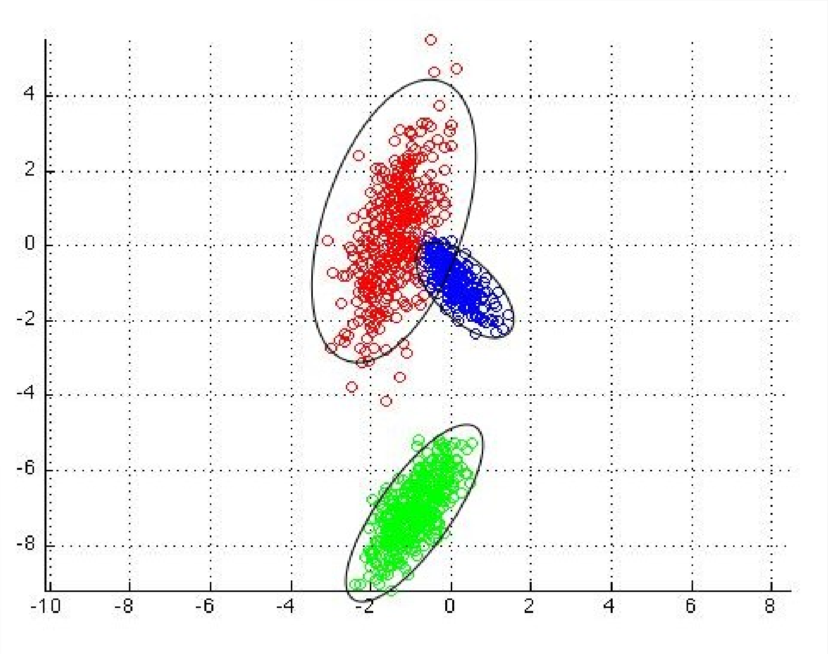
Partition Algorithms

- Mean Shift
  - Set the bandwidth (max. distance)

$$\|x_i - m_j\|^2 \le BW^2$$

- Mixture of Gaussian
  - Mahalanobis distance

$$\sum_{i=0}^{N} \min_{\mu_j \in C}\left(\left(x_i-\mu_j\right)^T \Sigma_j^{-1}\left(x_i-\mu_j\right)\right)$$

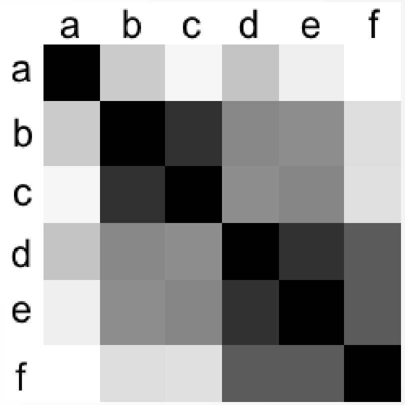


- Not highly scalable

Partition Algorithms

- Spectral Clustering
  - Set the number of clusters (*k*)
  - Similarity Matrix (pair-wise distance)

$$L = D - S \qquad D_{ii} = \sum_j S_{ij}$$

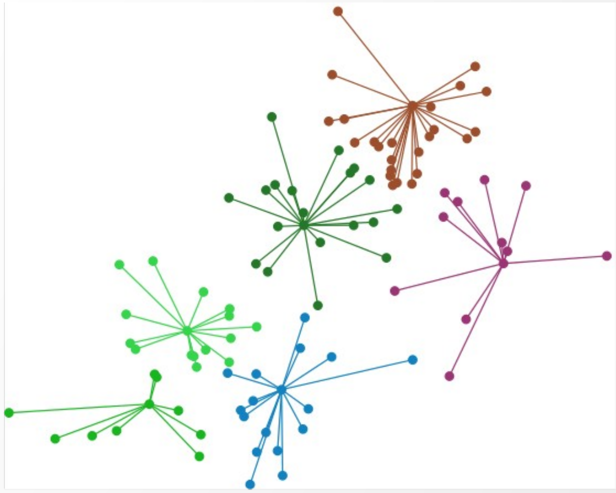


  - Laplacian Matrix
    - Eigenvalues $0 = \lambda_1 \leq \ldots \leq \lambda_n$
  - Take first *k* eigenvectors and cluster using K-means
  - Eigenvector computation could be a problem for large datasets

informatics mathematics Inria

Partition Algorithms

- Affinity Propagation
  - No need to specify number of clusters
  - Similarity Matrix
  - Responsibility Matrix
    - r(i,k) -> Quantify how well x_k will be to serve as “exemplar” for x_i
  - Availability Matrix
    - a(i,k) -> Quantify how appropriate it will be for x_i to pick x_k as its “exemplar”
  - “Message-passing” between data points
    - Initialize matrices R and A to zero
    - Iteratively update



$$r(i,k) \leftarrow s(i,k) - \max_{\acute{k} \neq k} \{a(i,\acute{k}) + s(i,\acute{k})\}$$

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{\acute{i} \notin i,k} \max\{0, r(\acute{i},k)\}\}$$

Partition Algorithms

- Affinity Propagation
  - Computation complexity
    - Time
    - Memory
  - Not suitable for large datasets

How do we cluster?

- In general two types of algorithms:
  - Partition Algorithms
    - Obtain a single level of *partition*
  - Hierarchical Algorithms
    - Obtain a *hierarchy* of clusters

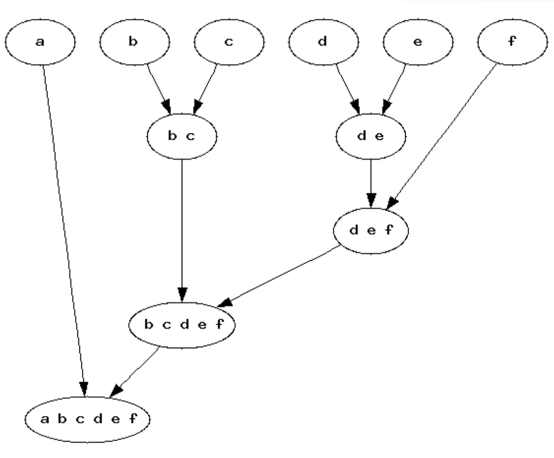
Hierarchical Algorithms

- Bottom up – agglomerative
  - Iteratively merging small clusters into larger ones
- Top down – divise
  - Iteratively splitting larger clusters

- Can scale to large number of samples

Bottom up Algorithms

- Incrementally build larger clusters out of smaller clusters
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there's only one cluster left
  - Obtain *dendrogram*



- Need to define "closeness" (metric and linkage criteria)

Bottom up Algorithms

- Linkage criteria
  - Ward: minimizing the sum of squared differences within all clusters (~K-means)
  - Single linkage: minimizes the distance between samples in a cluster (~K-NN)
  - Complete linkage: minimizes the maximum distance between samples in a cluster
  - Average linkage: minimizes the average of distances between samples in a cluster
- Distance Metric

Top down Algorithms

- Put all samples in one cluster and iteratively split the clusters
  - Distance metric to measure dissimilarity

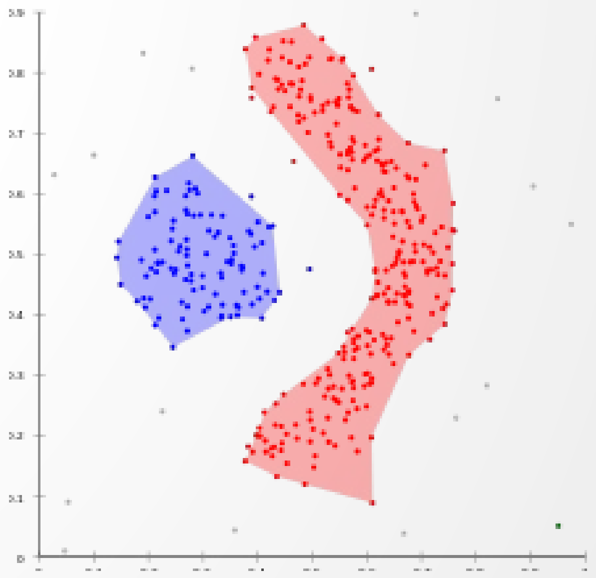
Other Algorithms

- DBSCAN*
  - *Core* samples: samples that are very close to each other
  - *Non-core* samples: samples that are close to *core* samples (except *core* samples themselves)
  - Set *epsilon (ε)* (distance) and *min. number of samples* to form a dense region
    - Take an arbitrary point
    - Check its *ε*-neighborhood
      - If it contains more samples than *min. number of samples*, create a cluster
      - If not mark as noise (outlier)

*Density-based spatial clustering of applications with noise

Other Algorithms

- DBSCAN
  - Can find arbitrarily shaped clusters
  - Can detect outliers
  - Can scale to very large datasets



Conclusion

- Clustering is a huge domain
- Need to select the approach suitable for the problem
  - Parameters to set (e.g., number of clusters)
  - Data geometry
  - Convergence: local / global optimum
  - Number of samples
  - Computation time

Conclusion

- Clustering performance evaluation
  - Adjusted Rand Index

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

  - Mutual Information
  - Homogeneity, completeness
  - Silhouette Coefficient
  - Davies-Bouldin Index

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

  - ...

THANK YOU

- References
  - Scikit-learn: Python Library

    http://scikit-learn.org/stable/modules/clustering.html
  - Anil K. Jain, M. N. Murty, and P. J. Flynn. “Data clustering: a review”, ACM Computing Surveys, 31(3):264–323, 1999
  - Nizar Grira, Michel Crucianu, Nozha Boujemaa, “Unsupervised and Semi-supervised Clustering: a Brief Survey”, A Review of Machine Learning Techniques for Processing Multimedia Content
  - Brendan J. Frey and Delbert Dueck, “Clustering by Passing Messages Between Data Points”, Science Feb. 2007

Inria informatics mathematics