**Clustering Algorithms**

1. Partition-based Clustering – relatively efficient
   * K-means
   * k-median
   * fuzzy c-means
2. Mixture Models (maybe part of partition-based clustering)
   * Model: **Mixture of Gaussians**, algorithm: EM – Expectation Maximization
3. Mix Membership Models (maybe part of partition-based clustering)
   * Model: **LDA (Latent Dirichlet allocation)**, Algorithm: Gibbs Sampling Algorithm
4. Hierarchical Clustering – produces trees of clusters
   * **Divisive** - a.k.a top-down. Example: recursive k-means
   * **Agglomerative** - a.k.a. bottom-up. Example: single linkage
5. Density-based Clustering – produces arbitrary shaped clusters
   * **DBSCAN**

**Applications of clustering**

Retail marketing:

* Identify buying patterns of customers
* Recommending new books or movies to new customers

Banking:

* Fraud detection in credit card use
* Identifying clusters of customers (eg. Loyal)

Insurance

* Fraud detection in claim analysis
* Insurance risk of customers

Medicine

* Characterizing patient behavior

Biology

* Clustering genetic markers to identify family ties

**Why clustering?**

* Exploratory data analysis
* Summary generation
* Outlier detection
* Finding duplicates
* Pre-processing step

**RETRIEVAL**

**Nearest Neighbor search:**

1. Brute force
2. KD - tree algorithm - problem with high dimensional data
3. LSH Algorithm - Locality sensitive hashing.
   * exact NN search is not important in most applications, so focus on approximation

Word count document representations:

1. **Bag of words model** - issue - common words like "the" dominate
2. **TF-IDF** document representation - focuses on important words (common locally, rare globally) (term frequency - common locally, inverse doc freq. - rare globally)

Similarity metrics => distance = 1 - similarity

1. in 1D - euclidean distance = abs(Xi - Xq)
2. in multi D - non-scaled euclidean
3. in multi D - scaled euclidean
4. inner product = transpose(Xi) Xq
5. cosine similarity
6. mahalanobis, rank-based, correlation-based, manhattan, jaccard, hamming

Normalizing:

* normalizing is good in general
* but problems when comparing long, short docs

Weighting different features:

1. scaled
2. non-scaled

**CLUSTERING**

1. Partition-based Clustering – relatively efficient

**1) K-means algorithm** - Divides into non-overlapping clusters

**2) K-means++** - smart initialization. costly than random initialization, but converges faster

1. Randomly place k centroids, one for each cluster
2. Calculate the distance of each point from each centroid
3. Assign each data point (object) to its closest centroid, creating a cluster
4. Recalculate the position of the k centroids
5. Repeat the steps 2-4, until the centroids no longer move
6. Mixture Models
   * **Mixture of Gaussians model**- probabilistic approach, soft assignments, accounts for cluster shapes not just centers
   * **EM algorithm** used to solve

EM Algorithm:

Part 1)

--> estimate cluster responsibilities (uses Bayes' rule to calculate responsibilitieS(soft assignments)

Part 2)

-- Maximum Likelihood Estimation (MLE) - MLE from soft assignment

Relationship to k-means

1. Mix Membership Models

* want to discover a set of memberships
* **LDA model (Latent Dirichlet allocation)**
* **Gibbs Sampling Algorithm** used to solve
* LDA model - Normally LDA is specified as a Bayesian model
* LDA requires documents to be represented as a bag of words
* corpus wide topic prevalence, topic specific word probabilities
* Gibbs sampling for Bayesian inference
* Gibbs sampling - iterative random hard assignment

Gibbs sampling steps:

1. words in a doc (z(iw)): randomly reassign all z(iw) based on doc topic proportions and topic vocab distributions
2. doc (same as above) - randomly reassign doc topic proportions based on (step 1) assignments z(iw) in current doc.
3. docs - repeat for all docs
4. topics - randomly reassign topic vocab distributions based on assignments z(iw) in entire corpus

REPEAT (Steps 1-4) UNTIL MAX ITER Reached

Collapsed Gibbs Sampling for LDA

-- How much doc “likes” each topic based on other assignments in doc

-- How much each topic likes the word “dynamic” based on assignments in other docs in corpus

1. Hierarchical Clustering – produces trees of clusters
2. Divisive
3. Agglomerative

**Divisive**

* + Top-down - Start with all data in one big cluster and recursively split.
  + Example: recursive k-means

Choices to make:

* Which algorithm to recurse
* How many clusters per split
* When to split vs. stop

**Agglomerative**

* Bottom-up: Start with each data point as its own cluster. Merge clusters until all points are in one big cluster
* Example: single linkage

Agglomerative algorithm:

1. Create n clusters, one for each data point
2. Compute the Proximity Matrix
3. Repeat
   1. Merge the two closest clusters
   2. Update the proximity matrix
4. Until only a single cluster remains

Distance between clusters

1. Single-Linkage Clustering
   1. Minimum distance between clusters
2. Complete-Linkage Clustering
   1. Maximum distance between clusters
3. Average-Linkage Clustering
   1. Average distance between clusters
4. Centroid-Linkage Clustering
   1. Distance between cluster centroids

Single linkage algorithm:

1. Initialize each point to be its own cluster
2. Define distance between clusters to be: distance(C1,C2) =min d(xi, xj) where xi in C1, xj in C2
3. Merge the two closest clusters
4. Repeat step 3 until all points are in one cluster

Choices to be made for sla algorithm:

1. Distance metric
2. Linkage function
3. Where and how to cut dendrogram

Computational considerations

• Computing all pairs of distances is expensive - Brute force algorithm is O(N2log(N))

• Smart implementations use triangle inequality to rule out candidate pairs

• Best known algorithm is O(N2)

The dendrogram for agglomerative clustering

1. Density-based Clustering – produces arbitrary shaped clusters

Most of the traditional clustering techniques, such as k-means, hierarchical and fuzzy clustering, can be used to group data without supervision. However, when applied to tasks with arbitrary shape clusters, or clusters within cluster, the traditional techniques might be unable to achieve good results. That is, elements in the same cluster might not share enough similarity or the performance may be poor. Additionally, Density-based Clustering locates regions of high density that are separated from one another by regions of low density. Density, in this context, is defined as the number of points within a specified radius.

* Arbitrary-shape clusters
* Density based clustering locates regions of high density and separates outliers

**DBSCAN – Density Based Spatial Clustering of Applications with Noise**

2 parameters:

1. R – (Radius of Neighborhood)
   1. Radius (R) that if includes enough number of points within, we call it a dense area
2. M – (Min number of neighbors)
   1. determine the minimum number of data points we want in a neighborhood to define a cluster

Advantages:

* Arbitrarily shaped cluster
* Robust to outliers
* Does not require specification of the number of clusters

ADDITIONAL INFO:

Hidden Markov models (HMMs)

So far, looked at clustering unordered data

-- Data index (i.e., when observation was recorded) does not influence clustering

-- What if we have time series data?

As in mixture model…

-- Every observation xt is associated with cluster assignment variable zt

-- Each cluster has a distribution over observed values

Difference from mixture model:

Probability of (z(t) = k) depends on previous cluster assignment z(t-1)

Inference in HMMs

• Learn MLE of HMM parameters using EM algorithm = Baum Welch

• Infer MLE of state sequence given fixed model parameters using dynamic programming = Viterbi algorithm

• Infer soft assignments of state sequence using dynamic programming = forward-backward algorithm

What we didnt cover:

Retrieval:

- Other distance metrics

- Distance metric learning

Clustering:

- Nonparametric clustering

- Spectral clustering

Related ideas:

- Density estimation

- Anomaly detection

5. Recommender Systems & Dimensionality Reduction

Models:

• Collaborative filtering

• Matrix factorization

• PCA

Algorithms:

• Coordinate descent

• Eigen decomposition

• SVD

Concepts:

• Matrix completion, eigenvalues, cold-start problem, diversity, scaling up