**Preliminaries**:

; ; ; ; 🡪

**PCA**:

Goal: To find an r-dimensional basis to reduce to that minimizes the *mean squared error* . Total Projected Variance: . Objective: . Choosing Percentile: . Speed: Covariance matrix computation: , Eigen-value decomposition: , PCA is: ; 🡪sum of products of a row in x and a row in U ;

**K-means**:

Goal: minimize the distance between each point with the mean of each point’s cluster assigned. Pros: Fast, Great when makes sense or spherical (Euclidean) distance is okay, tries to optimize global goal and achieves a local optimum. Cons: Doesn’t account for variance, K must be decided. Speed: . Objective:

**EM:**

Goal: maximize likelihood of each point in D to a set of clusters (Gaussians) defined by . ) = P(. Pros: Accuracy, Uses percentages instead of discrete values. Cons: Slow, Bad for concave data. Speed:

**Graph Clustering**:

Goal: minimize the weight of the edges cut over either number of nodes in a cluster (ratio-cut) or the sum of the weights *only* in a cluster (normalized radio-cut). Pros: No underlying assumptions, works well with clusters of different sizes. Cons: slow, clusters cannot be convex. Objective: **Ratio-cut**: , **Normalized Ratio-cut**: ; sum of weights on the cut over nodes in .

, thus the k smallest eigenvalues tell us the minimum ratio cut of L.

**Hierarchical Clustering:**

Goal: create a sequence of nested clusters that are subsets nested within another cluster that contains the previous. . At some intermediate level we will find meaningful clusters. Agglomerative: All points start as a cluster, then merge 2 closest clusters into 1 new . Divisive:. # Clusters for N objects: . Single Link: min distance between two points not in the same cluster; merge unite. Complete Link: max distance between two points not in the same cluster; merge unite. Group Average: . Group Average: . Pros: Finds K, eventually finds meaningful results, good for discrete data. Cons: Slow, greedy and potentially suboptimal.

**Clustering Validation**: Clustering Tendency: The higher the divergence, the better the clustering tendency of the dataset. KL-Divergence:

**Density clustering**: Objective: Mining non-convex clusters to account sensitivity to noise. DBSCAN: Uses local density of points to determine a cluster: , if that set contains *minpts* we call it **a core point**, **a border point** is not a core point, but has a core point in its -neighborhood, and **a** **noise point** doesn’t have a core or border point in its -neighborhood. Speed: . Pros: Fixes the issue representative clustering had in that two points from different clusters may be closer, merging criteria is more sensitive to noises. Cons: Cluster quality depends on the choice of parameters, does not work well if the clusters have non-uniform density. If too small, legitimate clusters can be treated as noise. If too big, legitimate micro-clusters get joined into 1 cluster.