Non-exhaustive, Overlapping k-means

Joyce Jiyoung Whang [1], Inderjit S. Dhillon [1], and David F. Gleich [2]
[1]The University of Texas at Austin, [2]Purdue University
SIAM International Conference on Data Mining (SDM), 2015.

Center for Big Data Analytics
http://bigdata.ices.utexas.edu

Problem Statement

- Clustering is the task of finding a set of cohesive data points.
- Traditional disjoint, exhaustive clustering
  - Every single data point is assigned to exactly one cluster.
- Non-exhaustive, overlapping clustering
  - A data point is allowed to be outside of any cluster.
  - Clusters are allowed to overlap with each other.

Main Contributions

- NEO-K-Means (Non-Exhaustive, Overlapping K-Means)
- Overlap and non-exhaustiveness – handled in a unified framework
- Simple and intuitive objective function
- Simple iterative algorithm that monotonically decreases the objective function
- Weighted Kernel NEO-K-Means
  - Non-exhaustive, overlapping graph clustering (overlapping community detection)
- Effective in finding ground-truth clusters

Revisit k-means

- k-means seeks k clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$ in $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_n\}$.
  - $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \ \forall i \neq j$ (disjoint), $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \cup \mathcal{C}_k = \mathcal{X}$ (exhaustive).
- Objective function: $\min_{\{\mathcal{C}_j\}_{j=1}^k} \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathcal{C}_j} \|\mathbf{x}_i - \mathbf{m}_j\|^2$, where $\mathbf{m}_j = \dfrac{\sum_{\mathbf{x}_i \in \mathcal{C}_j} \mathbf{x}_i}{|\mathcal{C}_j|}$.
- Lloyd's algorithm: repeatedly assigning data points to their closest clusters and recomputing centers until the change in the objective value becomes small.

The NEO-K-Means Objective

- Assignment matrix $U = [u_{ij}]_{n \times k}$: $u_{ij} = 1$ if $\mathbf{x}_i$ belongs to cluster j, $u_{ij} = 0$ otherwise.
  - $(U\mathbf{1})_i$: the number of clusters $\mathbf{x}_i$ belongs to
  - $trace(U^T U)$: the sum of cluster sizes (the total number of assignments)

$$U = \begin{matrix} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} \begin{matrix} c_1\ c_2\ c_3 \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{matrix} \quad U\mathbf{1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 2 \end{bmatrix} \quad U^T U = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

no. of clusters a data point belongs to

cluster sizes

- The NEO-K-Means objective function

$$\min_U \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2, \text{ where } \mathbf{m}_j = \frac{\sum_{i=1}^n u_{ij} \mathbf{x}_i}{\sum_{i=1}^n u_{ij}}$$
$$\text{s.t. } trace(U^T U) = (1+\alpha)n, \ \sum_{i=1}^n \mathbb{I}\{(U\mathbf{1})_i = 0\} \leq \beta n.$$

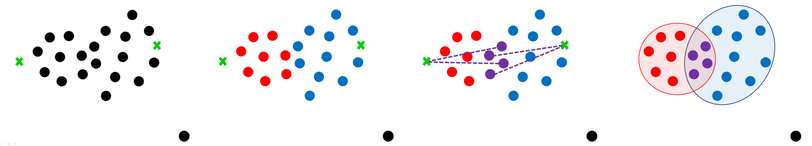
  - α and β control the degree of overlap and non-exhaustiveness, respectively.
  - $n + \alpha n$ assignments are made in total.
  - At most βn data points can have no membership in any cluster.
  - $\alpha = 0, \beta = 0$: equivalent to the standard k-means objective

$$\min_U \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2, \text{ where } \mathbf{m}_j = \frac{\sum_{i=1}^n u_{ij} \mathbf{x}_i}{\sum_{i=1}^n u_{ij}} \text{ s.t. } \hat{U}^T \hat{U} = I, \mathbb{I}\{(U\mathbf{1})_i = 0\} = 0,$$

where $\hat{U} = [\frac{\mathbf{u}_1}{\sqrt{n_1}}, \cdots, \frac{\mathbf{u}_k}{\sqrt{n_k}}]$, $\mathbf{u}_c$ is the c-th column of U, and n_c is the size of cluster c.

The NEO-K-Means Algorithm

- The NEO-K-Means Algorithm is a simple iterative algorithm that monotonically decreases the NEO-K-Means objective.
  - $\alpha = 0, \beta = 0$: identical to the standard k-means algorithm
- The NEO-K-Means Algorithm
  - Compute distances between every data point and clusters $[d_{ij}]_{n \times k}$
  - Assign $n - \beta n$ data points to their closest clusters
  - Make $\beta n + \alpha n$ assignments by taking minimum distances among $[d_{ij}]_{n \times k}$
  - Repeat until convergence
- Example ($n = 20, \alpha = 0.15, \beta = 0.05$)
  - Assign $n - \beta n$ (=19) data points to their closest clusters.
  - Make $\beta n + \alpha n$ (=4) assignments by taking minimum distances.



Parameter Selection

- α and β are intuitive parameters that allow users to specify how much overlap and non-exhaustiveness they want. These parameters can be estimated by running a disjoint, exhaustive clustering and performing cheap distance-based computation.

Normalized Cut for Overlapping Community Detection

- Community is a set of cohesive vertices that have more connections inside the set than outside. Community detection can be thought as a graph clustering problem where each community corresponds to a cluster in a graph.
- Normalized Cut is a popular graph clustering objective (disjoint & exhaustive).
  - k pairwise disjoint clusters in $G = (\mathcal{V}, \mathcal{E})$ such that $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \cup \mathcal{C}_k = \mathcal{V}$.

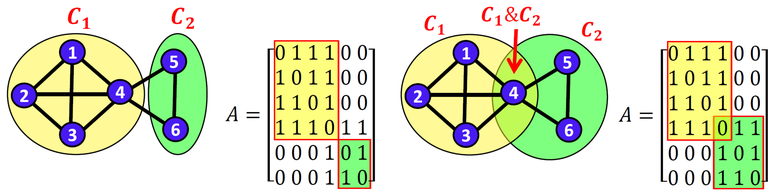
$$\texttt{ncut}(G) = \min_{\mathcal{C}_1, \mathcal{C}_2, \cdots \mathcal{C}_k} \sum_{j=1}^{k} \frac{\texttt{links}(\mathcal{C}_j, \mathcal{V} \backslash \mathcal{C}_j)}{\texttt{links}(\mathcal{C}_j, \mathcal{V})} = \max_{\mathbf{y}_1, \mathbf{y}_2, \cdots \mathbf{y}_k} \sum_{j=1}^{k} \frac{\mathbf{y}_j^T A \mathbf{y}_j}{\mathbf{y}_j^T D \mathbf{y}_j}$$

  - A: adjacency matrix, $\texttt{links}(\mathcal{P}, \mathcal{Q}) = \sum_{i \in \mathcal{P}, j \in \mathcal{Q}} A_{ij}$
  - D: the diagonal matrix of vertex degrees, $\mathbf{y}_j$: an indicator vector for cluster j
- Extending Normalized Cut to Non-exhaustive, Overlapping Graph Clustering
  - Assignment matrix $Y = [y_{ij}]_{n \times k}$: $y_{ij} = 1$ if v_i belongs to cluster j, $y_{ij} = 0$ otherwise.

$$\max_Y \sum_{j=1}^{k} \frac{\mathbf{y}_j^T A \mathbf{y}_j}{\mathbf{y}_j^T D \mathbf{y}_j} \text{ s.t. } trace(Y^T Y) = (1+\alpha)n, \ \sum_{i=1}^n \mathbb{I}\{(Y\mathbf{1})_i = 0\} \leq \beta n.$$

  - $\alpha = 0, \beta = 0$: equivalent to the traditional normalized cut
- Example: normalized cut values for two disjoint/overlapping communities

Figure: $\texttt{ncut}(G) = \dfrac{\texttt{links}(\mathcal{C}_1, \mathcal{V} \backslash \mathcal{C}_1)}{\texttt{links}(\mathcal{C}_1, \mathcal{V})} + \dfrac{\texttt{links}(\mathcal{C}_2, \mathcal{V} \backslash \mathcal{C}_2)}{\texttt{links}(\mathcal{C}_2, \mathcal{V})}$.



(a) Disjoint communities: $\texttt{ncut}(G) = \frac{2}{14} + \frac{2}{4}$ (b) Overlapping communities: $\texttt{ncut}(G) = \frac{2}{14} + \frac{3}{9}$

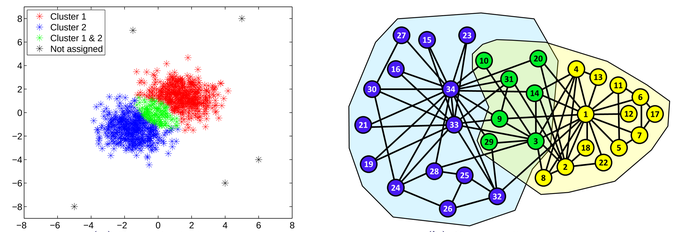
Overlapping Community Detection Using NEO-K-Means

- Weighted Kernel NEO-K-Means objective

$$\min_U \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij} w_i \|\phi(\mathbf{x}_i) - \mathbf{m}_j\|^2 = \min_U \sum_{j=1}^{k} \left(\sum_{i=1}^{n} u_{ij} w_i K_{ii} - \frac{\mathbf{u}_j^T W K W \mathbf{u}_j}{\mathbf{u}_j^T W \mathbf{u}_j} \right),$$
$$\text{where } \mathbf{m}_j = \frac{\sum_{i=1}^n u_{ij} w_i \phi(\mathbf{x}_i)}{\sum_{i=1}^n u_{ij} w_i}, \text{ s.t. } trace(U^T U) = (1+\alpha)n, \ \sum_{i=1}^n \mathbb{I}\{(U\mathbf{1})_i = 0\} \leq \beta n.$$

  - ϕ: a nonlinear mapping, w_i: a non-negative weight for data point $\mathbf{x}_i$
  - K: kernel matrix, i.e., $K_{ij} = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, W: the diagonal matrix of vertex weights
- Weighted Kernel NEO-K-Means objective is equivalent to the extended normalized cut objective (weight $W \equiv D$, kernel $K \equiv \sigma D^{-1} + D^{-1} A D^{-1}$).
- Weighted kernel NEO-K-Means algorithm provides a principled way to find a set of overlapping communities in networks.

Clustering results of NEO-K-Means



(a) Data clustering (b) Community detection

Figure: (a) NEO-K-Means correctly detects the outliers and finds the underlying overlapping clusters. (b) NEO-K-Means reveals the natural overlapping communities on the Karate club network.

Experimental Results on Data Clustering

Table: F_1 scores on vector datasets. NEO-K-Means achieves the highest F_1 score across all the datasets while the performance of other existing algorithms is not consistent across all the datasets.

	moc	*fuzzy*	*esp*	*isp*	*okm*	*rokm*	NEO
synth1	0.833	0.959	0.977	0.985	0.989	0.969	**0.996**
synth2	0.836	0.957	0.952	0.973	0.967	0.975	**0.996**
synth3	0.547	0.919	0.968	0.952	0.970	0.928	**0.996**
yeast	N/A	0.308	0.289	0.203	0.311	0.203	**0.366**
music	0.534	0.533	0.527	0.508	0.527	0.454	**0.550**
scene	0.467	0.431	0.572	0.586	0.571	0.593	**0.626**

Experimental Results on Overlapping Community Detection

Table: Average normalized cut of each algorithm on large real-world networks. Lower normalized cut indicates better communities. NEO-K-Means achieves the lowest normalized cut on all the datasets.

	demon	*oslom*	*bigclam*	*nise*	NEO
Amazon	0.555	0.221	0.392	0.116	**0.105**
DBLP	0.606	0.355	0.617	0.204	**0.188**
Flickr	N/A	N/A	0.596	0.515	**0.331**
LiveJournal	N/A	N/A	0.912	0.643	**0.373**

Future Work

- We plan to extend the NEO-K-Means method to the co-clustering problem and clustering with other Bregman divergences. We also plan to develop semidefinite programming models to optimize the NEO-K-Means objective.

J. J. Whang, I. S. Dhillon, and D. F. Gleich. Non-exhaustive, Overlapping k-means. In SDM, 2015.