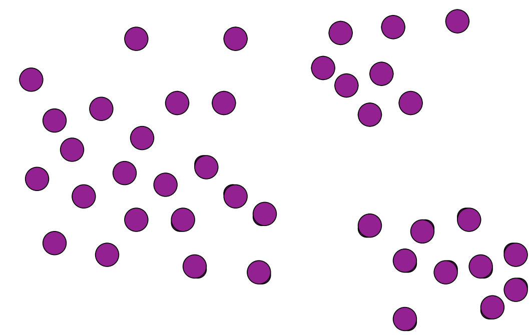
Hierarchical Divisive: Template

1. Put all objects in one cluster
2. Repeat until all clusters are singletons
   a) choose a cluster to split
      - what criterion?
   b) replace the chosen cluster with the sub-clusters
      - split into how many?
      - how split?
      - “reversing” agglomerative => split in two
- cutting operation: cut-based measures seem to be a natural choice.
  – focus on similarity across cut - lost similarity
- not necessary to use a cut-based measure

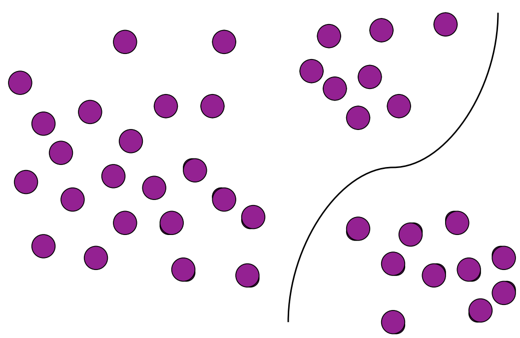
1

An Example



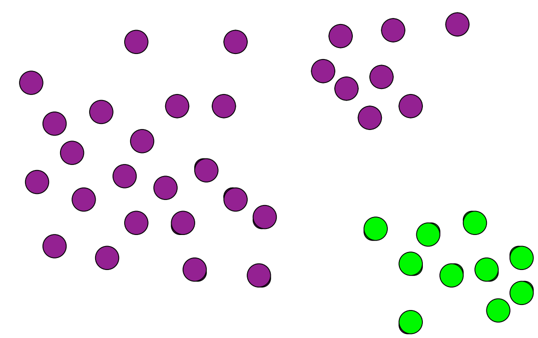
2

An Example: 1^{st} cut



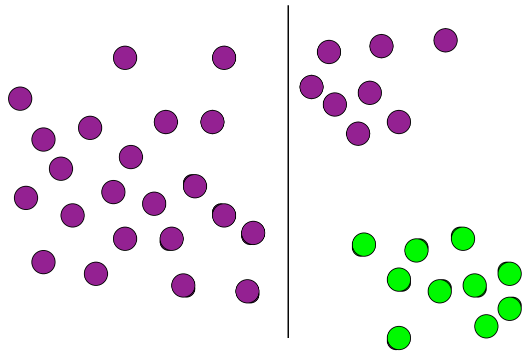
3

An Example: result of 1^{st} cut



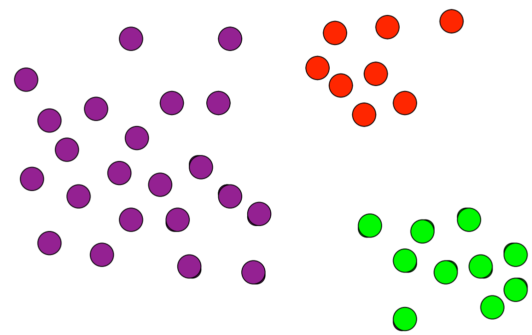
4

An Example: 2nd cut



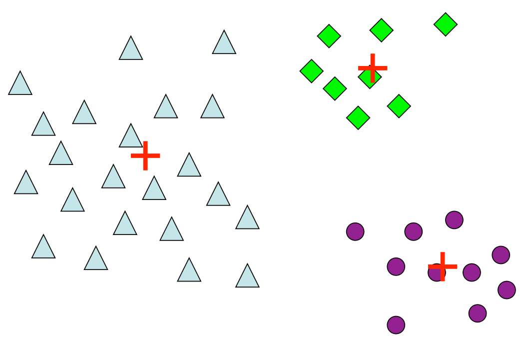
5

An Example: stop at 3 clusters



6

Compare k-means result



7

Cut-based optimization

- focus on weak connections between objects in different clusters *rather than* strong connections between objects within a cluster

- Are many cut-based measures
- We will look at two

8

Inter / Intra cluster costs

Given:

- $V = \{v_1, \dots, v_n\}$, the set of all objects
- A partitioning clustering $C_1, C_2, \dots C_k$ of the objects:

$$V = \cup_{i=1, \dots, k} C_i .$$

Define:

- $\text{cutcost}(C_p) = \sum_{\substack{v_i \text{ in } C_p \\ v_j \text{ in } V-C_p}} \text{sim}(v_i, v_j).$

- $\text{intracost}(C_p) = \sum_{(v_i, v_j) \text{ in } C_p} \text{sim}(v_i, v_j).$

Cost of a clustering

total relative cut cost $(C_1, \dots, C_k)$ =

$$\sum_{p=1}^{k} \frac{\text{cutcost}(C_p)}{\text{intracost}(C_p)}$$

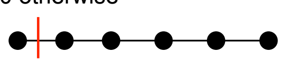
- contribution each cluster:
  ratio external similarity to internal similarity

Optimization

Find clustering $C_1, \dots, C_k$ that minimizes
total relative cut cost$(C_1, \dots, C_k)$

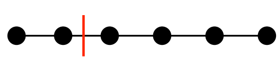
Simple example

- six objects
- similarity 1 if edge shown
- similarity 0 otherwise
- choice 1:



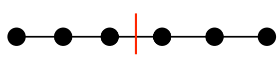
cost UNDEFINED (0?) + 1/4

- choice 2:



cost 1/1 + 1/3 = 4/3

- choice 3:



cost 1/2 + 1/2 = 1 *prefer balance

Second cut-based measure: Conductance

- define:
  $\text{s\_degree}(C_p) = \text{cutcost}(C_p)+2*\text{intracost}(C_p)$
  - model as graph, similarity = edge weights
  - s_degree is sum over all vertices in C_p of weights of edges touching vertex

- conductance (C_p) =

$$\frac{\text{cutcost}(C_p)}{\min\{\text{s\_degree}(C_p), \text{s\_degree}(V-C_p)\}}$$

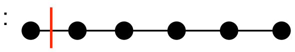
Optimization using conductance

- Choices:
  - minimize $\sum^{k}_{p=1}$ conductance (C_p)
  - minimize $\text{MAX}^{k}_{p=1}$ conductance (C_p)
- Observations
  - conductance (C_p) = conductance $(V\text{-}C_p)$
  - Finding a cut (C, V-C) with minimum conductance is NP-hard

13

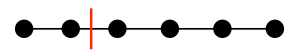
Simple example

- six objects
- similarity 1 if edge shown
- similarity 0 otherwise
- choice 1:



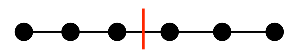
conductance 1/min(1,9) = 1

- choice 2:



conductance 1/min(3, 7) = 1/3

- choice 3:



conductance 1/min(5, 5) = 1/5 *prefer balance

14

Hierarchical divisive revisited

- can use one of cut-based algorithms to split a cluster
- how choose cluster to split next?
  - if building entire tree, doesn't matter
  - if stopping a certain point, choose next cluster based on measure optimizing
    - e.g. for total relative cut cost, choose C_i with largest cutcost(C_i) / intracost(C_i)

15

Divisive Algorithm:
Iterative Improvement; no hierarchy

1. Choose initial partition $C_1, \ldots, C_k$
2. repeat {
   unlock all vertices
   repeat {
      choose some C_i at random
      choose an unlocked vertex v_j in C_i
      move v_j to that cluster, if any, such that move gives maximum decrease in cost
      lock vertex v_j
   } until all vertices locked
}until converge

16

Observations on algorithm

- heuristic
- uses randomness
- convergence usually improvement < some chosen threshold between outer loop iterations
- vertex “locking” insures that all vertices are examined before examining any vertex twice
- there are many variations of algorithm
- can use at each division of hierarchical divisive algorithm with k=2
  - more computation than an agglomerative merge

17

Compare to k-means

- Similarities:
  - number of clusters, k, is chosen in advance
  - an initial clustering is chosen (possibly at random)
  - iterative improvement is used to improve clustering
- Important difference:
  - divisive algorithm can minimize a cut-based cost
    - total relative cut cost, conductance use external and internal measures
  - k-means maximizes only similarity within a cluster
    - ignores cost of cuts

18

Eigenvalues and clustering

General class of techniques for clustering a graph using eigenvectors of adjacency matrix (or similar matrix) called

Spectral clustering

First described in 1973

spectrum of a graph is list of eigenvalues, with multiplicity, of its adjacency matrix

19

Spectral clustering: *brief* overview

Given: k: number of clusters
nxn object-object sim. matrix S of non-neg. val.s

Compute:

1. Derive matrix L from S (straightforward computation)
   - variety of definitions of L
     - e.g. Laplacian L=I-E if similarity is edge/no edge
2. find eigenvectors corresp. to k smallest eigenval.s of L
3. use eigenvectors to define clusters
   - variety of ways to do this
   - all involve another, simpler, clustering
     - e.g. points on a line

Spectral clustering optimizes a cut measure
similar to total relative cut cost

20

HITS and clustering

Recall HITS matrix formulation:

$\boldsymbol{a} = E^T\boldsymbol{h}$ $\quad\quad$ $\boldsymbol{a} = E^TE\boldsymbol{a}$

$\boldsymbol{h} = E\boldsymbol{a}$ $\quad\quad$ $\boldsymbol{h} = EE^T\boldsymbol{h}$

for adjacency matrix E, authority vector **a**, hub vector **h**

- ***a*** is the eigenvector corresponding to the eigenvalue 1 for E^TE
- ***h*** is the eigenvector corresponding to the eigenvalue 1 for EE^T

HITS and clustering

- Non-principal eigenvectors of EE^T and E^TE have positive and negative component values
  - Denote $a_{e2}, a_{e3}, \dots$ matching $h_{e2}, h_{e3}, \dots$
- For a matched pair of eigenvectors $\boldsymbol{a}_{ej}$ and $\boldsymbol{h}_{ej}$
  - Denote k^{th} component of j^{th} pair: $\boldsymbol{a}_{ej}(k)$ and $\boldsymbol{h}_{ej}(k)$
  - Make a "community" of size *c* (chosen constant):
    - Choose c pages with most positive $\boldsymbol{h}_{ej}(k)$ - hubs
    - Choose c pages with most positive $\boldsymbol{a}_{ej}(k)$ - authorities
  - Make another "community" of size *c*:
    - Choose c pages with most negative $\boldsymbol{h}_{ej}(k)$ - hubs
    - Choose c pages with most negative $\boldsymbol{a}_{ej}(k)$ - authorities

Comparing clusterings

- Define external measure to
  - comparing two clusterings as to similarity
  - if one clustering "correct", one clustering by an algorithm, measures how well algorithm doing
    - refer to "correct" clusters as classes
      - "gold standard"
    - refer to computed clusters as clusters
- External measure independent of cost function optimized by algorithm

One measure: motivated by F-score in IR

- Given:
  - a set of classes $S_1, \dots S_k$ of the objects use to define relevance
  - a computed clustering $C_1, \dots C_k$ of the objects use to define retrieval
- Consider pairs of objects
  - pair in same class, call ***similar pair*** ≡ relevant
  - pair in different classes ≡ irrelevant
  - pair in same clusters ≡ retrieved
  - pair in different clusters ≡ not retrieved
- Use to define precision and recall

Clustering f-score

precision of the clustering w.r.t the gold standard =

$$\frac{\text{\# similar pairs in the same cluster}}{\text{\# pairs in the same cluster}}$$

recall of the clustering w.r.t the gold standard =

$$\frac{\text{\# similar pairs in the same cluster}}{\text{\# similar pairs}}$$

f-score of the clustering w.r.t the gold standard =

$$\frac{2\text{*precision*recall}}{\text{precision + recall}}$$

25

Properties of cluster F-score

- always ≤ 1
- Perfect match computed clusters to classes gives F-score = 1
- Symmetric
  - Two clusterings $\{C_i\}$ and $\{K_j\}$, neither "gold standard"
  - treat $\{C_i\}$ as if are classes and compute F-score of $\{K_j\}$ w.r.t. $\{C_i\}$ = F-score$_{\{Ci\}}(\{K_j\})$
  - treat $\{K_j\}$ as if are classes and compute F-score of $\{C_i\}$ w.r.t. $\{K_j\}$ = F-score$_{\{Kj\}}(\{C_i\})$
  - ➢ F-score$_{\{Ci\}}(\{K_j\})$ = F-score$_{\{Kj\}}(\{C_i\})$

26

another related external measure Rand index

$$\frac{\text{(\# similar pairs in the same cluster + \# dissimilar pairs in the different clusters)}}{N\,(N\text{-}1)/2}$$

percentage pairs that are correct

27

Clustering: wrap-up

- many applications
  - application determines similarity between objects
- menu of
  - cost functions to optimizes
  - similarity measures between clusters
  - types of algorithms
    - flat/hierarchical
    - constructive/iterative
  - algorithms within a type

28