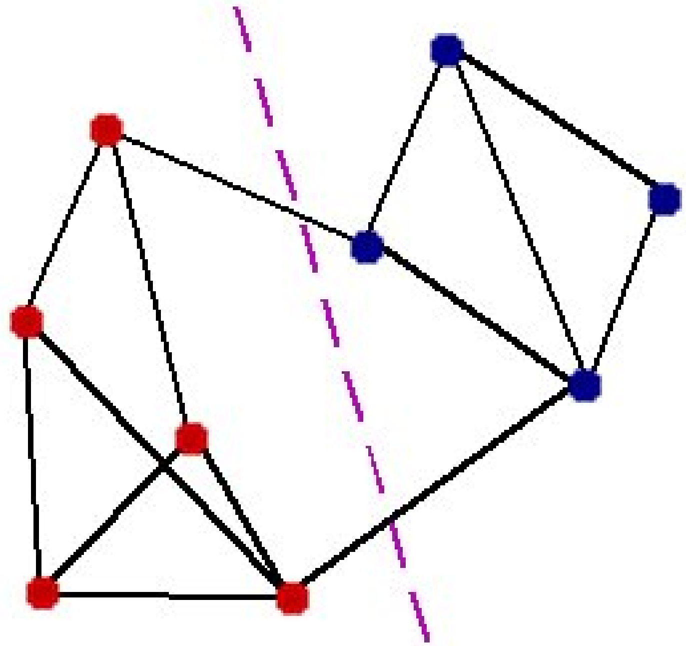
Data Clustering via Graph Min-Cuts



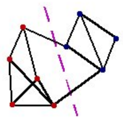
Erin R. Rhode
University of Michigan
EECS 767
March 15, 2006

Outline

- Why Data Clustering?

- Introduction to Graph Min-Cuts
  - Bipartite Graph Partitioning (Zha et. al., 2001)
  - Minimum-Cut Tree Partitioning (Flake et. al., 2002)

- Application as a Learning Algorithm (Blum & Chawla, 2001)
  - Use in Sentiment Analysis: Movie Reviews (Pang & Lee, 2004)



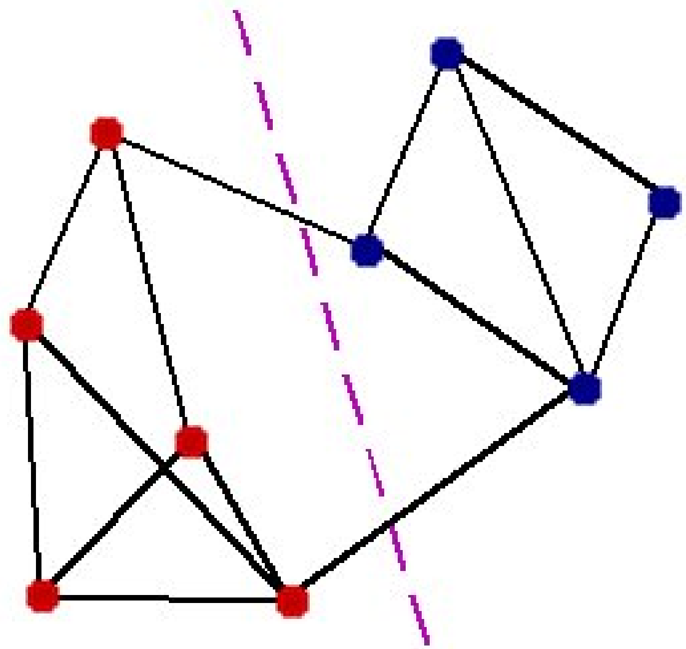
Data Clustering

- ⋆ Document Clustering
  - ⋆ Grouped by topic

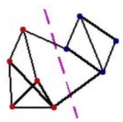
- ⋆ Sentiment Analysis: Polarity Clustering
  - ⋆ Positive vs. Negative Movie Reviews

Basic Idea: Think of data as a graph where edges relate to similarity – Cluster data by finding a minimum cut on the graph

What's a Graph Min-Cut?



- Minimum edge cut that separates graph into two clusters

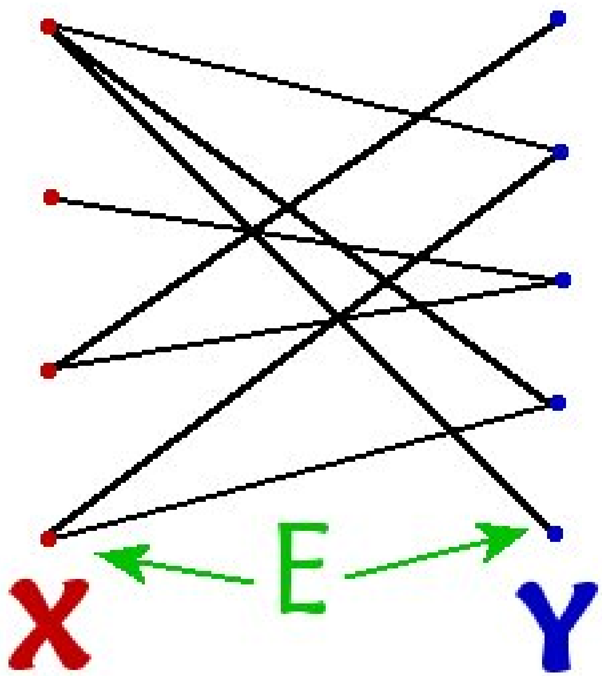


Weighted Bipartite Graphs – Formally

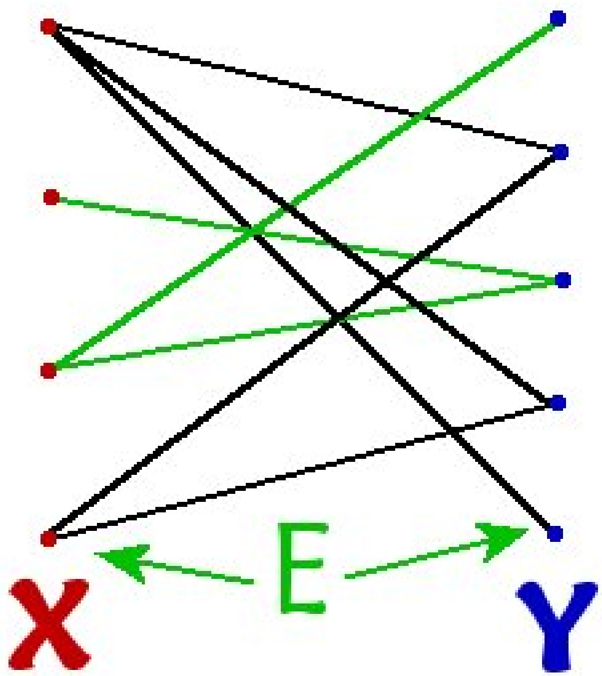
- For a graph $G(V, E, W)$
  - V = Vertices of G, E = Edges of G, W = Edge weights

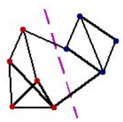
- Two Vertex classes X, Y s.t.:
  - $V = X \cup Y$
  - $\emptyset = X \cap Y$
  - Each edge $e \in E$ has one endpoint $v_x \in X$ and one endpoint $v_y \in Y$

Bipartite Graphs



Bipartite Graphs



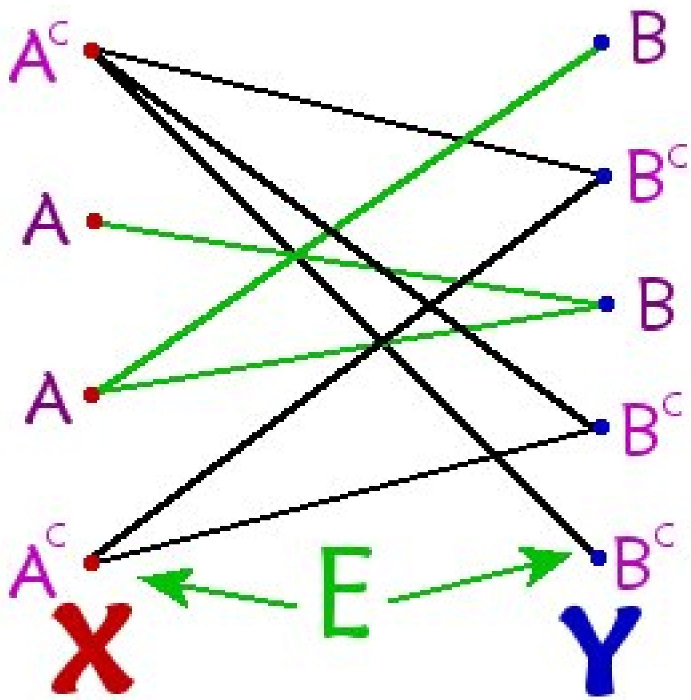


Partitions in Bipartite Graphs (Zha et.al.,2001)

- For a bipartite graph G(X,Y,W) (Note change in syntax), Vertex Partition $\prod(A,B)$:
  - $X = A \cup A^c$
  - $Y = B \cup B^c$
- Cut: Remove edges between A and B^c and the edges between A^c and B

$$\textbf{Association: } W(A,B) = \sum_{i \in A, j \in B} w_{ij}$$

Bipartite Graphs



Min-Cuts on Bipartite Graphs

- Given a partition $\prod(A,B)$, possible metric for "cut" to minimize:

$$\text{cut}(A,B) = W(A,B^c) + W(A^c,B)$$

- **Pro:** Minimizes the association between compliments

- **Con:** Note that if $A = X$, $B = Y$, $A^c = B^c = \emptyset$, then $\text{cut}(A,B) = 0$

- Right idea – needs tweaking!

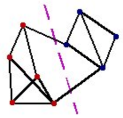
Min-Cuts on Bipartite Graphs

- Need to use a normalized metric that factors in partition size:

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{W(A, Y) + W(X, B)} + \frac{\text{cut}(A^c, B^c)}{W(A^c, Y) + W(X, B^c)}$$

- Minimize Ncut using spectral methods – Spectral Recursive Embedding (SRE)
  - See last week's presentation

- Good method for document clustering



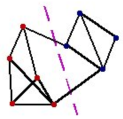
Partitions with Overlaps

- ⋆ What if a vertex belongs to more than one cluster?
  - ⋆ Basic idea: Allow for an "overlap" set, (O_X, O_Y)

    $$X = A \cup O_X \cup \bar{A} \text{ and } Y = B \cup O_Y \cup \bar{B}$$

  - ⋆ Disregard overlap in Ncut:

    $$\text{Ncut}(A, B, \bar{A}, \bar{B}) = \frac{\text{cut}(A, B)}{W(A, Y) + W(X, B)} + \frac{\text{cut}(\bar{A}, \bar{B})}{W(\bar{A}, Y) + W(X, \bar{B})}$$



Partitions with Overlaps

- ⋆ Problem: Can put everything in the overlap and Ncut = 0

- ⋆ Solution: Account for size of O_X and O_Y:

$$\text{Ncut}(A, B, \bar{A}, \bar{B}) + \alpha(|O_X| + |O_Y|)$$

  - ⋆ α = Regularization Parameter

- ⋆ Open Questions: How to determine α? How to minimize new Ncut?

Bipartite Experiment

- ⋆ Clustering newsgroup articles from 20 different newsgroups
- ⋆ Use pairs of newsgroups with uneven clusters
  - ⋆ **`alt.atheism`** vs. **`comp.graphics`**
  - ⋆ **`rec.sport.baseball`** vs. **`rec.sport.hockey`**
  - ⋆ **`talk.politics.mideast`** vs. **`talk.politics.misc`**
- ⋆ Compare against two other clustering methods: K-means and Principal direction Divisive Partition (PDDP)

Bipartite experiment – Results

⋆ `alt.atheism` vs. `comp.graphics`

Mixture	SRE	PDDP	K-means
50/200	$82.77 \pm 5.24\%$	$70.43 \pm 6.04\%$	$69.22 \pm 12.34\%$

⋆ `rec.sport.baseball` vs. `rec.sport.hockey`

Mixture	SRE	PDDP	K-means
50/200	$57.55 \pm 5.69\%$	$56.63 \pm 4.84\%$	$60.82 \pm 7.54\%$

⋆ `talk.politics.mideast` vs. `talk.politics.misc`

Mixture	SRE	PDDP	K-means
50/200	$61.23 \pm 9.88\%$	$60.76 \pm 5.55\%$	$64.50 \pm 7.58\%$

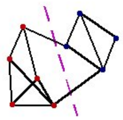
Network Flow Approach

(Flake & Tarjan & Tsioutsiouliklis, 2002)

- Completely different approach to the problem – doesn't require bipartite graph

- For weighted $G(V, E)$, $s \in V$ = source node, $t \in V$ = sink node

- **Community of s:** $S \subset V$
  - $s \in S, t \notin S$
  - Cut between S and $V - S$ is minimal



Minimum-Cut Trees

- ⋆ Also known as Gomory-Hu Cut Trees (Gomory & Hu, 1961)

- ⋆ For $G(V, E)$, T_G is a “cut equivalent” tree to G defined over V

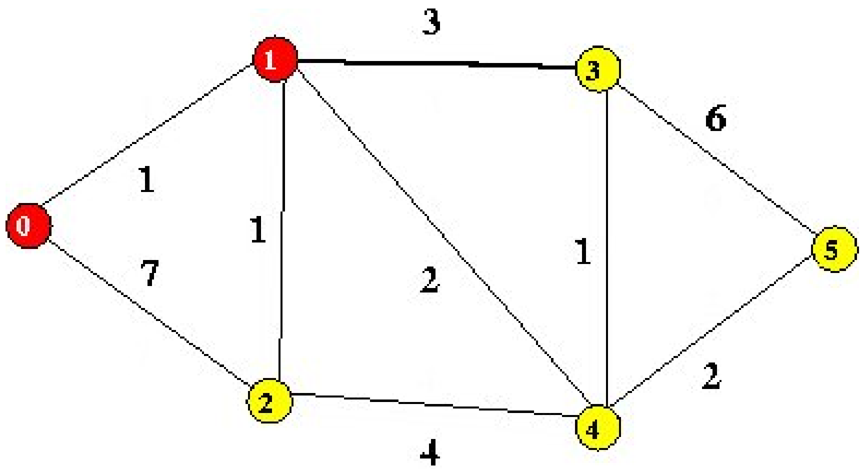
  - ⋆ Construct by calculating $|V| - 1$ min-cuts

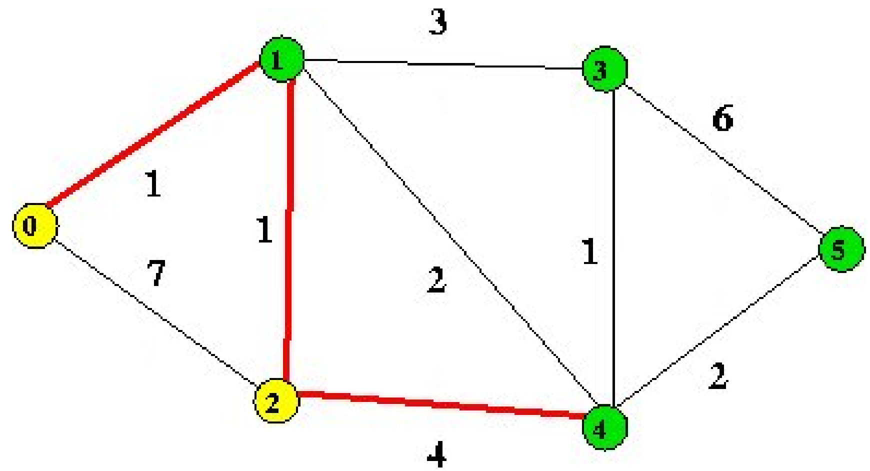
  - ⋆ Recursively find min-cuts between pairs on graph

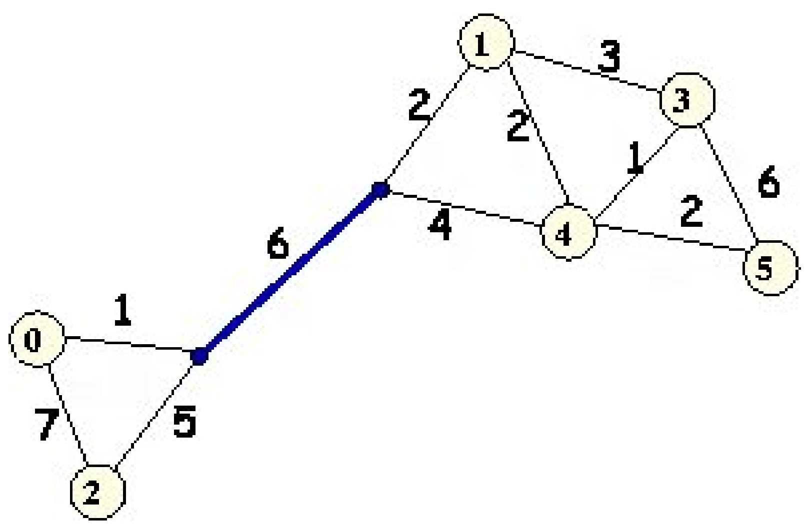
    - ⋆ Closely related to the max-flow problem

    - ⋆ Min-cut value is weight on tree edge

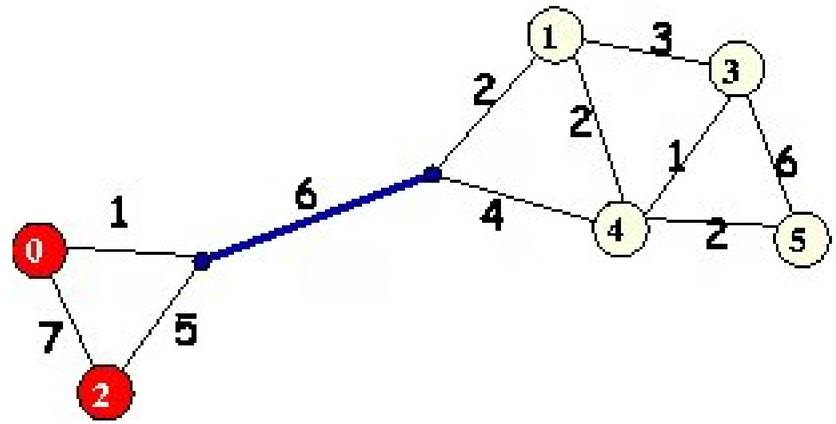
Min-Cut Tree Example

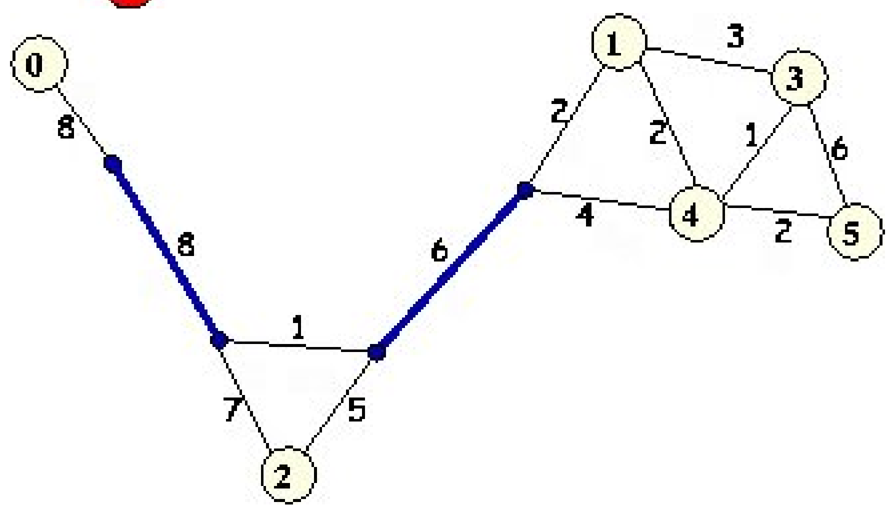




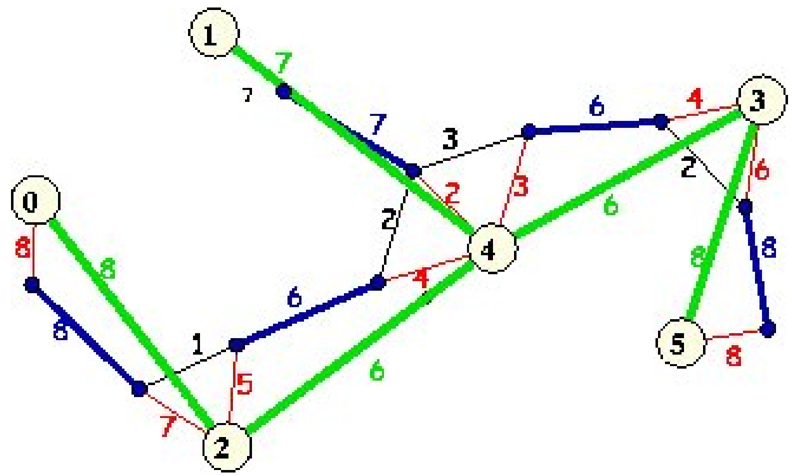


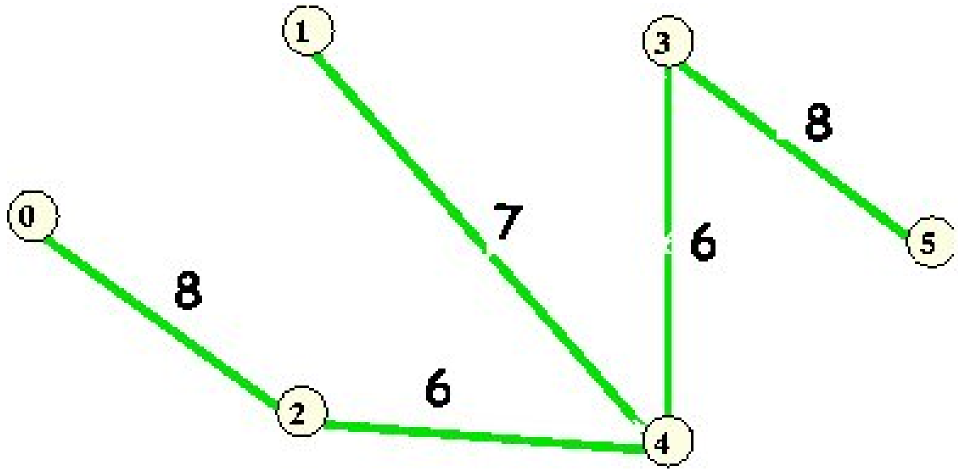
Min-Cut Tree Example





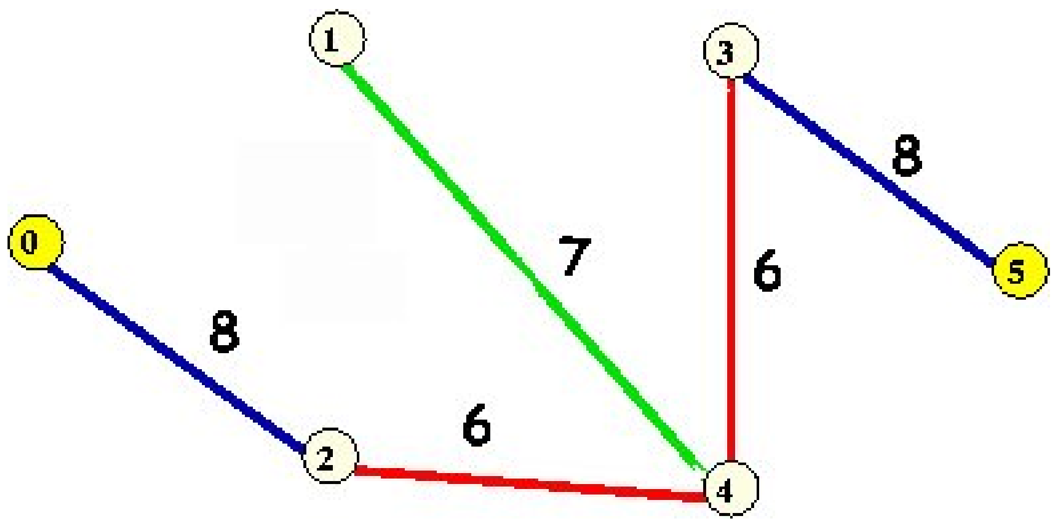
Min-Cut Tree Example





Determining Min-Cut

- Determine min-cut between s and t by examining path in T_G
  - Edge on path with minimum capacity $\rightarrow$ minimum cut



What's a "Good" Cut?

- ⋆ Expansion of a Cut:

$$\psi(S) = \frac{\sum_{i \in S, j \in \bar{S}} w_{ij}}{\min\{|S|, |\bar{S}|\}}$$

- ⋆ Expansion of a clustering: Minimum expansion over all clusters

- ⋆ Bigger the expansion of a clustering, higher its quality

- ⋆ Problem: NP-hard calculation

Artificial sinks

- What if you want to find a community around s, but you don't care what the "sink node" is?

- Add new node, t: "Artificial Sink"

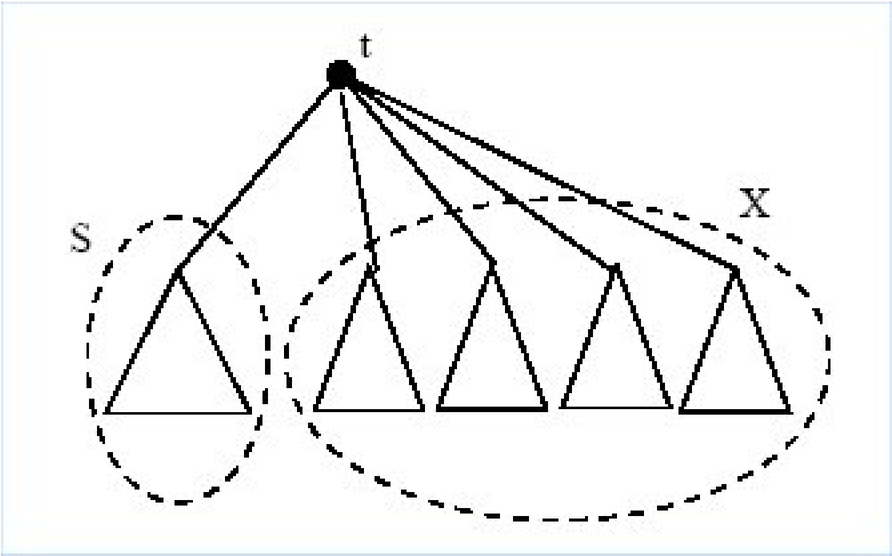
  - Connect t to every node in V with edge weight α

  - Find S with respect to t

  - Expansion of S is lower bounded by α

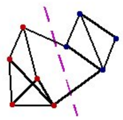
- Basis of Cut and Cluster Algorithm

Cut and Cluster Algorithm



Cut and Cluster Algorithm

- Run the basic "Cut and Cluster" algorithm
  - Add artificial sink using α as edge weight
  - Compute min-cut tree
  - Remove artificial sink
  - Return connected components as clusters
- Recursively try smaller values of α



Application as a Learning Algorithm

(Blum & Chawla, 2001)

- ⋆ Can apply network flow approach to classification problem
- ⋆ Classification vertices, v_+ and v_-, serve as source and sink
- ⋆ Find communities of v_+ and v_- using min-cut algorithms
- ⋆ Does not require large amounts of labeled data

Application as a Learning Algorithm

- Example vertices: Labeled (L_+, L_-) and unlabeled (U) data
  - $w(v, v_+) = \infty$ for all $v \in L_+$
  - $w(v, v_-) = \infty$ for all $v \in L_-$
- Weights between example vertices based on some notion of similarity defined for the dataset

Edges Between Example Vertices

- **Mincut-3** Edge weight one between 3 nearest neighbors
  - Require one of the neighbors to be a labeled example

- **Mincut-δ** Connect vertices with similarity greater than δ
  - **Mincut-δ_0** Choose max δ for which graph has cut of 0
  - **Mincut-$\delta_{1/2}$** Choose δ such that the largest connected component is half the number of datapoints
  - **Mincut-δ_{opt}** Choose δ with least classification error *in hindsight* (Used as a benchmark)

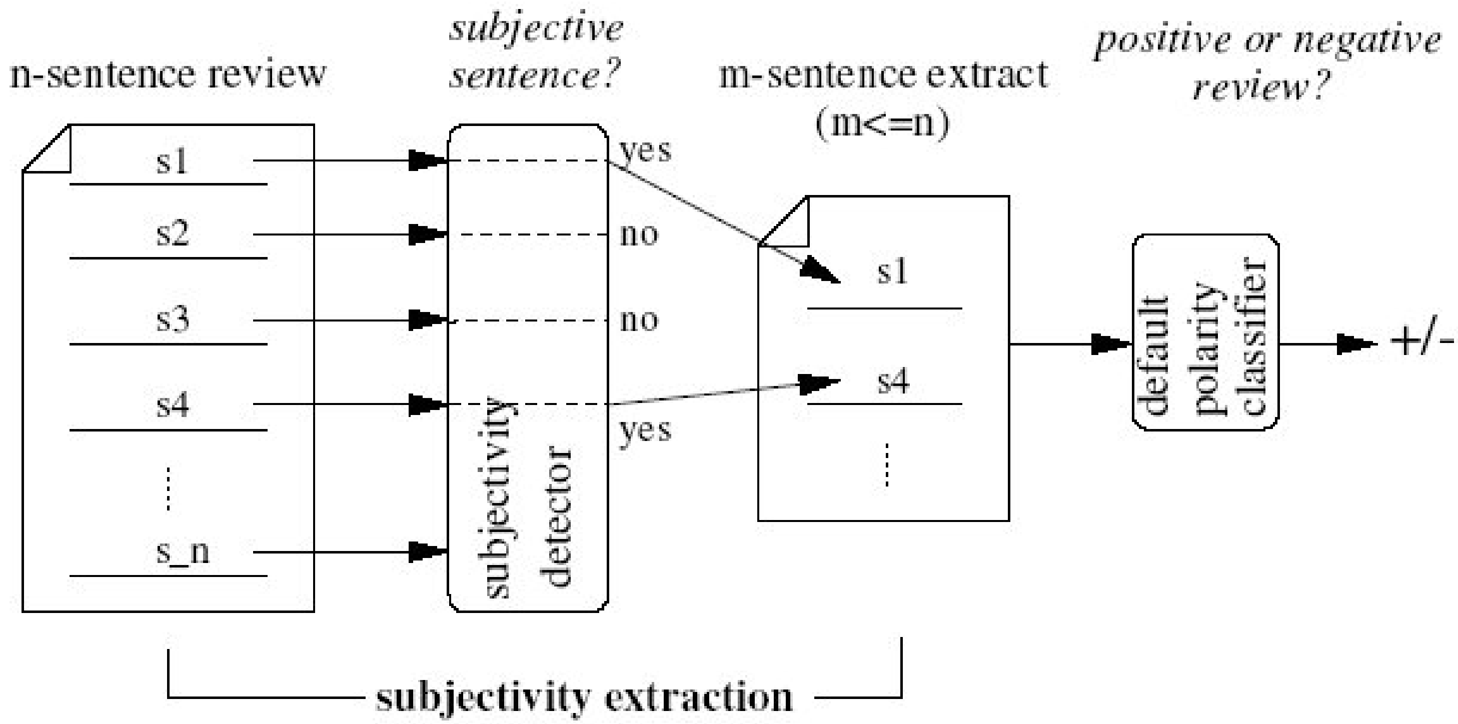
Classification Results

Dataset	$\|L\|\&\|U\|$	Mincut-3	Mincut-δ_{opt}	Mincut-δ_0	Mincut-$\delta_{1/2}$	3-NN
Mush	20+1000	82.1	**97.7**	**97.7**	97.0	91.1
Mush*	20+1000	74.2	**88.7**	56.9	**87.0**	83.3
Tae	10+100	86.0	**99.0**	96.0	**97.0**	80.0
Voting	45+390	89.1	**91.3**	66.1	83.3	**89.6**
Musk	40+200	73.0	**92.5**	91.0	**92.5**	87.0

- ⋆ Mush* includes noise
- ⋆ Mincut-$\delta_{1/2}$ tends to perform the best
- ⋆ Mincut-δ_0 does well in absence of noise

Sentiment Analysis: Movie Reviews

(Pang & Lee, 2004)



Movie Reviews: Experimental Design

- ⋆ Classify sentences as subjective or objective
  - ⋆ Subjective: ”This is a good movie.”
  - ⋆ Object: ”The protagonist tries to protect her good name.”
- ⋆ Objective sentences may skew subjectivity ratings
- ⋆ Extract subjective sentences using min-cut methods
  - ⋆ Use ”subjective” and ”objective” as source and sink

Movie Reviews: Experimental Design

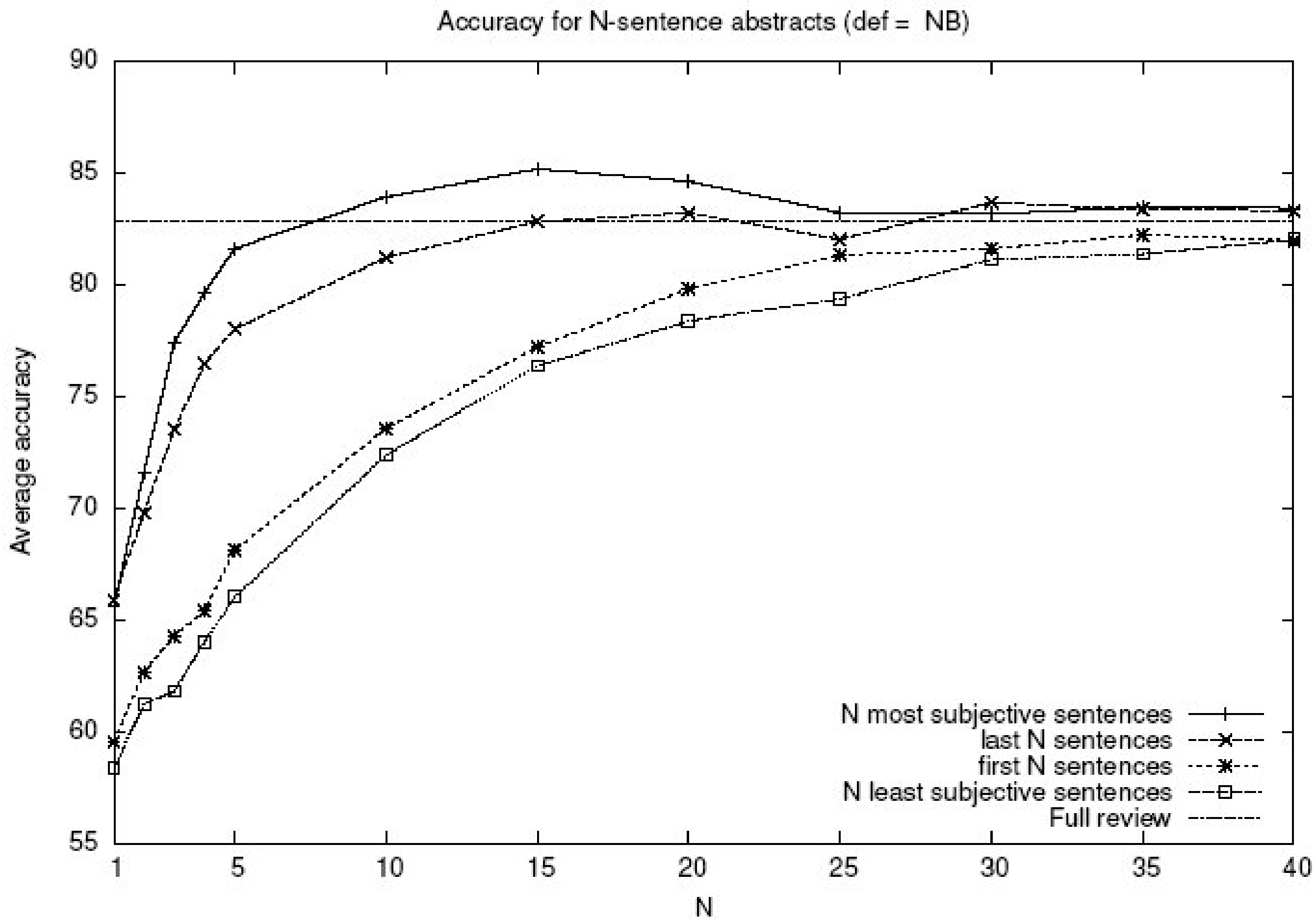
- Polarity (unlabeled) dataset: 1000 positive and 1000 negative reviews

- Labeled subjective data: 5000 movie review snippets from www.rottentomatoes.com

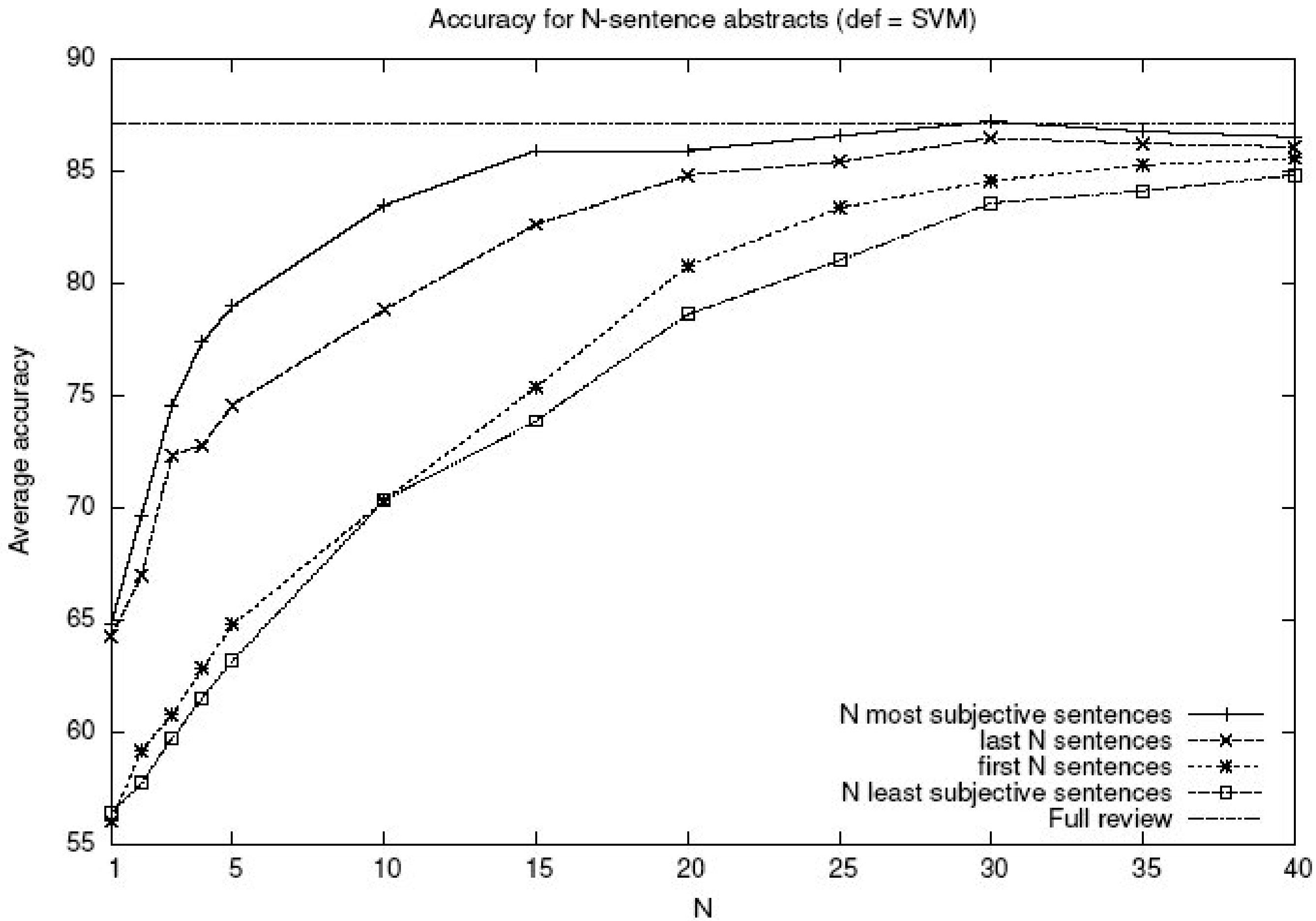
- Labeled objective data: 5000 sentences from IMBD plot summaries

- Use Naive Bayes and Support Vector Machine methods as polarity classifiers

Movie Reviews: Results



Movie Reviews: Results



To Review...

- Use Graph Min-Cuts to Cluster Data
- Two different Approaches:
  - Bipartite Graph Partitioning (Zha et. al., 2001)
  - Minimum-Cut Tree Partitioning (Flake et. al., 2002)
- Can be used as a Learning Algorithm (Blum & Chawla, 2001)
  - Sentiment Analysis: Movie Reviews (Pang & Lee, 2004)