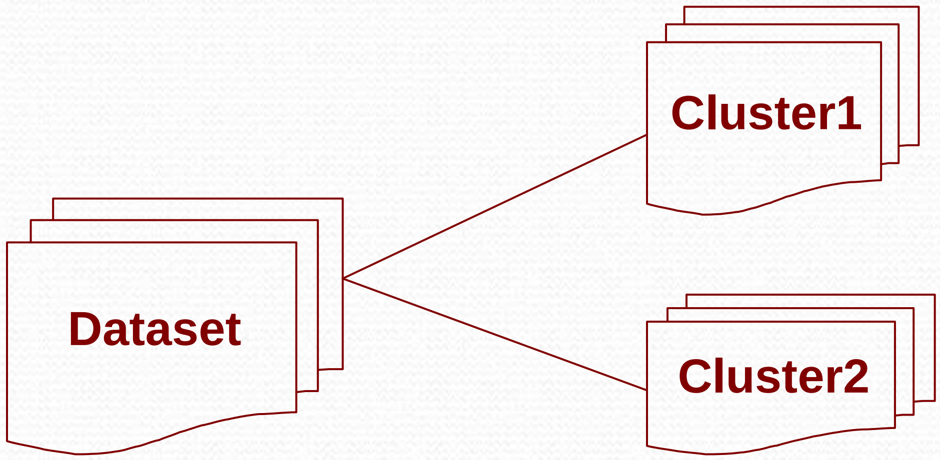
Topic-Wise, Sentiment-Wise, or Otherwise? Identifying the Hidden Dimension for Unsupervised Text Classification

Sajib Dasgupta and Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas

Goal



- **Unsupervised Text Classification**
  - Why unsupervised? Labeled data is expensive

Goal

- Investigate

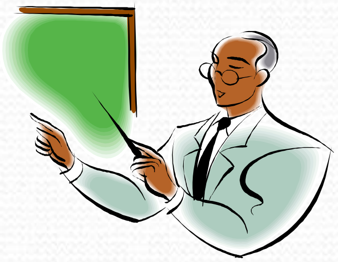
  "Why text clustering fails more often?"

- Possible Reasons:
  - Complexity of the task: Ambiguity
  - Large and complex feature space
  - No labeled data

- **One more reason!**

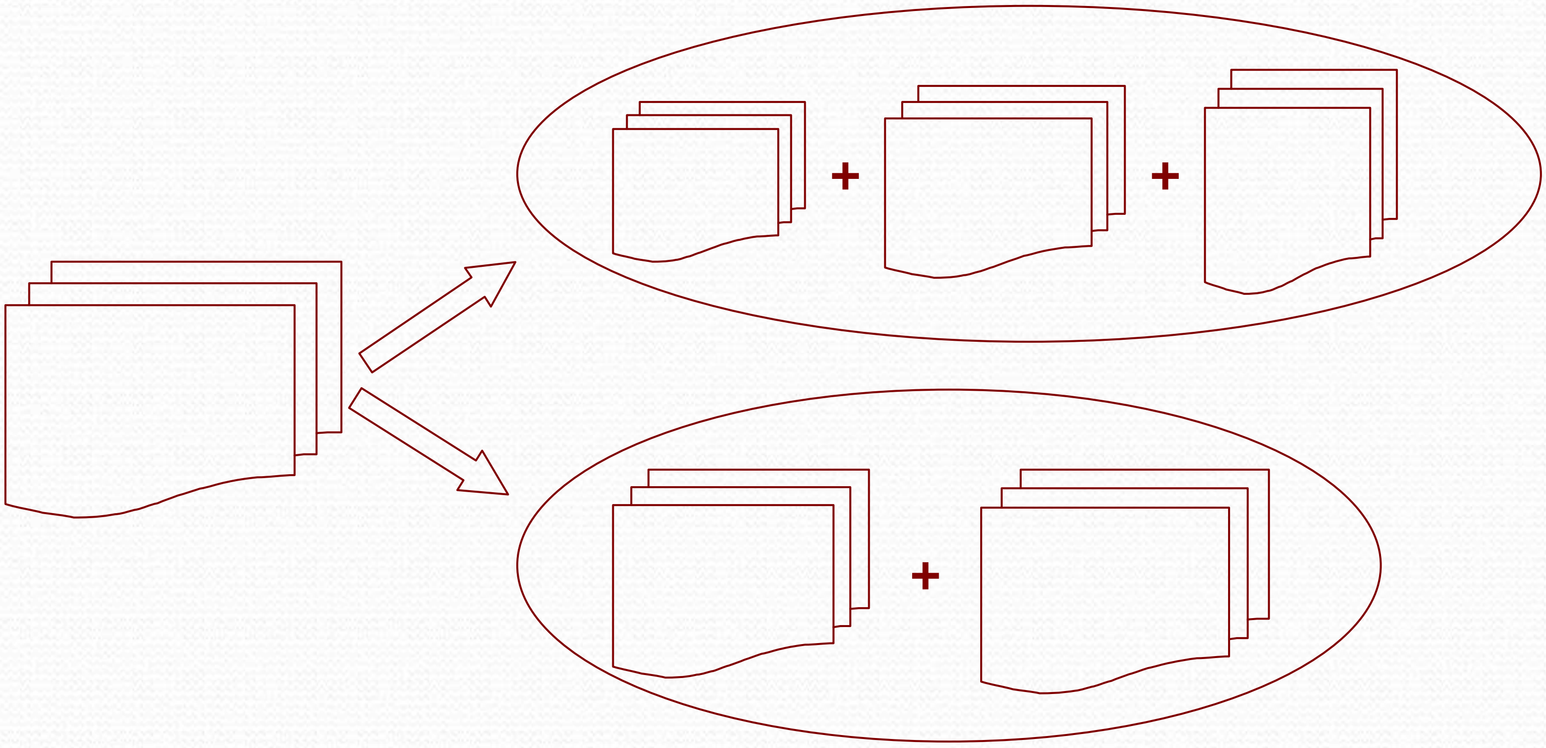
Why Text Clustering Fails?

- **Conflict of interest between machine and human!**



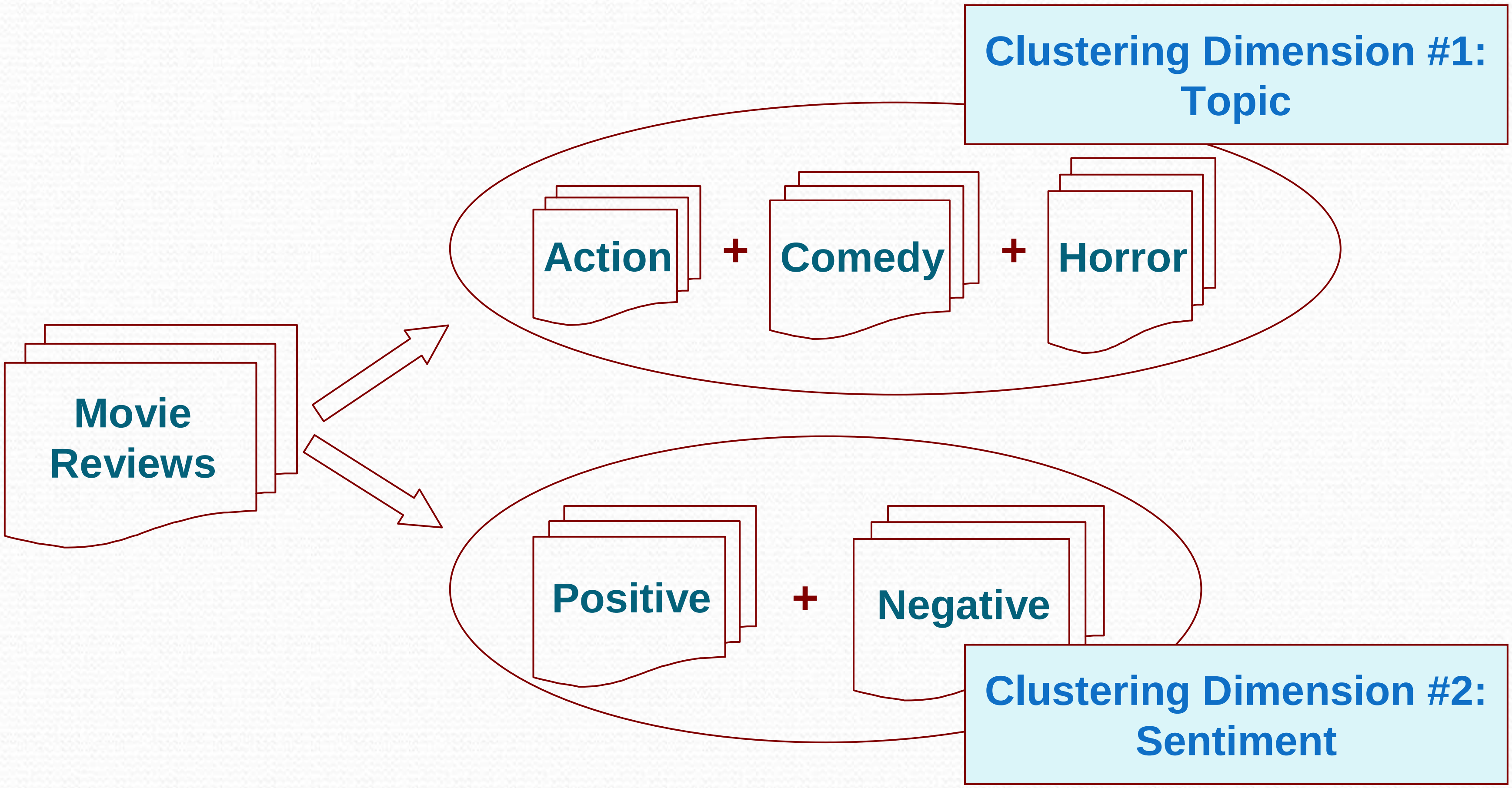
Why Text Clustering Fails?

- **Same data can be clustered different way**



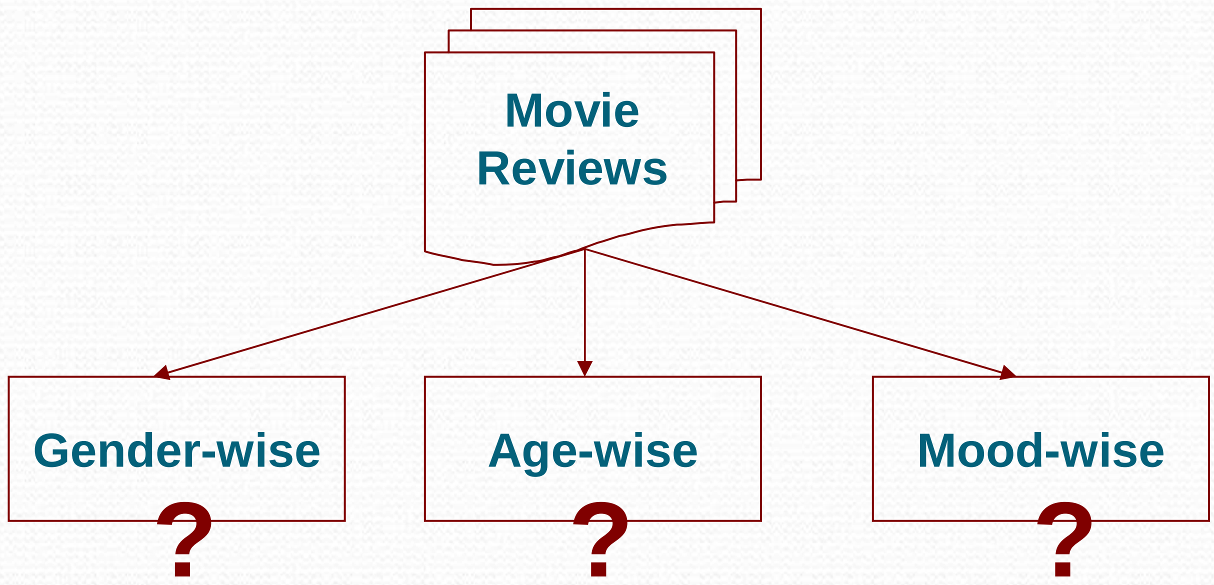
Why Text Clustering Fails?

- **Same data can be clustered different way**



Why Text-Clustering Fails?

- **Same data can be clustered different way**



Why Text-Clustering Fails?

- **What clustering dimension user wants?**
  - Topic-wise?
  - Sentiment-wise?
  - Otherwise?

Why Text Clustering Fails?

- **What clustering-dimension user wants?**
  - Topic-wise?
  - Sentiment-wise?
  - Otherwise?

- **What clustering-dimension clustering algorithm produces?**
  - Topic-wise?
  - Sentiment-wise?
  - Otherwise?

Clustering algorithm fails if it fails to meet user demand

Goal

- **Cluster a set of documents along the user-specified dimension**

Dataset

- **Sentiment Domain:**
  - 2-way clustering
  - Classify a review as “thumbs up” or “thumbs down”

Why sentiment domain?

Why Sentiment Domain?

- **Clustering of sentiment is tough**
- **A lot tougher than topic-based clustering**

Why Clustering of Sentiment is Tough?

- **Sentimental ambiguity**
  - A review may contain both positive and negative words!

> the sound from my system did seem to be a little better (the CD's were not skipping as much). but the bottom line is it didn't fix the problem as the CDs are still skipping noticeably, although not as bad as before. ...

- **A machine is more likely to label it positive.**

Why Clustering of Sentiment is Tough?

- **Sentimental ambiguity**
  - A review may contain both positive and negative words!

> the sound from my system did seem to be a little better (the CD's were not skipping as much). but the bottom line is it didn't fix the problem as the CDs are still skipping noticeably, although not as bad as before. ...

It's the bottom line that determines sentiment!

- **But … it's actually a negative review!**

Why Clustering of Sentiment is Tough?

- **Subjective vs. Objective material**

John Lynch wrote a classic in Spanish-American Revolutions 1808-1826.
He describes all the events that led to the independence of Latin America from Spain.
The book starts in Rio de La Plata and ends in Mexico and Central America.
Curiously one can note a common pattern of highly stratified societies lead by Spanish …….
The reluctance of Spanish Monarchy (and later even of liberals) led to independence ……..
r all of those who are interested in a better understanding of Latin ……… this great book is a must.
nch cleverly combines historical and economic facts about the Hispanic American societies ……….

- **Too much objective material might mislead a machine**
  - if not human!

Why Clustering of Sentiment is Tough?

- **Objective Features --> Topic-Wise Clusters**
- **Subjective Features --> Sentiment-Wise Clusters**

- Too much of objective content might influence clustering
- Clustering algorithm is more likely to produce topic-wise clusters

Goal

- **Given all these complexities,**

  Is it possible to cluster a set of reviews along the sentiment dimension?

Goal

- Is it possible to cluster a set of reviews along the sentiment dimension?

- **Yes**, with the help of
  - **Spectral learning techniques**
  - **A simple user feedback scheme**

Lets start with spectral clustering,

Spectral Clustering: An Overview

- **Algorithm for 2-way clustering (Ng et al. 2002)**
- Given a data matrix *D* (dimension: *n* x *f*),

Spectral Clustering: An Overview

- **Algorithm for 2-way clustering (Ng et al. 2002)**
- Given a data matrix D (dimension: n x f),
  - **Form similarity matrix S=∅(D) (dimension: n x n)**

We used dot product

Spectral Clustering: An Overview

- **Algorithm for 2-way clustering (Ng et al. 2002)**
- Given a data matrix D (dimension: n x f),
  - Form similarity matrix $S=\varnothing(D)$ (dimension: n x n)
  - **Form diagonal matrix G (dimension: n x n), where $G(i,i)$=sum of the i-th row of S**
  - **Form Laplacian matrix $L=G^{-1/2}\, S\, G^{1/2}$**

Spectral Clustering: An Overview

- **Algorithm for 2-way clustering (Ng et al. 2002)**
- Given a data matrix D (dimension: n x f),
  - Form similarity matrix $S=\varnothing(D)$ (dimension: n x n)
  - Form diagonal matrix G (dimension: n x n),

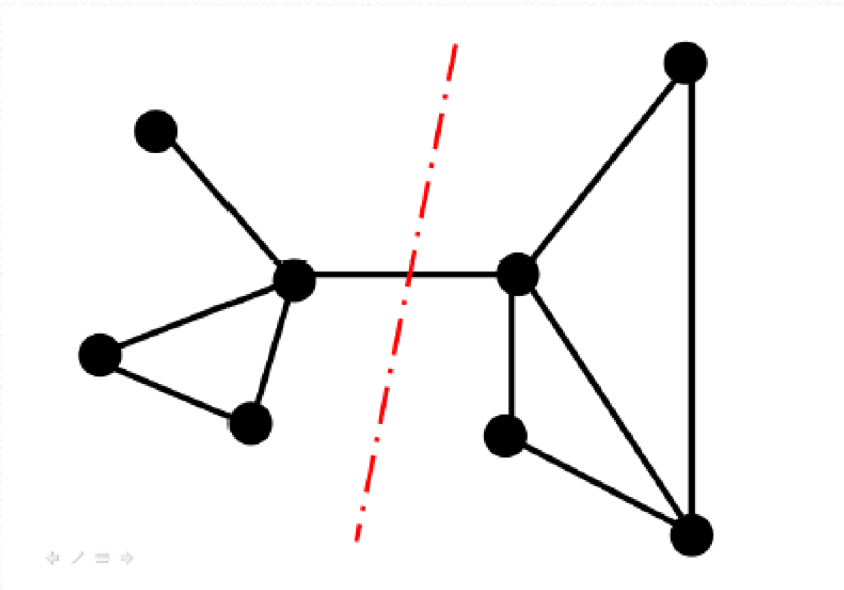
    where $G(i,i)$=sum of the i-th row of S
  - Form Laplacian matrix $L=G^{-1/2}\, S\, G^{1/2}$
  - **Find the eigenvector *e* corresponding to 2^{nd} largest eigenvalue of *L***

Spectral Clustering: An Overview

- **Algorithm for 2-way clustering (Ng et al. 2002)**
- Given a data matrix D (dimension: n x f),
  - Form similarity matrix $S=\varnothing(D)$ (dimension: n x n)
  - Form diagonal matrix G (dimension: n x n),
    where $G(i,i)$=sum of the i-th row of S
  - Form Laplacian matrix $L=G^{-1/2}\,S\,G^{1/2}$
  - Find the eigenvector e corresponding to 2^{nd} largest eigenvalue of L
  - **Use 2-means to cluster *n* data points using *e***

Why 2^{nd} Eigenvector?

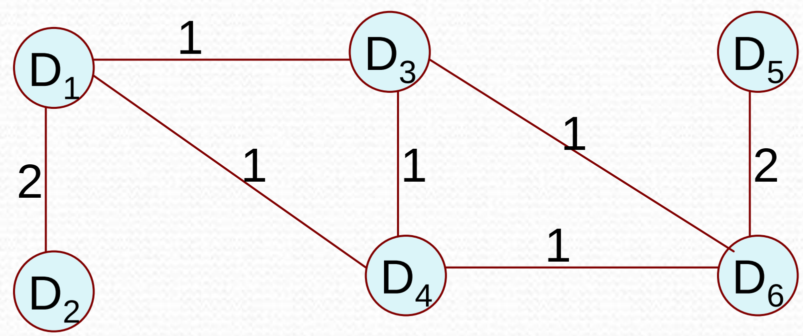
- Shi and Malik (2000): Normalized Cut and Image Segmentation
- 2^{nd} eigenvector of the Laplacian induces the **normalized mincut** of a graph formed from the similarity matrix S



An Example

		-			+	
	f1	f2	f3	f4	f5	f6
D1	1	1	1	0	0	0
D2	1	1	0	0	0	0
D3	0	0	1	0	1	0
D4	0	0	1	1	0	0
D5	0	0	0	1	0	1
D6	0	0	0	1	1	1

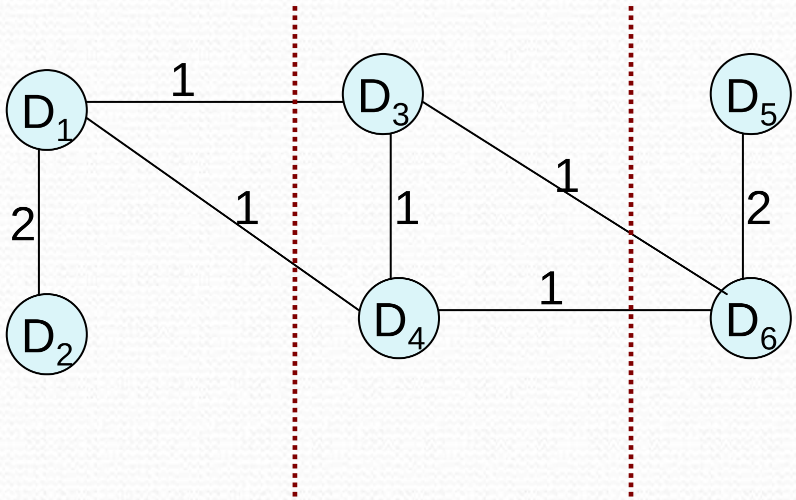
Data Matrix



Similarity Graph

An Example

- Similarity Graph

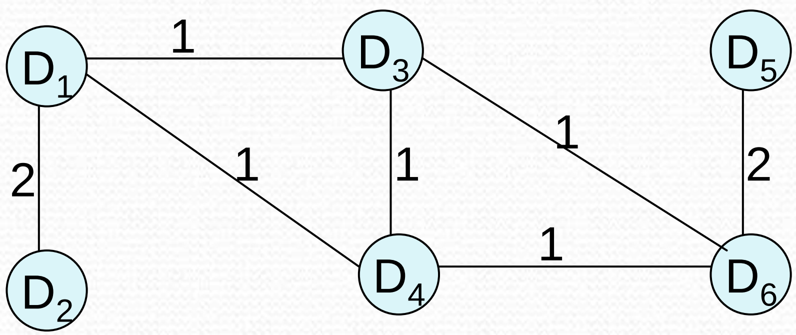


Two possible normalized mincut partitions

Let's see what partition 2nd eigenvector produces

An Example

- Similarity Graph

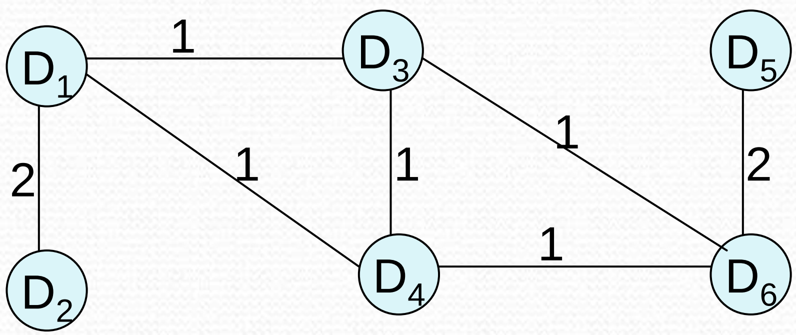


- Top 2 eigenvectors of its Laplacian

$$
\begin{array}{c} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \end{array}
\begin{pmatrix}
0.6362 & -0.7709 \\
0.4161 & -0.7502 \\
0.4838 & 0.0135 \\
0.8082 & 0.2593 \\
0.5130 & 0.5637 \\
0.6973 & 0.6605
\end{pmatrix}
$$

An Example

- Similarity Graph



- Top 2 eigenvectors of its Laplacian

$$\begin{array}{c} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \end{array} \begin{pmatrix} 0.6362 & -0.7709 \\ 0.4161 & -0.7502 \\ 0.4838 & 0.0135 \\ 0.8082 & 0.2593 \\ 0.5130 & 0.5637 \\ 0.6973 & 0.6605 \end{pmatrix} \begin{array}{l} \left.\begin{array}{c} \\ \\ \end{array}\right\} \boxed{-} \\ \left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} \boxed{+} \end{array}$$

2^{nd} Eigenvector

- 2^{nd} eigenvector captures the **most prominent** clustering dimension
- **But does it capture sentiment?**
  - **Sentiment might be hidden**

How to cluster along the desired (possible hidden) dimension?

Our Algorithm

- **We propose a novel feedback-oriented spectral clustering algorithm**
- It allows us to achieve the desired clustering
- It has 4 steps,

Our Algorithm

Steps:

- Identify important dimensions
- Identify unambiguous reviews
- Identify relevant features and get user-feedback
- Cluster along the selected dimension

Our Algorithm

Steps:

- **Identify important dimensions**
- Identify unambiguous reviews
- Identify relevant features and get user-feedback
- Cluster along the selected dimension

Identify Important Dimensions

- Spectral clustering setup:
  - **Features:** Bag of words
  - **Similarity metric:** dot product

Identify Important Dimensions

Given a data matrix *D*,

- We form similarity matrix *S*, diagonal matrix *G*, *and* Laplacian matrix $L=G^{-1/2}\ S\ G^{1/2}$
- Compute the **5 eigenvectors** (with largest eigenvalues) from the Laplacian matrix

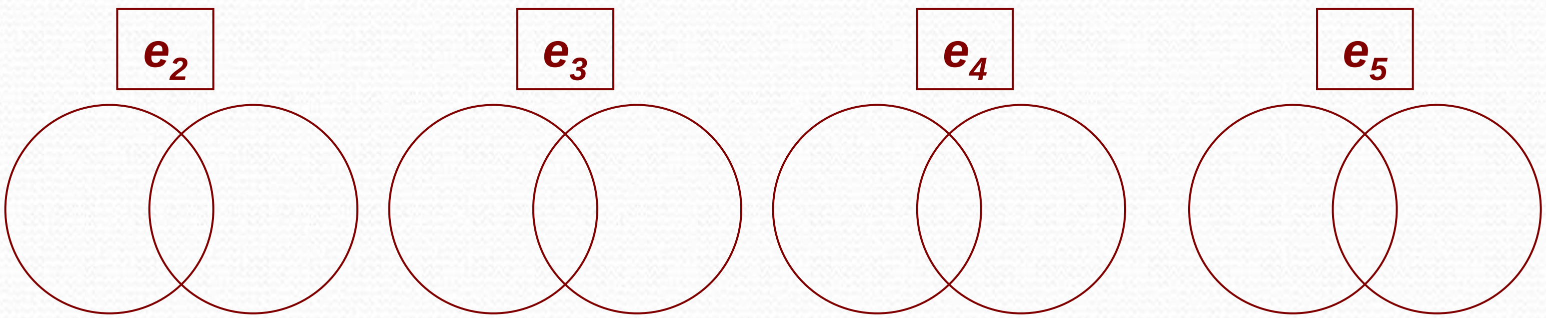
Identify Important Dimensions

Given a data matrix *D*,

- We form similarity matrix *S*, diagonal matrix *G*, *and* Laplacian matrix $L = G^{-1/2} S G^{1/2}$
- Compute the **5 eigenvectors** (with largest eigenvalues) from the Laplacian matrix
  - Each eigenvector (starting from 2nd) captures an important clustering dimension
    - They capture the largest variance in the data
  - First eigenvector is uninformative

Identify Important Dimensions

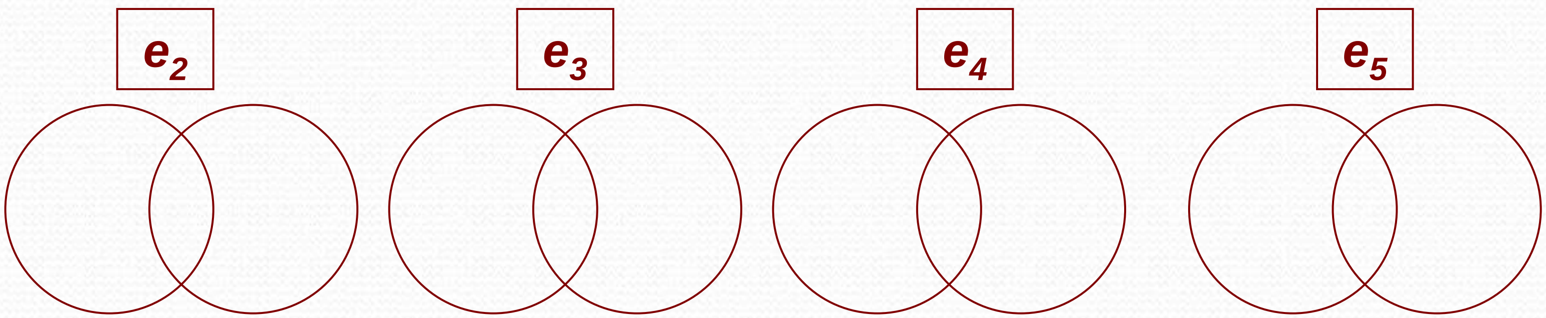
- Each of the 2nd to 5th eigenvector leads to a certain clustering along an important dimension



Which one is sentiment-oriented?

Identify Important Dimensions

- Each of the 2nd to 5th eigenvector leads to a certain clustering along an important dimension



Which one is sentiment-oriented?

Ask human!

Human Feedback

- **We don't give *clusters* to humans to judge!**
  - It's time consuming
  - It will make it a supervised method

- **We give them *features* representative of each cluster**

Human Feedback

- We don't give *clusters* to humans to judge!
  - It's time consuming
  - It will make it a supervised method

- **We give them *features* representative of each cluster**
  - **It's simple, as simple as a cursory look!**
  - At least, it's simple to 5 graduate students

- Before we proceed to human-feedback step,

We have one other important step

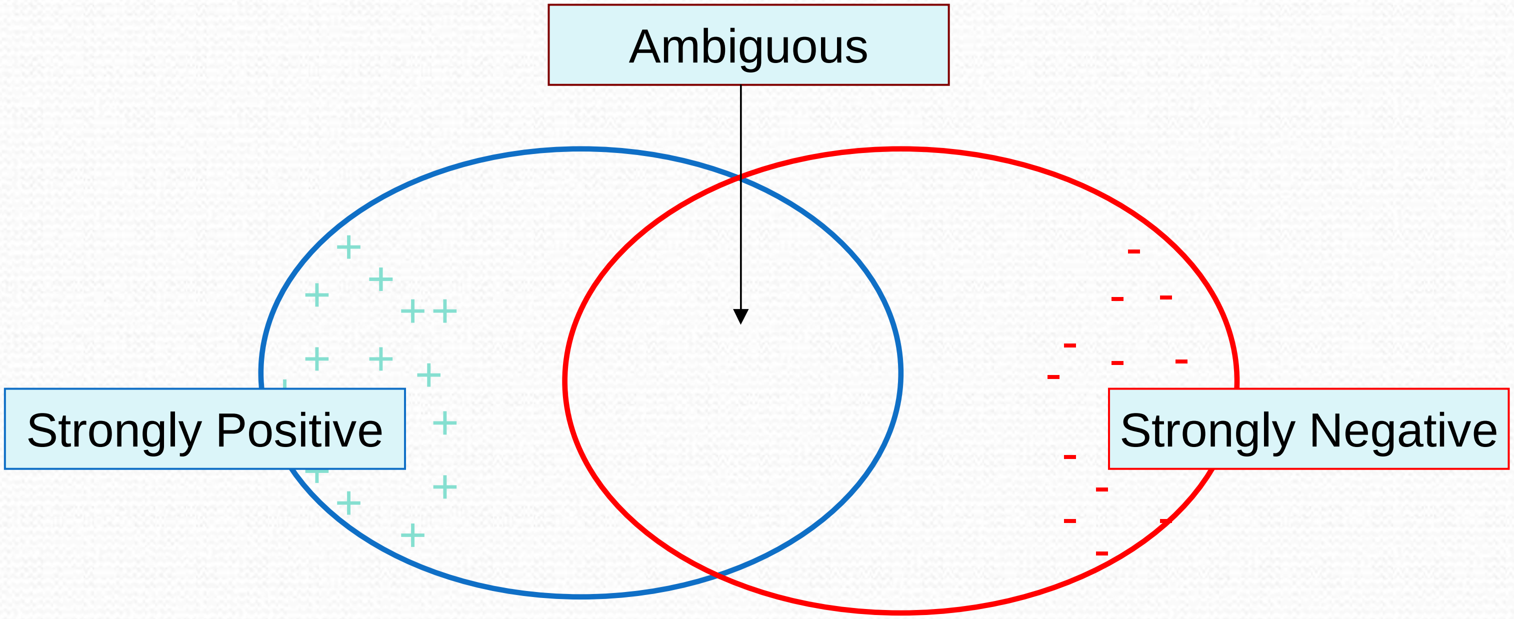
Our Algorithm

Steps:

- Identify important dimensions
- **Identify unambiguous reviews**
- Identify relevant features and get user-feedback
- Cluster along the selected dimension

Identify Unambiguous Reviews

- For each partition corresponding to $\mathbf{e}_2$ to $\mathbf{e}_5$,
- **We separate out unambiguous reviews**
  - in an unsupervised manner

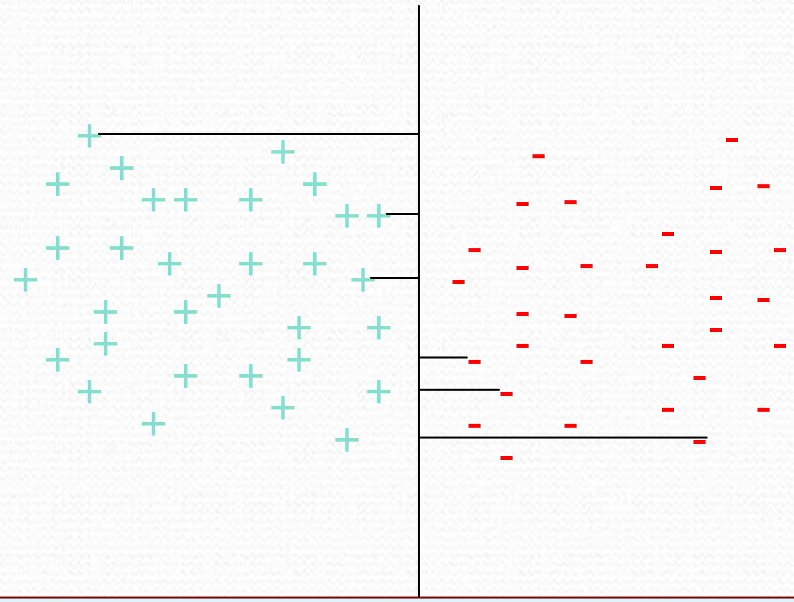


Identify Unambiguous Reviews

- **Why separate unambiguous reviews?**
  - Feature distribution learned from only unambiguous portion is more reliable
  - **Ambiguous reviews can only mislead!**

Identify Unambiguous Reviews

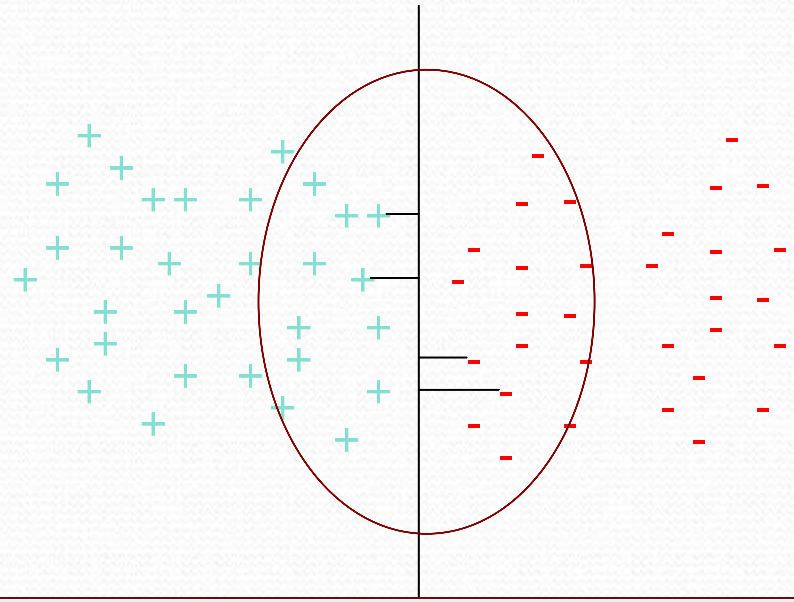
- How to separate unambiguous reviews?



Orthogonal projections on a learned-dimension

Identify Unambiguous Reviews

- How to separate unambiguous reviews?



Ambiguous points have small projections

Identify Unambiguous Reviews

Our Algorithm

- **For each of the 2nd to 5th eigenvector *e*,**
  - we sort the data points according to *e* components
  - keep only top $\alpha/2$ and bottom $\alpha/2$ points
  - create two clusters U_1 and U_2 with top $\alpha/2$ and bottom $\alpha/2$ points respectively

Identify Unambiguous Reviews

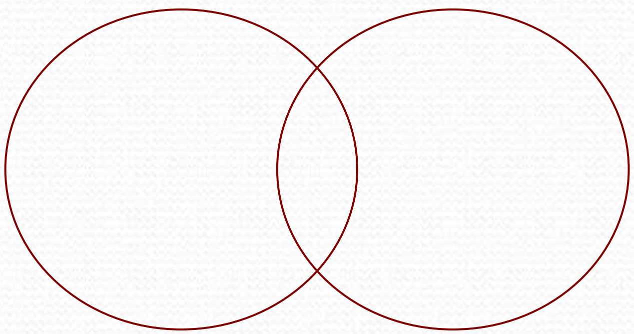
Our Algorithm

- **For each of the 2nd to 5th eigenvector *e*,**
  - we sort the data points according to e components
  - keep only top $\alpha/2$ and bottom $\alpha/2$ points
  - create two clusters U_1 and U_2 with top $\alpha/2$ and bottom $\alpha/2$ points respectively

  - **We use $\alpha = n/4$**
    - Assuming that at least 25% reviews are unambiguous

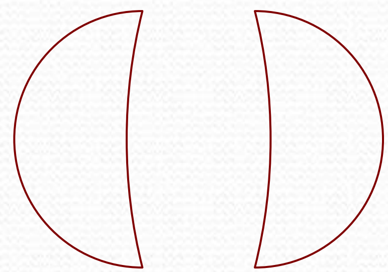
Clustering Accuracy

All data points



Clustering Accuracy: Low

Unambiguous data points



Clustering Accuracy: High

Clustering accuracy of unambiguous reviews

- Accuracy of 500 unambiguous reviews out of 2000.

	Clustering Accuracy (%)
Movie	**87.0**
Kitchen	**87.6**
Electronics	**77.6**
Books	**86.2**
DVD	**87.4**

Clustering accuracy of unambiguous reviews

- Accuracy of 500 unambiguous reviews out of 2000.

	Clustering Accuracy (%)
Movie	**87.0**
Kitchen	**87.6**
Electronics	**77.6**
Books	**86.2**
DVD	**87.4**

High-accuracy is essential to learn reliable feature distribution

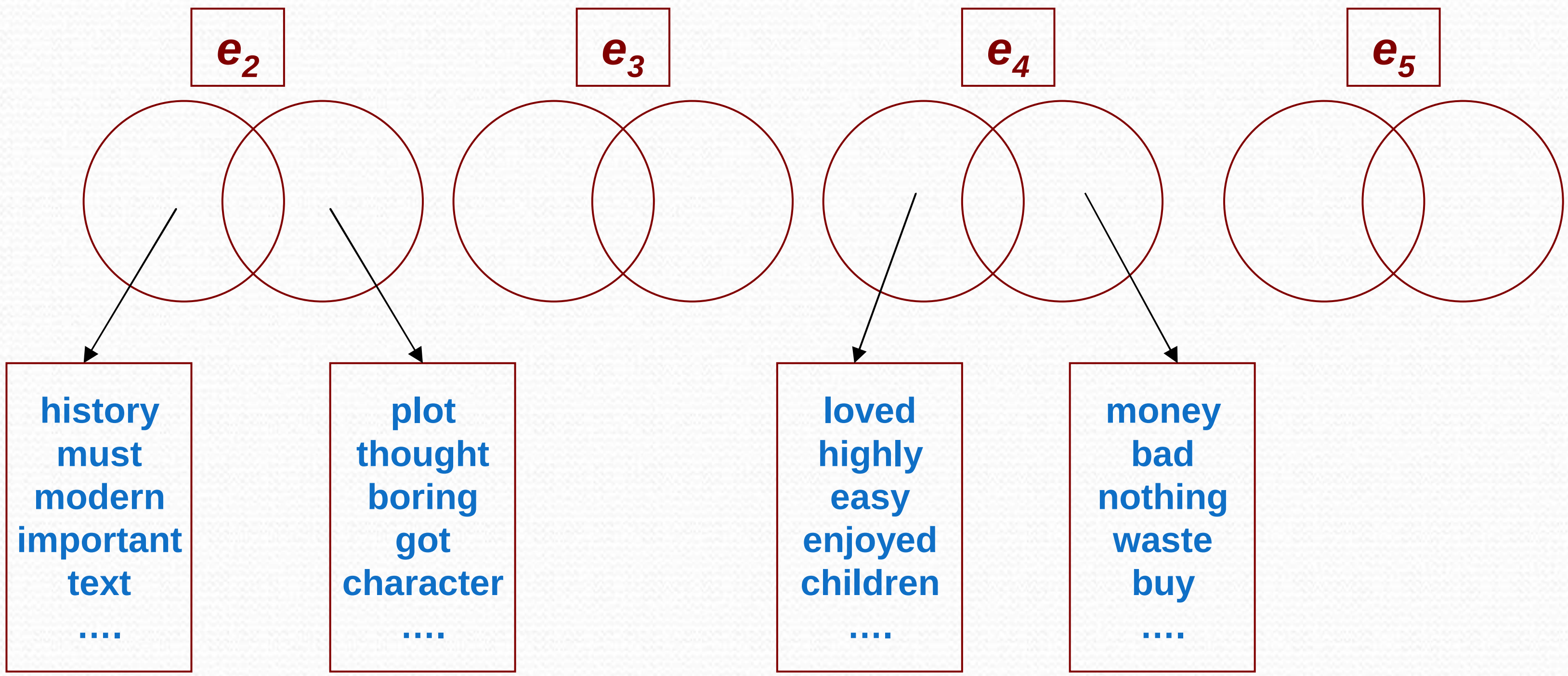
Our Algorithm

Steps:

- Identify important dimensions
- Identify unambiguous reviews
- **Identify relevant features and get user-feedback**
- Cluster along the selected dimension

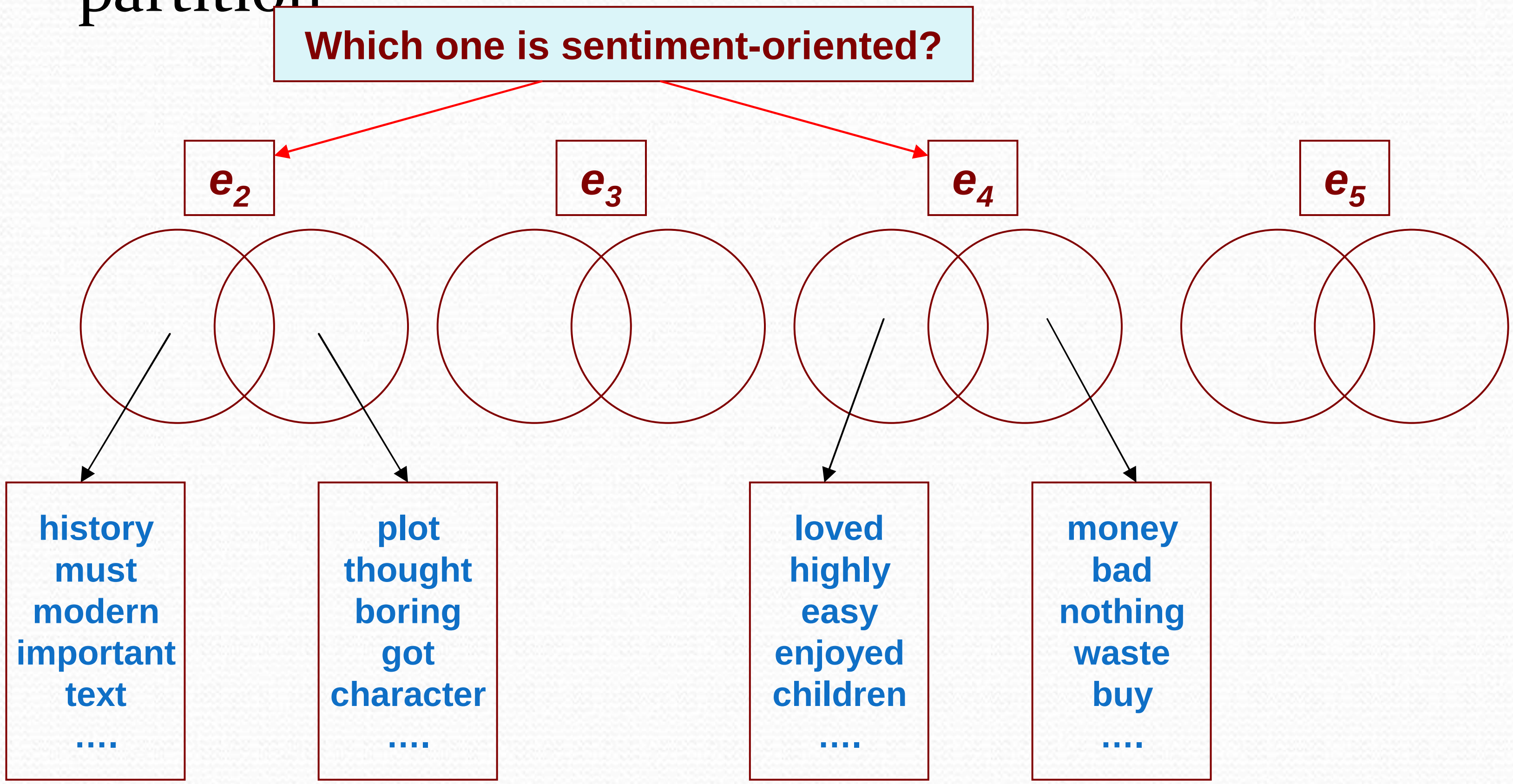
Identify relevant features

- We extract informative features characterizing each partition



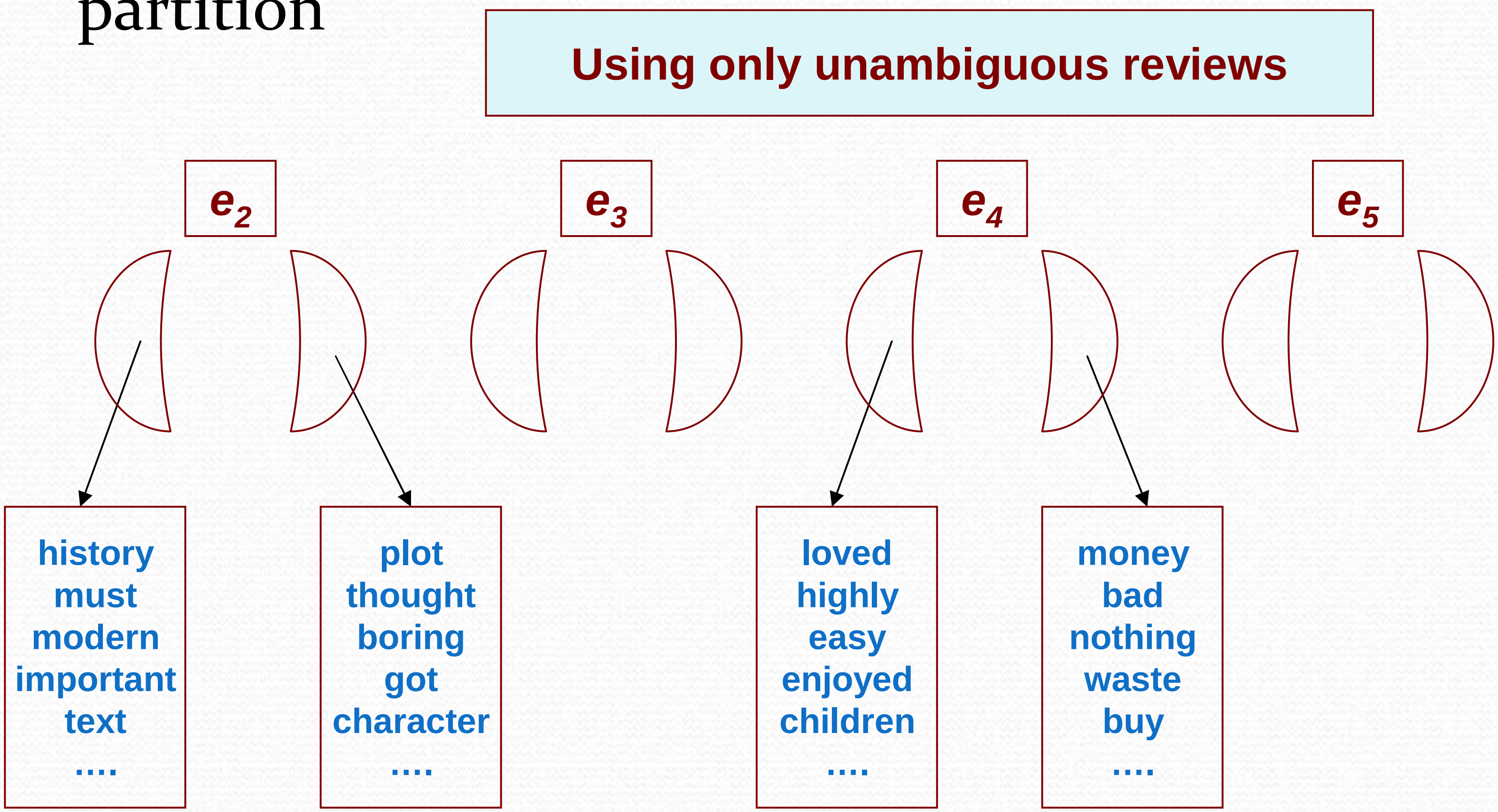
Identify relevant features

- We extract informative features characterizing each partition



Identify relevant features

- We extract informative features characterizing each partition



Identify Relevant Features

- **How to extract informative features characterizing a partition?**

Identify Relevant Features

- How to extract informative features characterizing a partition?

  - **Use Support Vector Machine**
  - **Why?**

Max-margin systems learn feature distribution well

Identify Relevant Features

- **Use SVM to learn the max-margin hyperplane (i.e., $\boldsymbol{w} \cdot \boldsymbol{x} - \boldsymbol{b} = 0$)**
- **Use *w* to extract informative features**
  - Features with large positive weight indicative of positive class
  - Features with large negative weight indicative of negative class

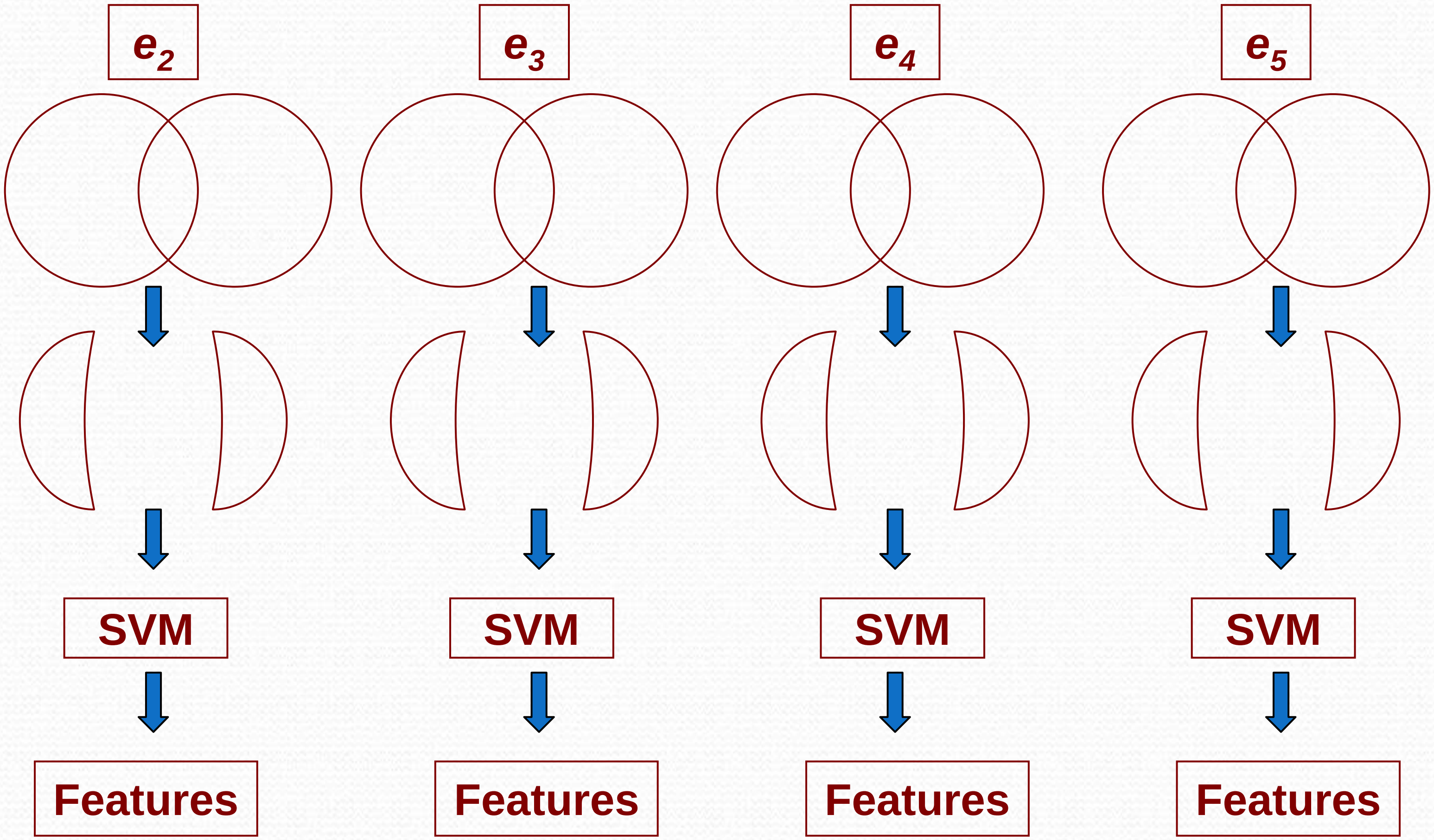
Identify Relevant Features

- **For each of the 2nd to 5th eigenvector *e*,**
  - Train SVM classifier on unambiguous clusters U_1 and U_2
  - Sort the features according to learned $\boldsymbol{w}$
  - List_1 -> top 100
  - List_2 -> bottom 100

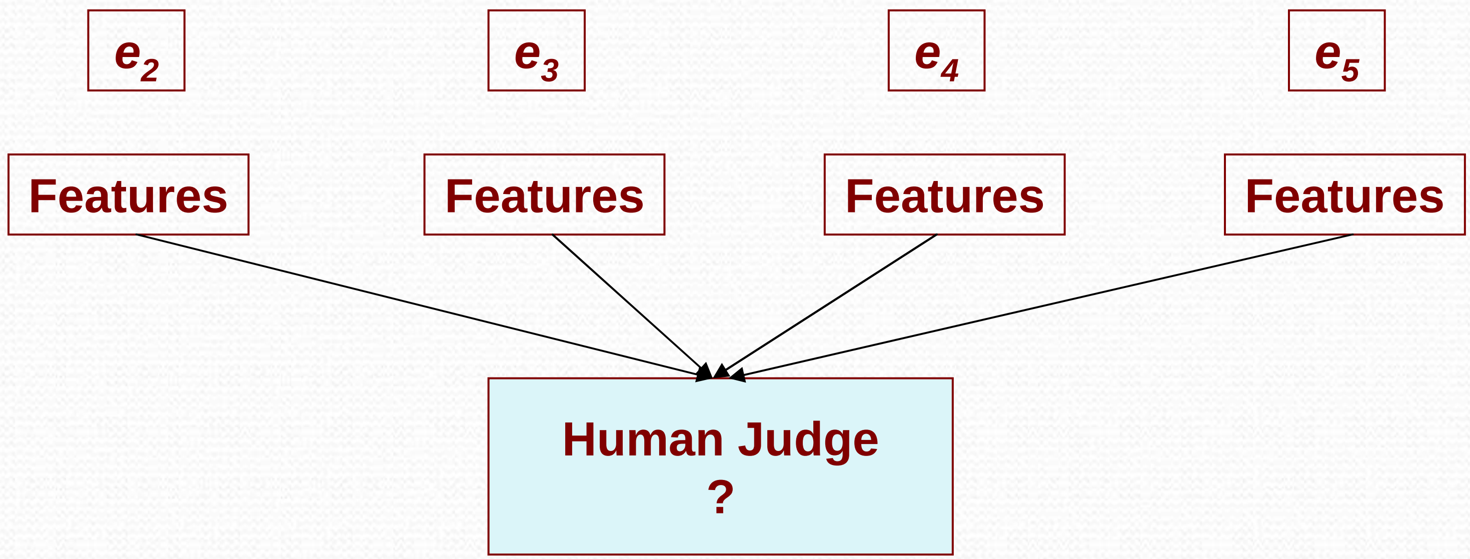
List_1: Informative Features for + Class

List_2: Informative Features for - Class

Identify Relevant Features



Get Human Feedback



Which one is sentiment-oriented?

Our Algorithm

Steps:

- Identify important dimensions
- Identify unambiguous reviews
- Identify relevant features and get user-feedback
- **Cluster along the selected dimension**

Cluster along selected dimension

- Get the dimension (eigenvector) selected by human
- **Use 2-means to cluster all data points using the selected eigenvector**

- End of algorithm

- **The algorithm for producing the feature list is unsupervised**

- **Except for human feedback, the whole clustering process is unsupervised**

- **Except for human feedback, the whole clustering process is unsupervised**

But, does it actually work?

Evaluation

- **Let's do the evaluation,**

Evaluation

- **Datasets**
  - Movie (Pang et al., 2002)
  - 4 datasets from Blitzer et al. (2007)
    - Kitchen and Housewares
    - Electronics
    - Books
    - DVD
  - each has 2000 points (1000 positives & 1000 negatives)

Evaluation

- **One more dataset**
  - 2-Newsgroup: sci.crypt and talks.politics
  - Goal: Science vs. Politics
  - It has 2000 points (1000 Science & 1000 Politics)

Show the difference between topic-based clustering and sentiment-based clustering

Evaluation

1. **Show the SVM-learned feature list used for user feedback**
   - **Is our feedback system easy and feasible?**

2. **Human Agreement Rate**
   - **Did humans actually find it easy?**

3. **Clustering Accuracy**
   - **Did we achieve sentiment-oriented clustering?**

Evaluation

1. **Show the SVM-learned-feature-list used for user-feedback**
   - **Is our feedback system easy and feasible?**

2. Human Agreement Rate
   - Did humans actually find it easy?

3. Clustering Accuracy
   - Did we achieve sentiment-oriented clustering?

Top Features

- **DVD**

Top Features: **DVD**

2nd eigenvector		3rd eigenvector	
C_1	C_2	C_1	C_2
worth	young	music	worst
bought	between	collection	money
series	actors	excellent	thought
money	men	wonderful	boring
got	cast	must	nothing
season	seems	loved	minutes
fan	job	perfect	waste
collection	beautiful	highly	saw
music	around	makes	pretty
tv	us	special	reviews
thought	our	performance	interesting
quality	director	picture	maybe
…..	…..	…..	…..

Top Features: **DVD**

2nd eigenvector		3rd eigenvector	
C_1	C_2	C_1	C_2
worth	young	music	**worst**
bought	between	collection	money
series	actors	**excellent**	thought
money	men	**wonderful**	**boring**
got	cast	**must**	**nothing**
season	seems	**loved**	minutes
fan	job	**perfect**	**waste**
collection	**beautiful**	**highly**	saw
music	around	makes	pretty
tv	us	**special**	reviews
thought	our	**performance**	interesting
quality	director	picture	maybe
…..	…..	…..	…..

Top Features: **DVD**

2nd eigenvector

C_1	C_2
worth	young
bought	between
series	actors
money	men
got	cast
season	seems
fan	job
collection	beautiful
music	around
tv	us
thought	our
quality	director
…..	…..

Doesn't Capture Sentiment

3rd eigenvector

C_1	C_2
music	worst
collection	money
excellent	thought
wonderful	boring
must	nothing
loved	minutes
perfect	waste
highly	saw
makes	pretty
special	reviews
performance	interesting
picture	maybe
…..	…..

Captures Sentiment

Top Features: **DVD**

4^{th} eigenvector C_1	4^{th} eigenvector C_2	5^{th} eigenvector C_1	5^{th} eigenvector C_2
video	series	saw	money
music	cast	watched	**quality**
found	fan	**loved**	video
feel	stars	**enjoy**	worth
bought	**original**	whole	found
workout	comedy	got	version
daughter	actors	family	picture
recommend	worth	series	**waste**
our	**classic**	season	**special**
disappointed	action	**liked**	sound
right	season	**entertaining**	stars
moves	big	lot	star
…..	…..	…..	…..

Top Features: **DVD**

4^{th} eigenvector

C_1	C_2
video	series
music	cast
found	fan
feel	stars
bought	original
workout	comedy
daughter	actors
recommen	worth
our	classic
disappointed	action
right	season
moves	big
…..	…..

Doesn't Capture Sentiment

5^{th} eigenvector

C_1	C_2
saw	money
watched	quality
loved	video
enjoy	worth
whole	found
got	version
family	picture
series	waste
season	special
liked	sound
entertaining	stars
lot	star
…..	…..

Doesn't Capture Sentiment

Top Features: **DVD**

2nd eigenvector

C_1	C_2
worth	young
bought	between
series	actors
money	men
got	cast
season	seems
fan	job
collection	beautiful
music	around
tv	us
thoug	
quali	
…..	

Doesn't Capture Sentiment

3rd eigenvector

C_1	C_2
music	worst
collection	money
excellent	thought
wonderful	boring
must	nothing
loved	minutes
perfect	waste
highly	saw
makes	pretty
special	reviews
f	resting
	aybe
	…..

Captures Sentiment

2nd eigenvector fails to capture sentiment for DVD

Top Features

- **Books**

Top Features: **Books**

2^{nd} eigenvector		**3^{rd} eigenvector**	
C_1	**C_2**	**C_1**	**C_2**
history	plot	series	buy
must	didn	man	bought
modern	thought	history	information
important	**boring**	character	**easy**
text	got	death	money
reference	character	between	recipes
excellent	couldn	war	pictures
provides	ll	seems	look
business	ending	political	**waste**
both	fan	american	copy
understanding	pages	during	**disappointed**
given	**funny**	plot	sure
…..	…..	…..	…..

Top Features: **Books**

2nd eigenvector

C_1	C_2
history	plot
must	didn
modern	thought
important	boring
text	got
reference	character
excellent	couldn
provides	ll
business	ending
both	fan
understanding	pages
given	funny
…..	…..

Doesn't Capture Sentiment

3rd eigenvector

C_1	C_2
series	buy
man	bought
history	information
character	easy
death	money
between	recipes
war	pictures
seems	look
political	waste
american	copy
during	disappointed
plot	sure
…..	…..

Doesn't Capture Sentiment

Top Features: **Books**

4th eigenvector C1	4th eigenvector C2	5th eigenvector C1	5th eigenvector C2
loved	money	**must**	**boring**
highly	**bad**	**wonderful**	series
easy	**nothing**	old	history
enjoyed	**waste**	feel	pages
children	buy	away	information
again	**anything**	children	between
although	**doesn**	year	**highly**
excellent	already	someone	page
understand	**instead**	man	**excellent**
three	seems	made	couldn
both	believe	thing	instead
looking	thought	buy	plot
…..	…..	…..	…..

Top Features: **Books**

4^{th} eigenvector

C_1	C_2
loved	money
highly	bad
easy	nothing
enjoyed	waste
children	buy
again	anything
although	doesn
excellent	already
understan	instead
three	seems
both	believe
looking	thought
…..	…..

Captures Sentiment

5^{th} eigenvector

C_1	C_2
must	boring
wonderful	series
old	history
feel	pages
away	information
children	between
year	highly
someone	page
man	excellent
made	couldn
thing	instead
buy	plot
…..	…..

Doesn't Capture Sentiment

Top Features: **Books**

4th eigenvector

C_1	C_2
loved	money
highly	bad
easy	nothing
enjoyed	waste
children	buy
again	anything
although	doesn
excellent	already
understan	instead
three	seems
bot	
look	
…	

Captures Sentiment

5th eigenvector

C_1	C_2
must	boring
wonderful	series
old	history
feel	pages
away	information
children	between
year	highly
someone	page
man	excellent
made	couldn
	instead
	plot
	…..

Doesn't Capture Sentiment

2nd eigenvector fails to capture sentiment for Books

Human Feedback

- **Lets see what happens to 2-Newsgroup**
- It will show the difference between topic clustering and sentiment clustering

Top Features: **Politics vs. Science**

2nd eigenvector	
C_1	**C_2**
serdar	sternlight
armenian	wouldn
turkey	pgp
armenians	crypto
muslims	algorithm
sdpa	isn
argic	likely
davidian	access
dbd@urart	idea
troops	cryptography
genocide	net
muslim	digex
…..	……

Captures desired dimension

3rd eigenvector	
C_1	**C_2**
beyer	escrow
arabs	sternlight
andi	algorithm
research	access
israelis	net
tim	des
uci	privacy
ab	uk
z@virginia	systems
holocaust	pgp
employer	care
clock	strnlght@netcom
…..	…..

Captures desired dimension

Top Features: **Politics vs. Science**

4th eigenvector

C_1	C_2
serbs	standard
palestinians	sternlight
muslims	des
wrong	escrow
department	employer
bosnia	net
live	york
matter	jake
freedom	code
politics	algorithm
mcgill	private
side	jake@bony
…..	…..

Captures desired dimension

5th eigenvector

C_1	C_2
escrow	internet
serial	uucp
algorithm	uk
chips	net
ensure	quote
care	ac
strong	co
police	didn
omissions	ai
excepted	mit
accuracy	guy
-bit	cute
…..	…..

Doesn't capture desired

Top Features: **Politics vs. Science**

2^{nd}, 3^{rd}, 4^{th} eigenvectors all capture right dimension

Topic based clustering is different from sentiment based clustering

Evaluation

1. Show the SVM-learned-feature-list used for user-feedback
   - Is our feedback system easy and feasible?

2. **Human Agreement Rate**
   - Did humans actually find it easy?

3. Clustering Accuracy
   - Did we achieve sentiment-oriented clustering?

Human Agreement Rate

- **Human Judges:**
  - 5 computer science graduate students

- **Guidelines:**
  - We show top 100 features for each cluster
  - We inform them of desired clustering dimension beforehand

Human Agreement Rate

- **Did they find sentiment dimension correctly?**

	Correctly found
Movie	**5/5**
Books	**5/5**
DVD	**5/5**
Electronics	**5/5**
Kitchen	**4/5**

Human Agreement Rate

- **Almost perfect human agreement rate**
- **Feedback approach is feasible**

Human Feedback

- Eigenvector selected by all human-judges

	Eigenvector
Movie	2
Books	4
DVD	3
Electronics	3
Kitchen	2

For 3 out of 5 domains, sentiment is hidden

Evaluation

1. Show the SVM-learned-feature-list used for user-feedback
   - Is our feedback system easy and feasible?

2. Human Agreement Rate
   - Did humans actually find it easy?

3. **Clustering Accuracy**
   - Did we achieve sentiment-oriented clustering?

Clustering Accuracy

- **Baselines:**
- 2nd eigenvector only

- **Evaluation Metric:**
  - Accuracy

Clustering Accuracy

	POL	MOV	KIT	BOO	DVD	ELE
2nd eigenvector only						

2 Newsgroup

5 Sentiment Domains

Clustering Accuracy

	POL	MOV	KIT	BOO	DVD	ELE
2nd eigenvector only	93.7%	70.9%	69.7%	58.9%	55.3%	50.8%

Clustering accuracy high for POL

Clustering of sentiment is tough

Clustering Accuracy

	POL	MOV	KIT	BOO	DVD	ELE
2nd eigenvector only	93.7%	70.9%	69.7%	58.9%	55.3%	50.8%

Clustering accuracy low
2nd Eigenvectors fails to capture sentiment

Clustering Accuracy

	POL	MOV	KIT	BOO	DVD	ELE
2nd eigenvector only	93.7%	70.9%	69.7%	58.9%	55.3%	50.8%
Our approach	93.7%	70.9%	69.7%	69.5%	70.8%	65.8%

We achieve best performance for all sentiment domain

Clustering Accuracy

	POL	MOV	KIT	BOO	DVD	ELE
2nd eigenvector only	93.7%	70.9%	69.7%	58.9%	55.3%	50.8%
Our approach	93.7%	70.9%	69.7%	69.5%	70.8%	65.8%

Human helps to identify the hidden sentiment dimension

Conclusion

- Proposed a new feedback-oriented clustering algorithm:
  - **It produces the desired clustering**
  - **It requires only simple human feedback**

Conclusion

- Finally, we have a system where

Machine meets human

- Thank you