Correlation Clustering

Bounding and Comparing Methods Beyond ILP

Micha Elsner and Warren Schudy

Department of Computer Science
Brown University

May 26, 2009

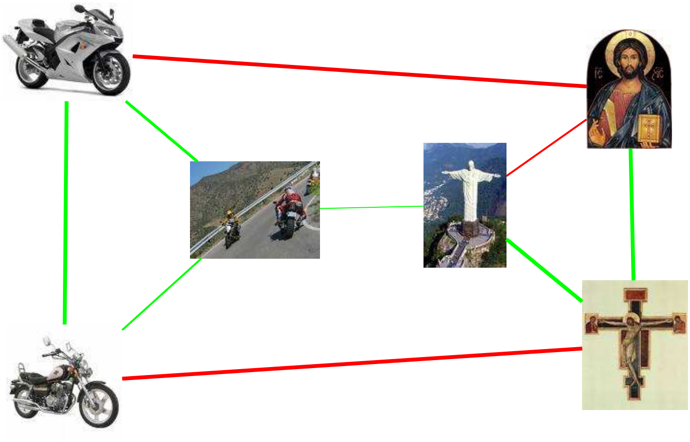
Document clustering

rec.motorcycles

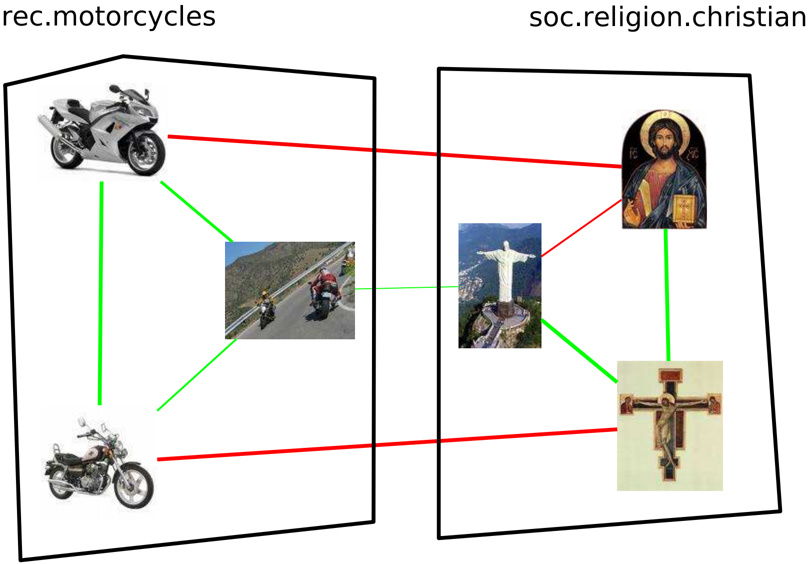
soc.religion.christian

Document clustering: pairwise decisions

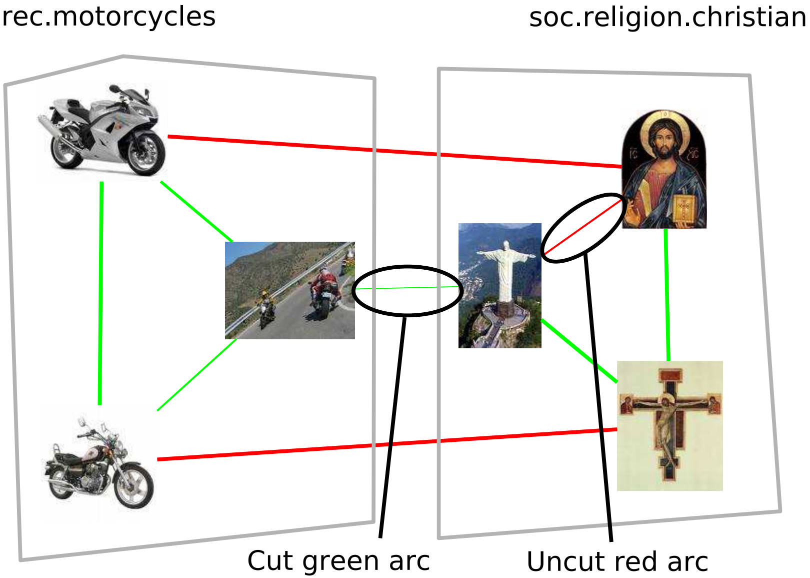
rec.motorcycles   soc.religion.christian



Document clustering: partitioning



How good is this?



Correlation clustering

Given green edges w^+ and red edges w^-...
Partition to minimize disagreement.

$$\min_x x_{ij} w_{ij}^- + (1 - x_{ij}) w_{ij}^+$$

$$\text{s.t.} \quad \begin{array}{l} x_{ij} \text{ form a consistent clustering} \\ \text{relation must be transitive: } x_{ij} \text{ and } x_{jk} \rightarrow x_{ik} \end{array}$$

Minimization is NP-hard (Bansal *et al.* '04).
How do we solve it?

ILP scalability

ILP:

- $O(n^2)$ variables (each pair of points).
- $O(n^3)$ constraints (triangle inequality).
- Solvable for about **200 items**.

Good enough for single-document coreference or generation.
Beyond this, need something else.

Previous applications

- Coreference resolution (Soon *et al.* '01), (Ng+Cardie '02), (McCallum+Wellner '04), (Finkel+Manning '08).
- Grouping named entities (Cohen+Richman '02).
- Content aggregation (Barzilay+Lapata '06).
- Topic segmentation (Malioutov+Barzilay '06).
- Chat disentanglement (Elsner+Charniak '08).

Solutions: **heuristic**, **ILP**, **approximate**, **special-case**,

This talk

Not about when you should use correlation clustering.

- When you can't use ILP, what should you do?
- How well can you do in practice?
- Does the objective predict real performance?

This talk

Not about when you should use correlation clustering.

- When you can't use ILP, what should you do?
  - Greedy voting scheme, then local search.
- How well can you do in practice?
- Does the objective predict real performance?

This talk

Not about when you should use correlation clustering.

- When you can't use ILP, what should you do?
  - Greedy voting scheme, then local search.
- How well can you do in practice?
  - Reasonably close to optimal.
- Does the objective predict real performance?

This talk

Not about when you should use correlation clustering.

- When you can't use ILP, what should you do?
  - Greedy voting scheme, then local search.
- How well can you do in practice?
  - Reasonably close to optimal.
- Does the objective predict real performance?
  - Often, but not always.

Overview

- Motivation
- Algorithms
- Bounding
- Task 1: Twenty Newsgroups
- Task 2: Chat Disentanglement
- Conclusions

Algorithms

Some fast, simple algorithms from the literature.

Greedy algorithms

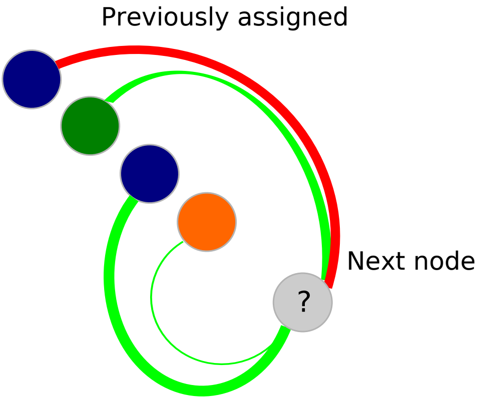
- First link
- Best link
- Voted link
- Pivot

Local search

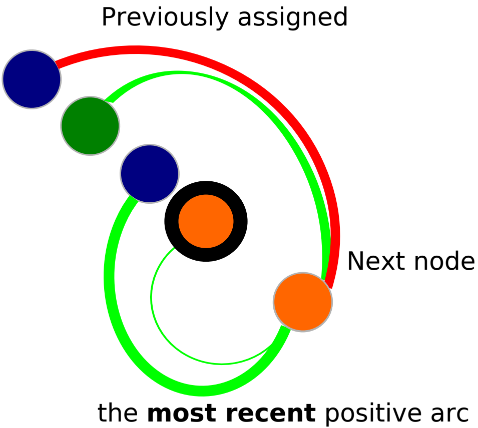
- Best one-element move (BOEM)
- Simulated annealing

Greedy algorithms

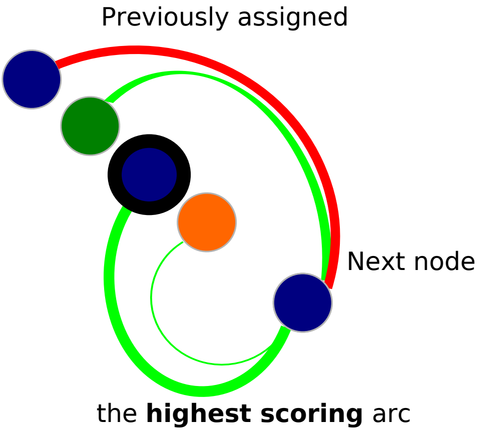
Step through the nodes in random order.
Use a linking rule to place each unlabeled node.



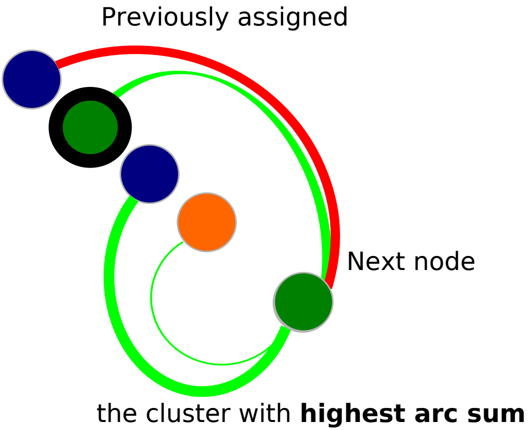
First link (Soon '01)



Best link (Ng+Cardie '02)

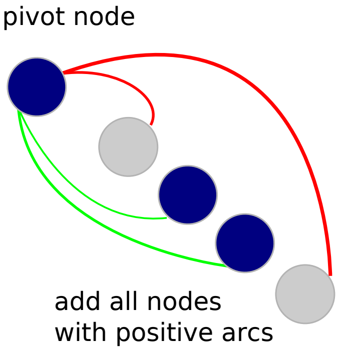


Voted link


Previously assigned
Next node
the cluster with **highest arc sum**

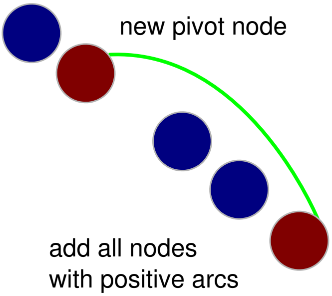
Pivot (Ailon+al '08)

Create each whole cluster at once.
Take the first node as the pivot.



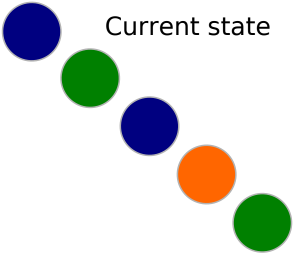
Pivot

Choose the next unlabeled node as the pivot.



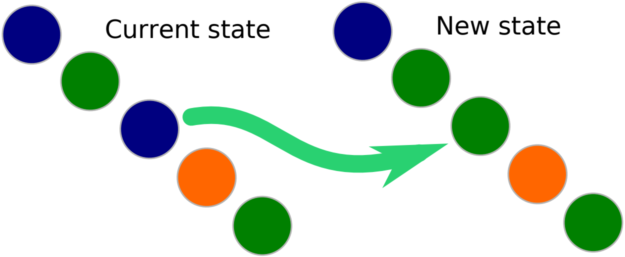
Local searches

One-element moves change the label of a single node.



Local searches

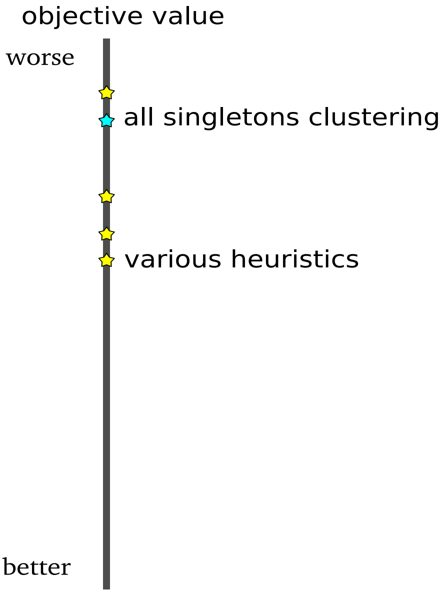
One-element moves change the label of a single node.



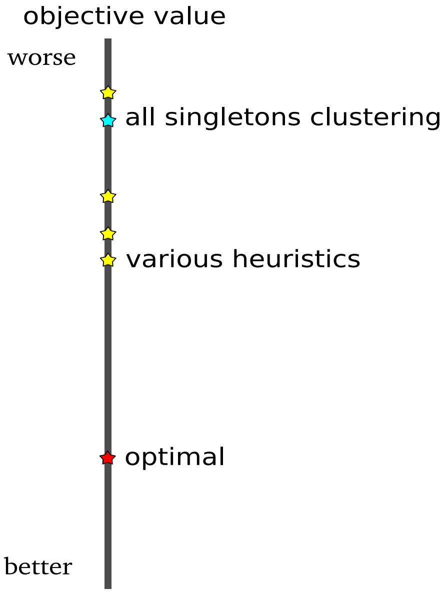
- Greedily: best one-element move (BOEM)
- Stochastically (annealing)

Overview

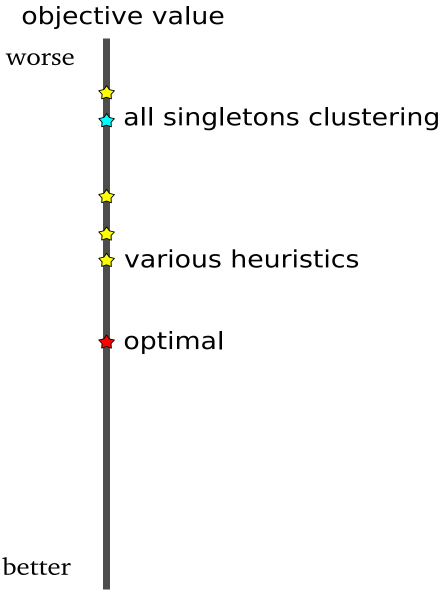
Why bound?


objective value
worse
all singletons clustering
various heuristics
better

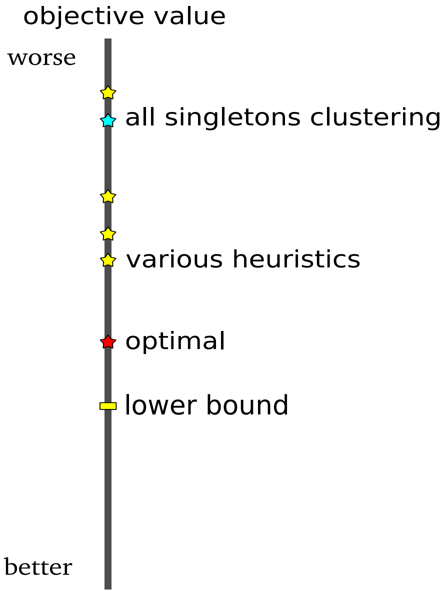
Why bound?


objective value
worse
all singletons clustering
various heuristics
optimal
better

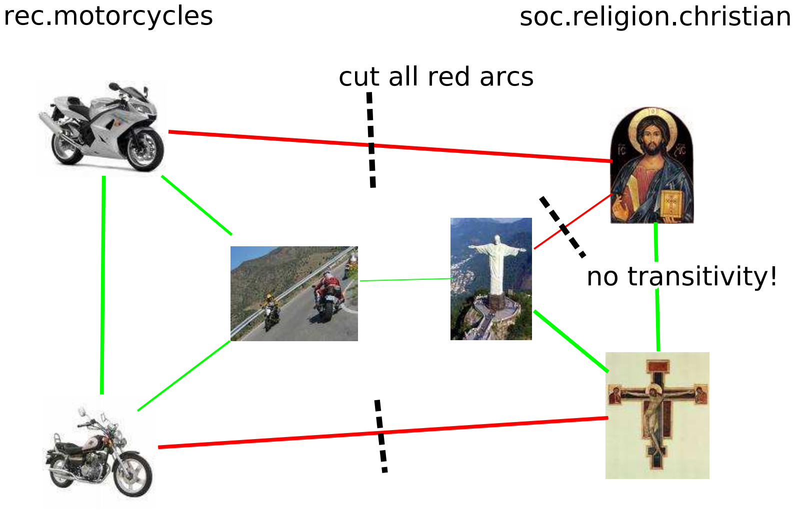
Why bound?


objective value
worse
all singletons clustering
various heuristics
optimal
better

Why bound?


objective value
worse
all singletons clustering
various heuristics
optimal
lower bound
better

Trivial bound from previous work



Semidefinite programming bound (Charikar *et al.*‘05)

Represent each item by an n-dimensional basis vector:

For an item in cluster c, vector r is:

$$(\underbrace{0, 0, \ldots, 0,}_{c-1} 1, \underbrace{0, \ldots, 0}_{n-c})$$

For two items clustered together, $r_i \bullet r_j = 1$.
Otherwise $r_i \bullet r_j = 0$.

Semidefinite programming bound (Charikar *et al.*'05)

Represent each item by an n-dimensional basis vector:

For an item in cluster c, vector r is:

$$(\underbrace{0, 0, \ldots, 0,}_{c-1} 1, \underbrace{0, \ldots, 0}_{n-c})$$

For two items clustered together, $r_i \bullet r_j = 1$.
Otherwise $r_i \bullet r_j = 0$.

Relaxation

Allow r_i to be any real-valued vectors with:

- Unit length.
- All products $r_i \bullet r_j$ non-negative.

Semidefinite programming bound (2)

Semidefinite program (SDP)

$$\min_r \sum (r_i \bullet r_j) w_{ij}^- + (1 - r_j \bullet r_j) w_{ij}^+$$
$$\begin{aligned} & r_i \bullet r_i = 1 && \forall i \\ \text{s.t.} \quad & r_i \bullet r_j \geq 0 && \forall i \neq j \end{aligned}$$

Objective and constraints are linear in the dot products of the r_i.

Semidefinite programming bound (2)

Semidefinite program (SDP)

$$
\begin{aligned}
&\min_x \sum x_{ij} w_{ij}^- + (1 - x_{ij}) w_{ij}^+ \\
&\qquad x_{ij} = 1 \qquad\qquad \forall i \\
\text{s.t.}\ &\qquad x_{ij} \geq 0 \qquad\qquad \forall i \neq j
\end{aligned}
$$

Objective and constraints are linear in the dot products of the r_i.

Replace dot products with variables x_{ij}.
New constraint: x_{ij} must be dot products of some vectors r!

Semidefinite programming bound (2)

Semidefinite program (SDP)

$$\begin{aligned} \min_x \sum x_{ij} w^-_{ij} + (1 - x_{ij}) w^+_{ij} & \\ x_{ij} = 1 \quad & \forall i \\ \text{s.t.} \quad x_{ij} \geq 0 \quad & \forall i \neq j \\ \text{matrix } X \text{ PSD} & \end{aligned}$$

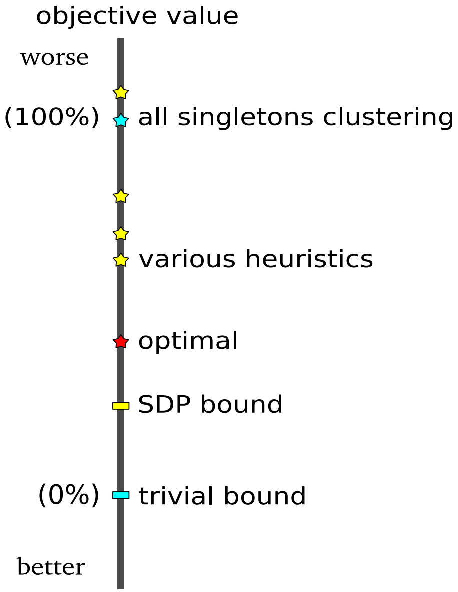
Objective and constraints are linear in the dot products of the r_i.

Replace dot products with variables x_{ij}.
New constraint: x_{ij} must be dot products of some vectors r!
Equivalent: matrix X is *positive semi-definite*.

Solving the SDP

- SDP bound previously studied in theory.
- We actually solve it!
- Conic Bundle method (Helmberg '00).
  - Scales to several thousand points.
- Iteratively improves bounds.
  - Run for 60 hrs.

Bounds


objective value
worse
(100%)
all singletons clustering
various heuristics
optimal
SDP bound
(0%)
trivial bound
better

Overview

Twenty Newsgroups

A standard clustering dataset.
Subsample of 2000 posts.

Hold out four newsgroups to train a pairwise classifier:

Twenty Newsgroups

A standard clustering dataset.
Subsample of 2000 posts.

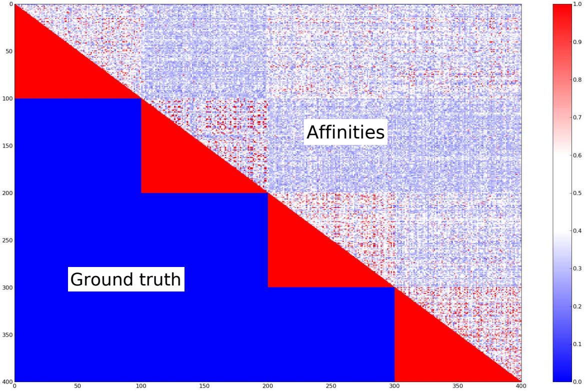
Hold out four newsgroups to train a pairwise classifier:

Is this message pair from the same newsgroup?

- Word overlap (bucketed by IDF).
- Cosine in LSA space.
- Overlap in subject lines (by IDF).

Max-ent model with F-score of 29%.

Affinity matrix



Results

		Objective	F-score	One-to-one
Bounds	Trivial bound	0%		
	SDP bound	51.1%		

Results

		Objective	F-score	One-to-one
Bounds	Trivial bound	0%		
	SDP bound	51.1%		
Local search	Vote/BOEM	55.8%		
	Sim Anneal	56.3%		
	Pivot/BOEM	56.6%		
	Best/BOEM	57.6%		
	First/BOEM	57.9%		
	BOEM	60.1%		

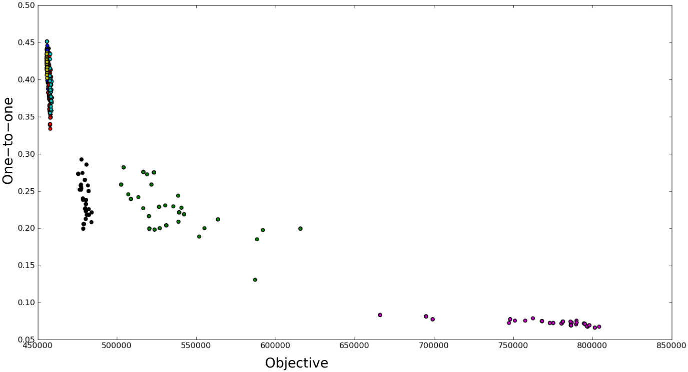
Results

		Objective	F-score	One-to-one
Bounds	Trivial bound	0%		
	SDP bound	51.1%		
Local search	Vote/BOEM	55.8%		
	Sim Anneal	56.3%		
	Pivot/BOEM	56.6%		
	Best/BOEM	57.6%		
	First/BOEM	57.9%		
	BOEM	60.1%		
Greedy	Vote	59.0%		
	Pivot	100%		
	Best	138%		
	First	619%		

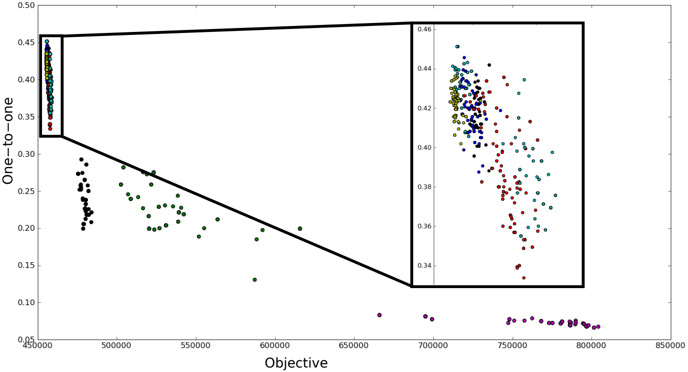
Results

		Objective	F-score	One-to-one
Bounds	Trivial bound	0%		
	SDP bound	51.1%		
Local search	Vote/BOEM	55.8%	33	41
	Sim Anneal	56.3%	31	36
	Pivot/BOEM	56.6%	32	39
	Best/BOEM	57.6%	31	38
	First/BOEM	57.9%	30	36
	BOEM	60.1%	30	35
Greedy	Vote	59.0%	29	35
	Pivot	100%	17	27
	Best	138%	20	29
	First	619%	11	8

Objective vs. metrics



Objective vs. metrics



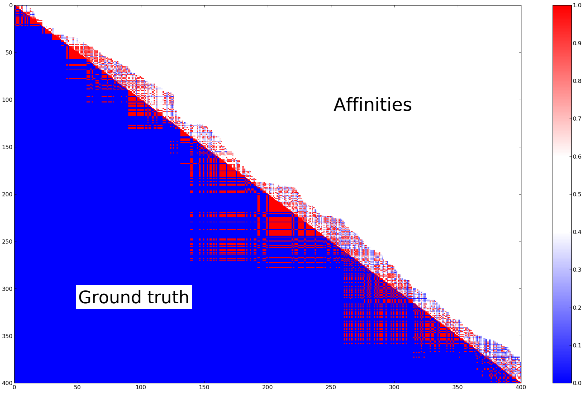
Overview

Chat disentanglement

Separate IRC chat log into threads of conversation.
800 utterance dataset and max-ent classifier from (Elsner+Charniak '08).
Classifier is run on pairs less than 129 seconds apart.

Ruthe	question: what could cause linux not to find a dhcp server?
Christiana	Arlie: I dont eat bananas.
Renate	Ruthe, the fact that there isn't one?
Arlie	Christiana, you should, they have lots of potassium goodness
Ruthe	Renate, xp computer finds it
Renate	eh? dunno then
Christiana	Arlie: I eat cardboard boxes because of the fibers.

Affinity matrix



Results

		Objective	Local	One-to-one
Bounds	Trivial bound	0%		
	SDP bound	13.0%		

Results

		Objective	Local	One-to-one
Bounds	Trivial bound	0%		
	SDP bound	13.0%		
Local search	First/BOEM	19.3%		
	Vote/BOEM	20.0%		
	Sim Anneal	20.3%		
	Best/BOEM	21.3%		
	BOEM	21.5%		
	Pivot/BOEM	22.0%		

Results

		Objective	Local	One-to-one
Bounds	Trivial bound	0%		
	SDP bound	13.0%		
Local search	First/BOEM	19.3%		
	Vote/BOEM	20.0%		
	Sim Anneal	20.3%		
	Best/BOEM	21.3%		
	BOEM	21.5%		
	Pivot/BOEM	22.0%		
Greedy	Vote	26.3%		
	Best	37.1%		
	Pivot	44.4%		
	First	58.3%		

Results

		Objective	Local	One-to-one
Bounds	Trivial bound	0%		
	SDP bound	13.0%		
Local search	First/BOEM	19.3%	74	41
	Vote/BOEM	20.0%	73	**46**
	Sim Anneal	20.3%	73	42
	Best/BOEM	21.3%	73	43
	BOEM	21.5%	72	22
	Pivot/BOEM	22.0%	72	**45**
Greedy	Vote	26.3%	72	44
	Best	37.1%	67	40
	Pivot	44.4%	66	39
	First	58.3%	62	39

Objective doesn't always predict performance

Most edges have weight .5:

- Some systems link too much.
- Doesn't affect local metric much...
- But global metric suffers.

In this situation, useful to have an external measure of quality.

Better inference is still useful:

- Vote/BOEM 12% better than (Elsner+Charniak '08).
- Exact same classifier!

Overview

Motivation

Algorithms

Bounding

Task 1: Twenty Newsgroups

Task 2: Chat Disentanglement

Conclusions

Conclusions

- Always use local search!
- Best greedy algorithm is voting.

Conclusions

- Always use local search!
- Best greedy algorithm is voting.
- SDP provides a tighter bound than previous work.
- Best heuristics are not too far from optimal.

Conclusions

- Always use local search!
- Best greedy algorithm is voting.
- SDP provides a tighter bound than previous work.
- Best heuristics are not too far from optimal.
- Better inference usually provides better solutions.
- But not always!
  - Especially for the top few solutions.
  - Useful to check statistics like number of clusters.

Conclusions

- Always use local search!
- Best greedy algorithm is voting.
- SDP provides a tighter bound than previous work.
- Best heuristics are not too far from optimal.
- Better inference usually provides better solutions.
- But not always!
  - Especially for the top few solutions.
  - Useful to check statistics like number of clusters.
- More experiments and discussion in the paper.

Acknowledgements

Software is available:

http://cs.brown.edu/∼melsner

Thanks:

- Christoph Helmberg
- Claire Matheiu
- Lidan Wang and Doug Oard
- Three reviewers