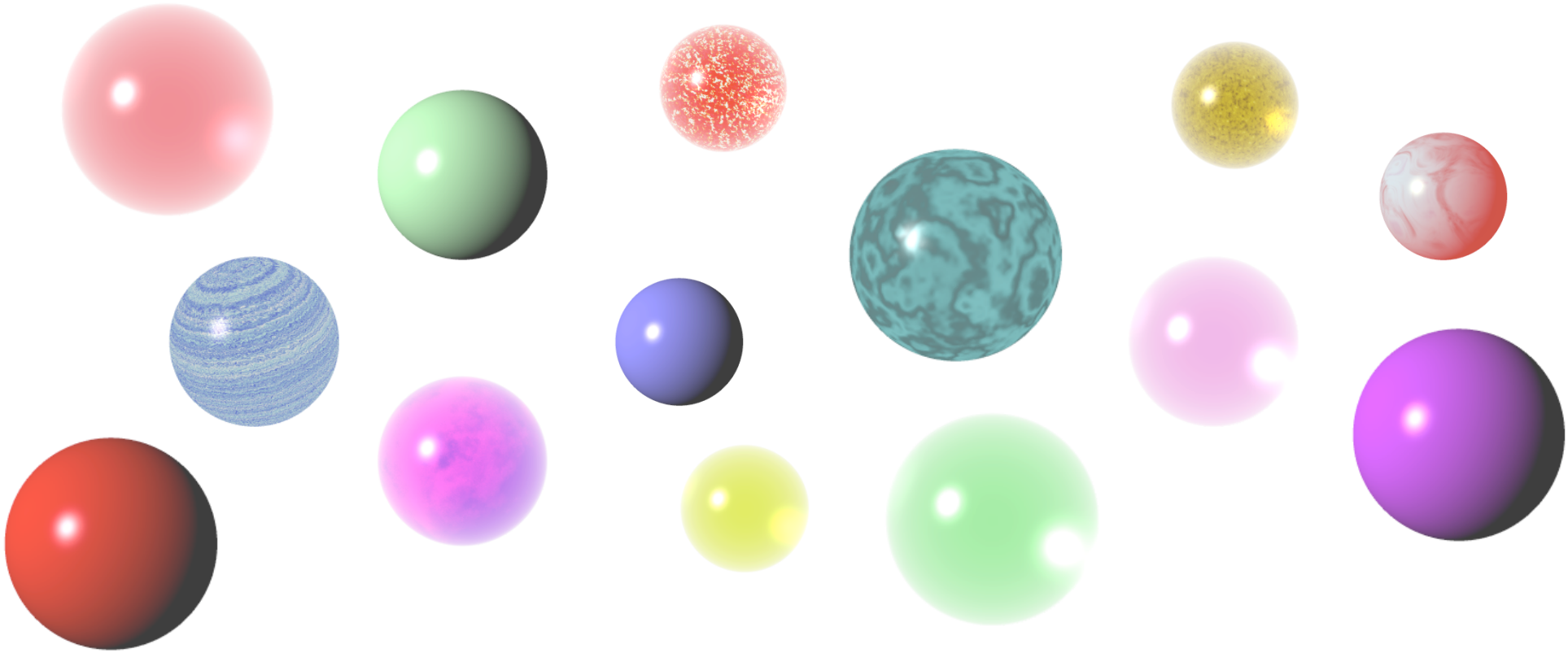
Supervised Clustering with SVMs

Thomas Finley, Thorsten Joachims
Cornell University
August 2005

Supervised Clustering Talk Outline

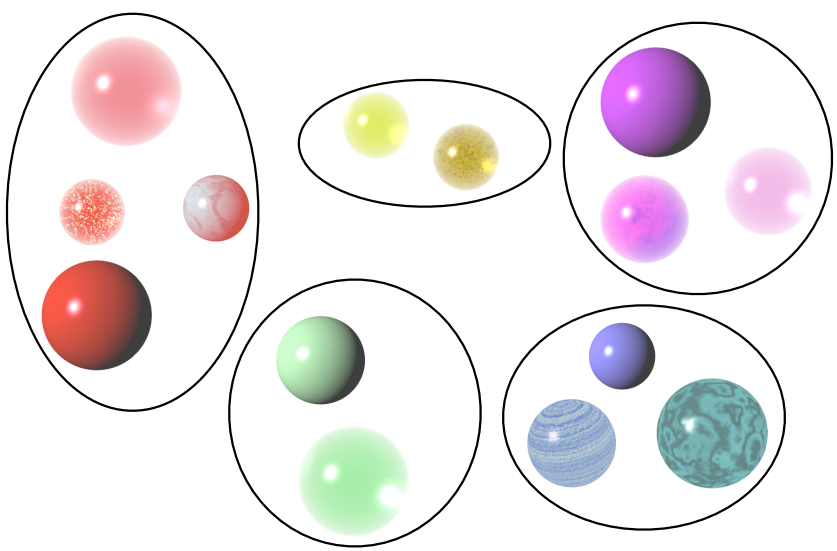
- Supervised clustering motivation
- Problem statement, difficulties
- Learning to cluster with SVM^{struct}
- Application to real problems

Clustering Marbles

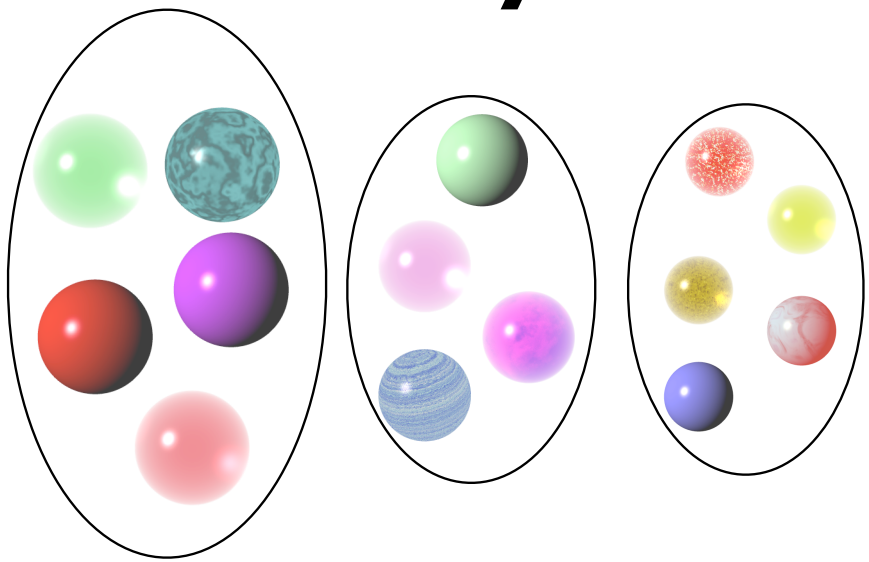


By what criteria do we consider two marbles “similar”?

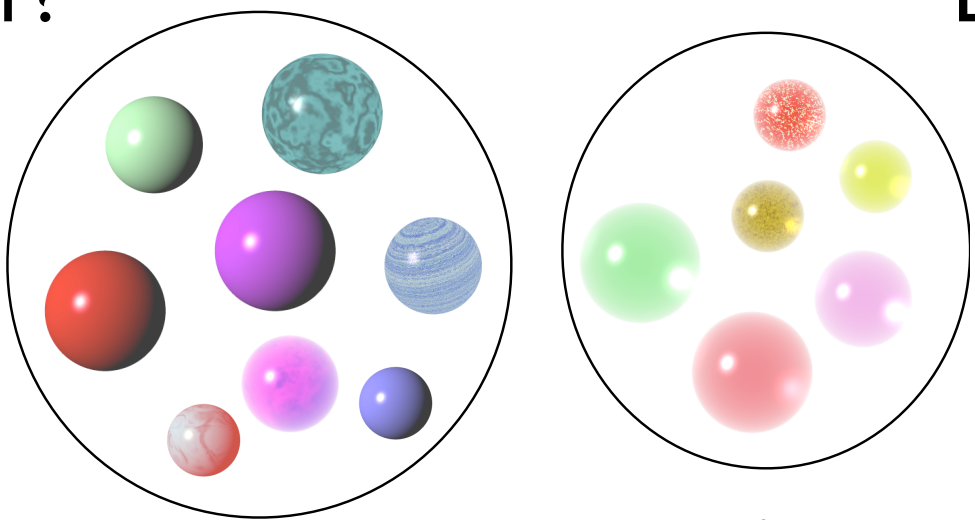
Multiple Possible Criteria for Similarity



By Color?



By Size?

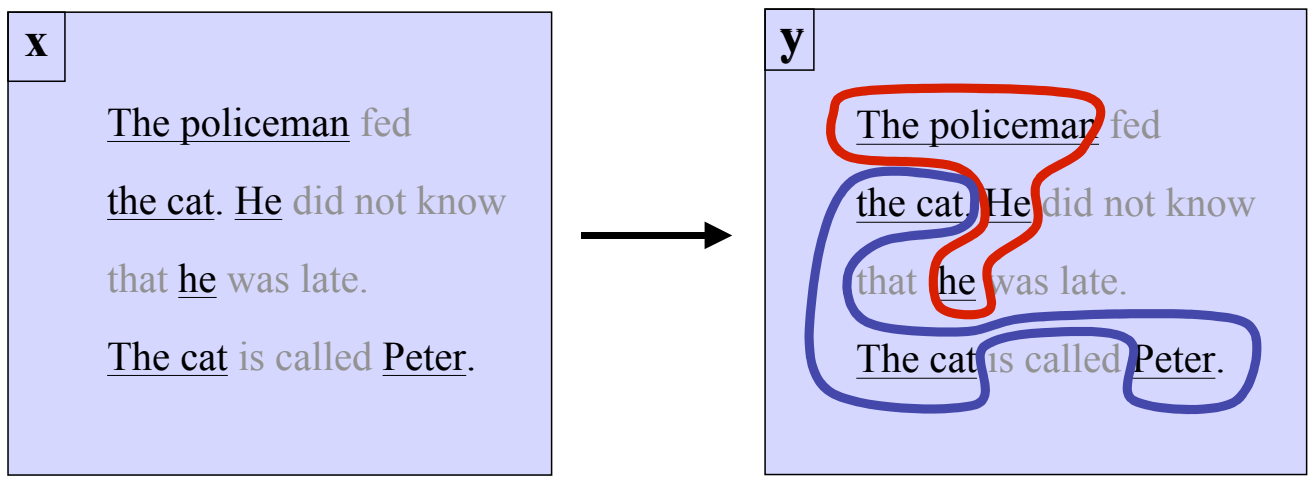


By Transparency?

How to Adjust

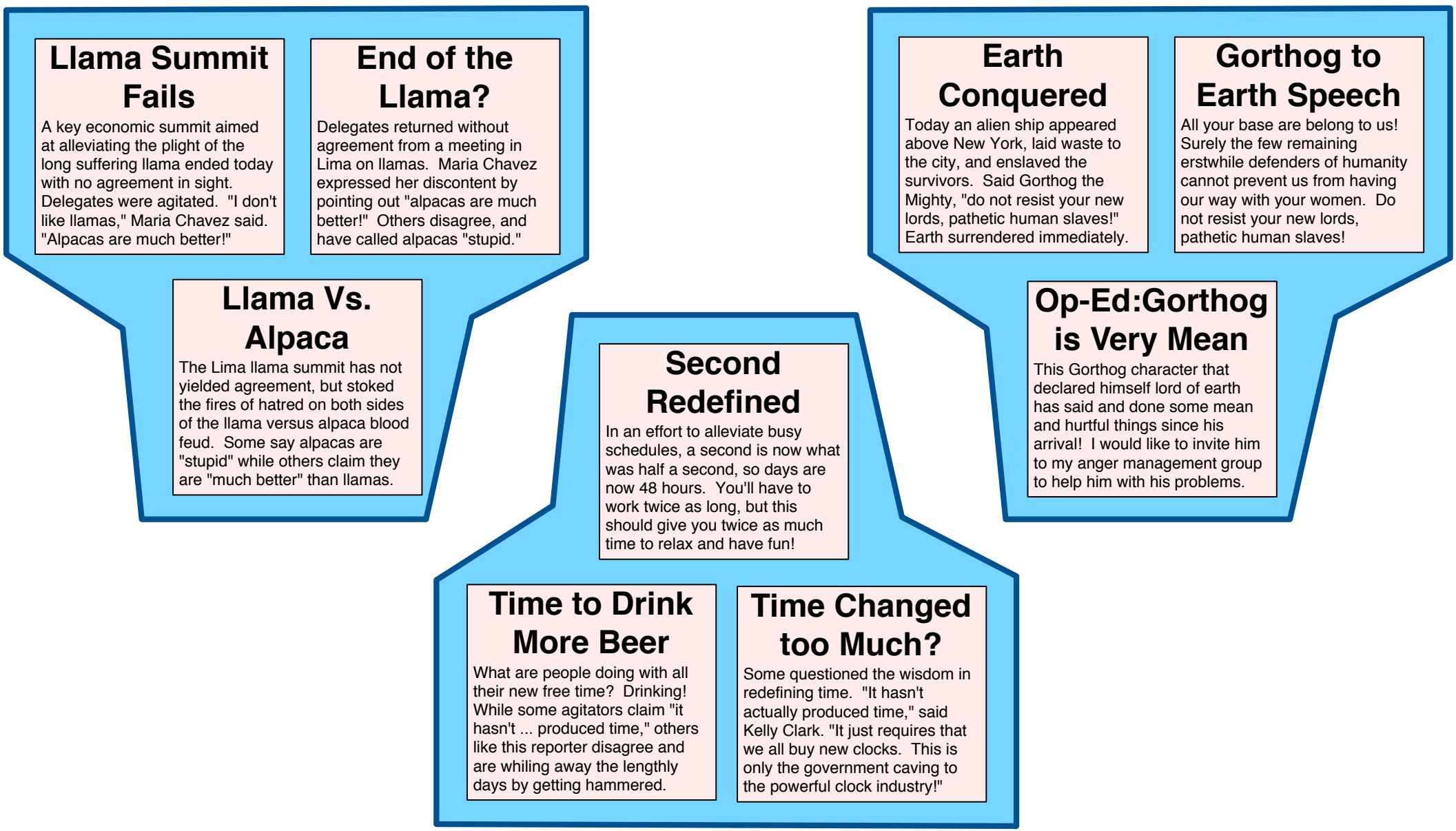
- **Manual -** Adjust to get desired clusters.
- **Semi-supervised clustering -** Provide constraints on item pairs. Clustering algorithm works to satisfy the constraints.
- **Supervised clustering -** Provide a series of tuples: an item set, and a clustering of that set, and learning how to cluster.

Noun Phrase Coreference



- Given the noun phrases in a document, cluster by which refer to the same entity.

News Story Clustering



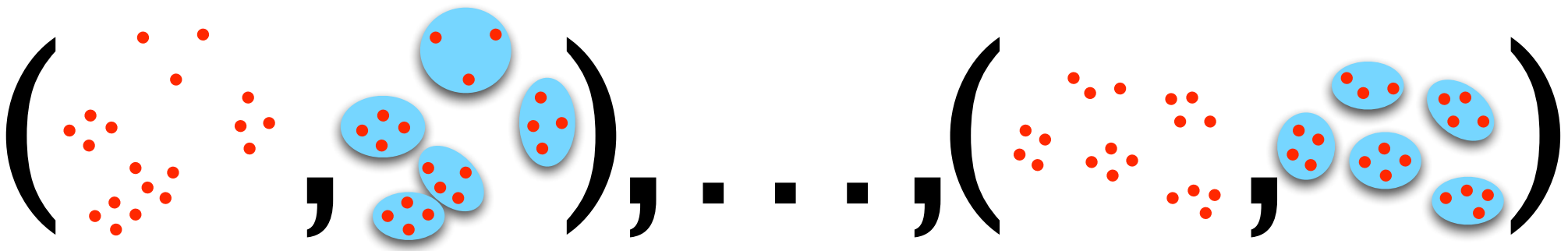
Supervised Clustering Talk Outline

- Supervised clustering motivation
- Problem statement, difficulties
- Learning to cluster with SVM^{struct}
- Application to real problems

How Do We Learn?

- A sequence of *n* training examples.

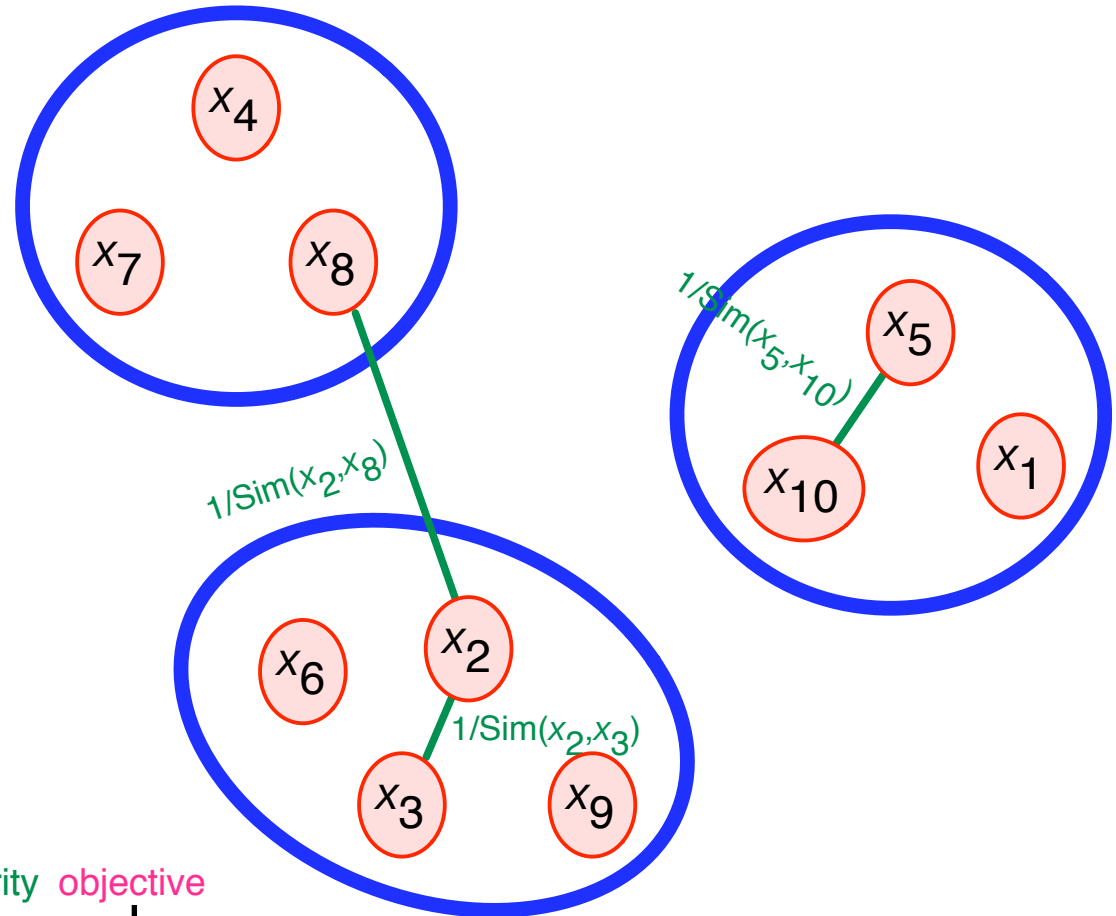
$$(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$$

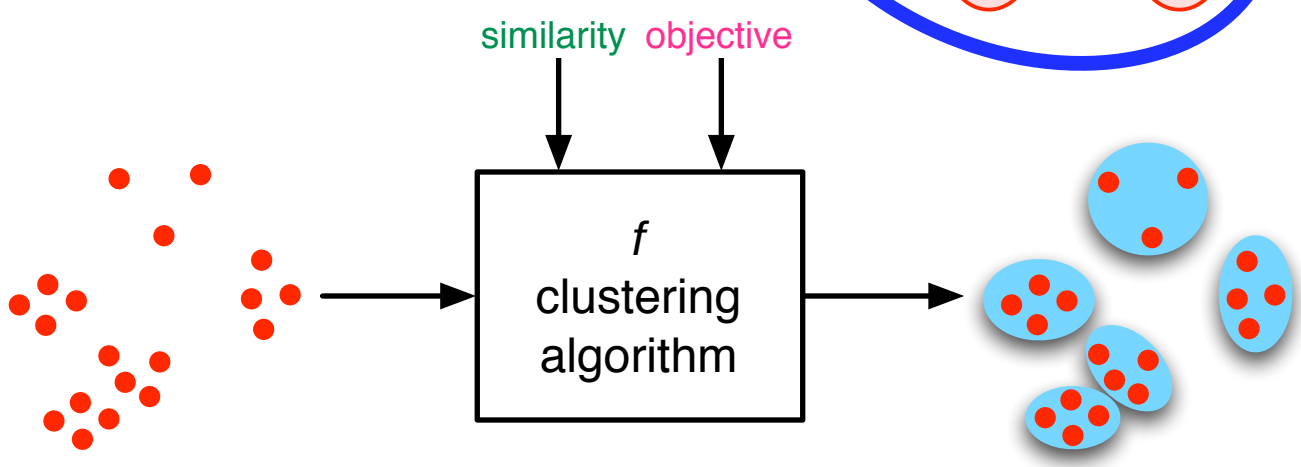


- From this training data we learn how to cluster future sets of items.

Simple Clustering

- Given a set of m items.
- Similarity measure between item pairs.
- Produce partitioning of the item set w.r.t. similarity measure over an objective function.





Clustering Objective Function

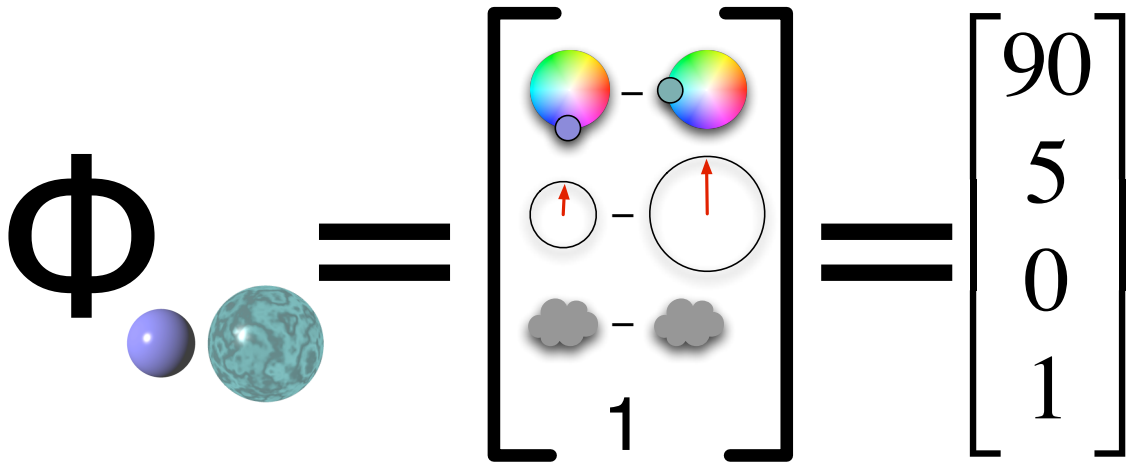
- Clusterer assigns items to clusters to maximize an objective function.
- Objective function here: sum of similarity of pairs in same cluster (Bansal, et al. 2002). Result: {a,b,c,d}, {e,f,g}, {h,i}
- Allows discrepancies if net effect is positive.

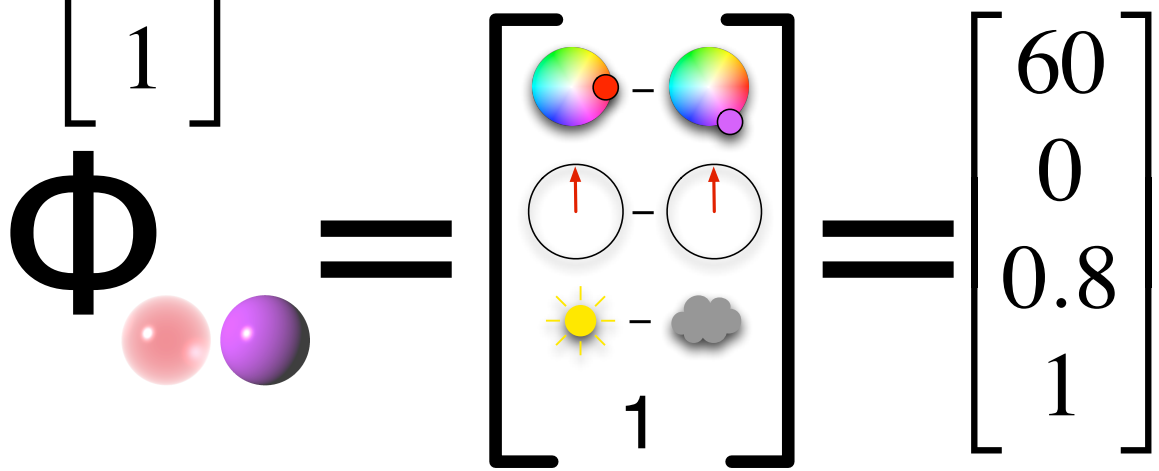
	a	b	c	d	e	f	g	h	i	
a	·	9	-9	1	-7	-5	-2	-6	-8	a
b		·	7	9	-8	-3	-4	8	-6	b
c			·	9	-4	-3	-4	-9	-5	c
d				·	-8	-4	-9	-9	-3	d
e					·	4	7	-3	-6	e
f						·	6	-6	-5	f
g							·	-8	-4	g
h								·	4	h
i									·	i

Matrix of similarities!

Pairwise Features & Similarity

- For each pair *i* and *j* in an item set, there is a pairwise feature vector ϕ_{ij}.



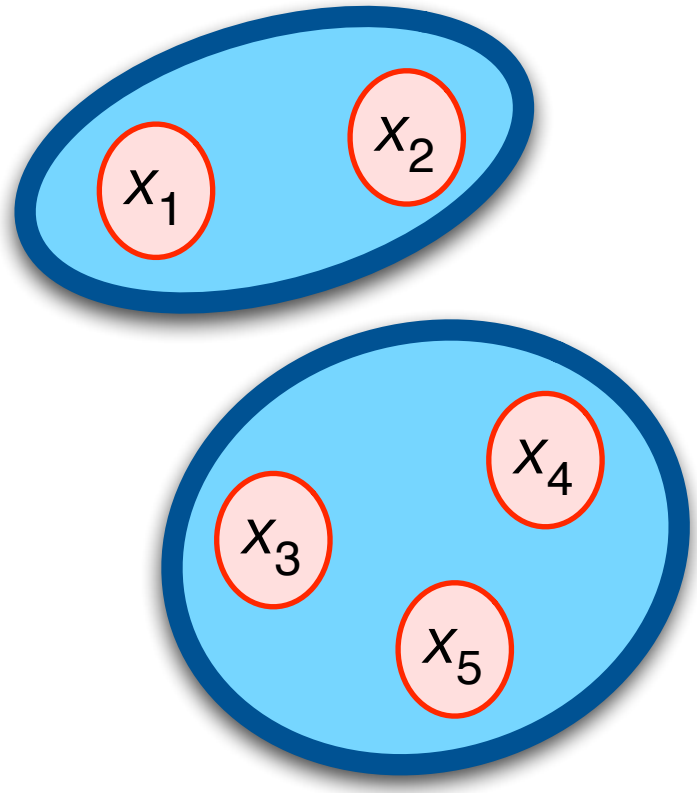


- Similarity of *i* and *j* is an inner product of ϕ_{ij} and a learned vector **w**.

$$Sim(x_i, x_j) = \langle \mathbf{w}, \phi_{ij} \rangle$$

Naïve Training Example

- Set **x** with partitioning **y**. Learn by simple classifier (Ng, Cardie, 2002).
- Positive examples: $\phi_{12}, \phi_{34}, \phi_{35}, \phi_{45}$.
- Negative examples: $\phi_{13}, \phi_{14}, \phi_{15}, \phi_{23}, \phi_{24}, \phi_{25}$.
- Linear SVM trained on these will learn a weight vector **w**.



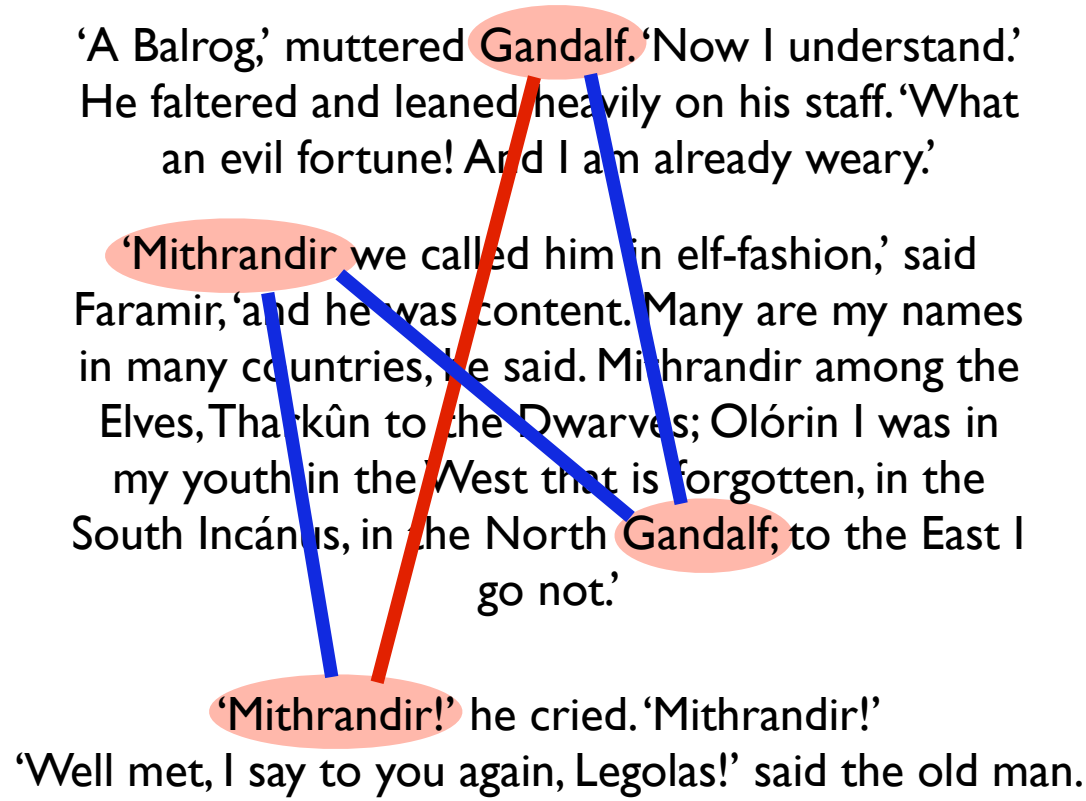
$$Sim(x_i, x_j) = \langle \mathbf{w}, \phi_{ij} \rangle$$

Problem 1: Hard Coded Performance Measure

- Application needs different performance measure, e.g., MITRE F-measure for NP coreference.
- Imbalanced positive/negative ratio.

Problem 2: Clustering Interactions

- Consider the NP coreference problem.
- Can a classifier learn considering pairs like these in isolation?
- Perhaps can learn with indirect dependencies.



Supervised Clustering Talk Outline

- Supervised clustering motivation
- Problem statement, difficulties
- Learning to cluster with SVM^{struct}
- Application to real problems

SVM^{struct} Overview

- SVM^{struct}: adaptation of SVM^{light} for learning functions with complex output space. (Tsochantaridis, et al. 2004)
- Can learn functions from **x** to **y** of the form

$$f(\mathbf{x};\mathbf{w}) = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}} \langle \mathbf{w}, \Psi(\mathbf{x},\mathbf{y}) \rangle$$

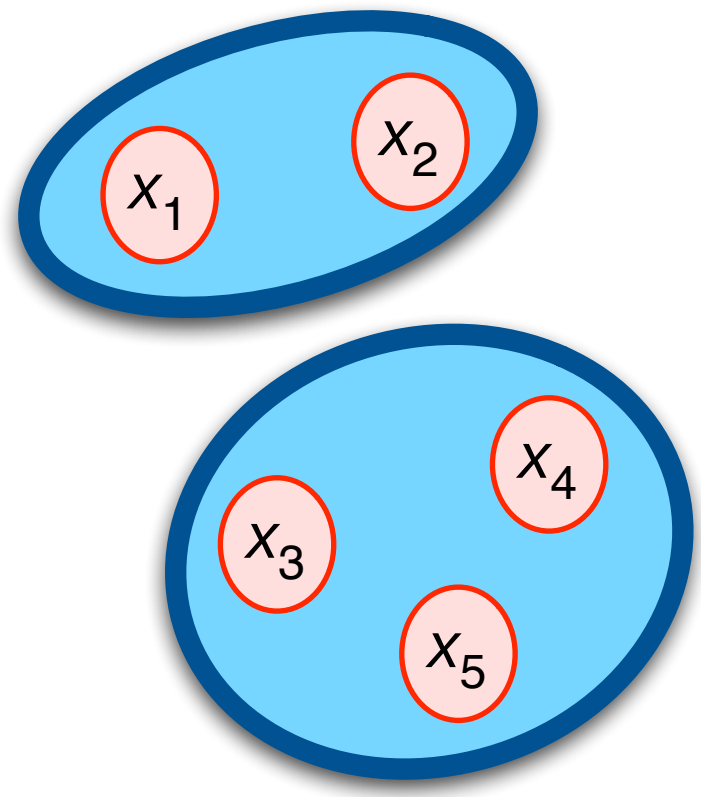
- $\Psi(\mathbf{x},\mathbf{y})$ characterizes relationship between input **x** and output **y**.

Ψ for Clustering

- Let Ψ sum of ϕ_{ij} for all *i,j* in **x** in the same cluster in **y** divided by square of number of items, e.g.:

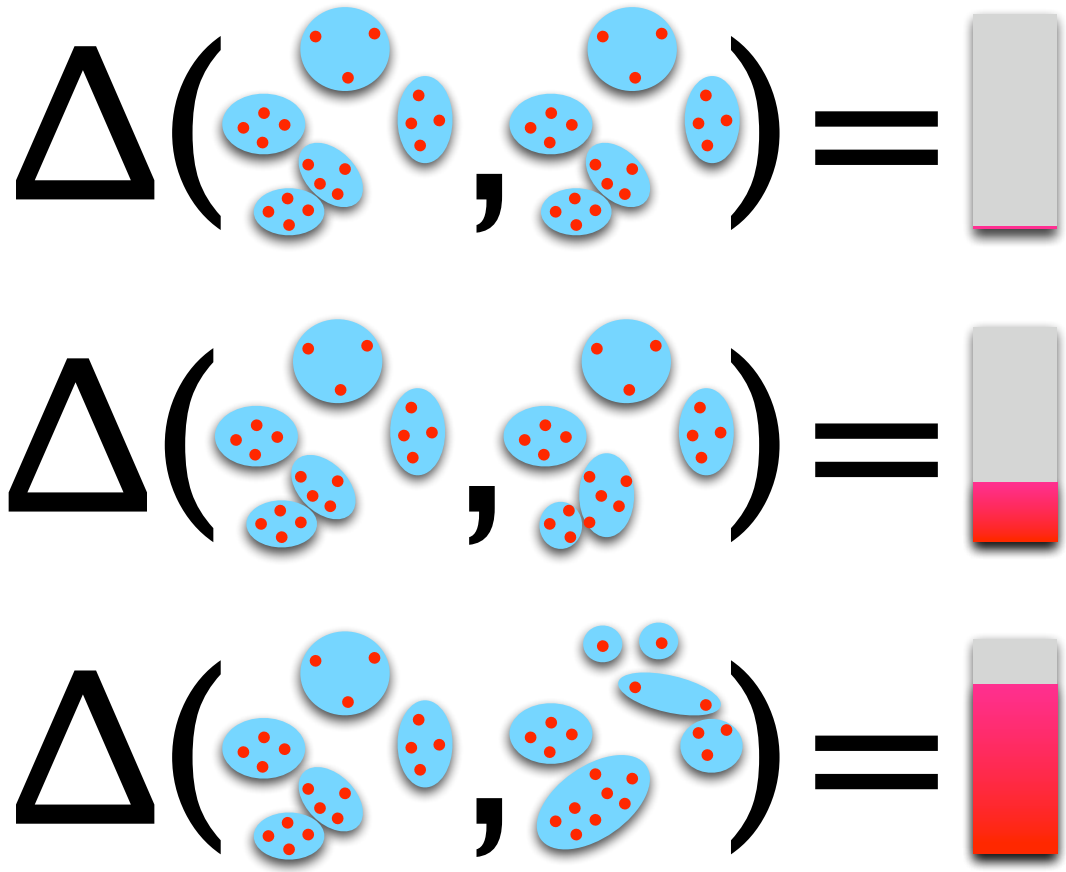
$$\Psi(\mathbf{x},\mathbf{y}) = \frac{1}{5^2}\left(\phi_{1,2} + \phi_{3,4} + \phi_{3,5} + \phi_{4,5}\right)$$

- This $\langle \mathbf{w}, \Psi(\mathbf{x},\mathbf{y}) \rangle$ equivalent to correlation clustering objective!



Δ for Clustering

- Measures how unrelated two clusterings are.
- Pairwise loss. Over pairs, get proportion of disagreement w.r.t. cluster relationships.
- MITRE loss, specific to NP coreference.



Linear Constraint

$$\forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_i\} : \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}_i) \rangle \geq \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}) \rangle + \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i$$

- For all training examples, and for all possible wrong clusterings, keep the value of the objective function for the correct clustering greater than the value of the objective function for every wrong clustering, and make sure they differ by at least the loss between the right and wrong clustering. We allow slack as an upper bound on loss.

Quadratic Program Formulation

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$\text{s.t. } \forall i : \xi_i \geq 0$$

$$\forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_i\} : \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i$$

- Can't really introduce a constraint for every wrong clustering for every example!
- Instead, find a "select few" constraints, and introduce those! (Tsochantaridis, et al. 2004)

Algorithm to Select Constraints

- Iteratively find clustering $\hat{\mathbf{y}}$ associated with the most violated constraint.
- Ignore the constant, and this is our clustering objective plus the loss.
- We can find $\hat{\mathbf{y}}$ for this argmax with a clustering function variant.

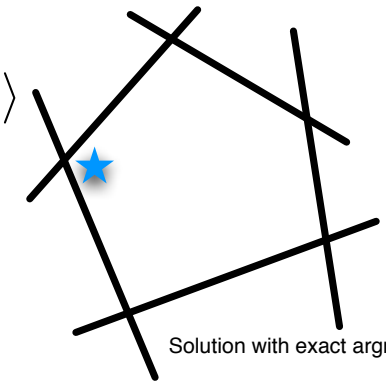
1: Input: $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$, C, ϵ
2: $S_i \leftarrow \emptyset$ for all $i = 1, \ldots, n$
3: **repeat**
4: **for** $i = 1, \ldots, n$ **do**
5: $H(\mathbf{y}) \equiv \Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w}^T \Psi(\mathbf{x}_i, \mathbf{y}) - \mathbf{w}^T \Psi(\mathbf{x}_i, \mathbf{y}_i)$
6: compute $\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y})$
7: compute $\xi_i = \max\{0, \max_{\mathbf{y} \in S_i} H(\mathbf{y})\}$
8: **if** $H(\hat{\mathbf{y}}) > \xi_i + \epsilon$ **then**
9: $S_i \leftarrow S_i \cup \{\hat{\mathbf{y}}\}$
10: $\mathbf{w} \leftarrow$ optimize primal over $S = \bigcup_i S_I$
11: **end if**
12: **end for**
13: **until** no S_i has changed during iteration

Computing the Argmax

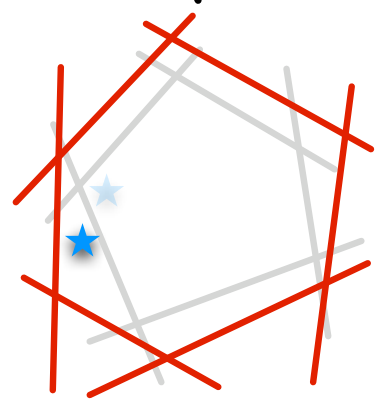
$$H(\mathbf{y}) = \Delta(\mathbf{y}_i, \mathbf{y}) + \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}) \rangle - \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}_i) \rangle$$

compute $\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y})$

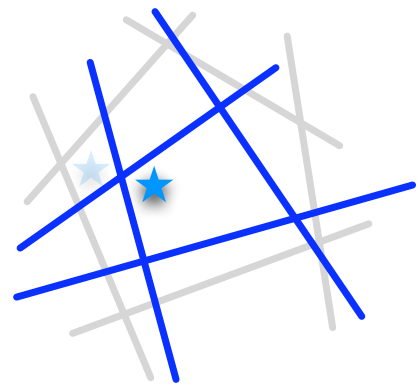
- **Exact argmax** - impractical and must approximate.
- **Greedy Clustering** - Get a lower bounded argmax approx, underconstrained QP.
- **Real relaxation** - (Demaine, Immorlica, 2003), get upper bounded argmax approx, overconstrained QP.



Solution with exact argmaxes.



Greedy approximate argmaxes, underconstrained.

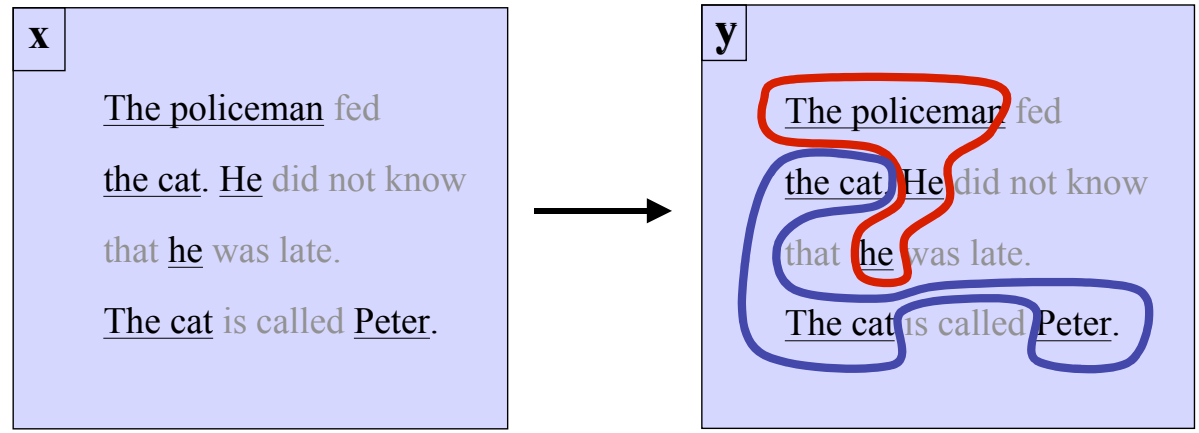


Real relaxation approximate argmaxes, overconstrained.

Supervised Clustering Talk Outline

- Supervised clustering motivation
- Problem statement, difficulties
- Learning to cluster with SVM^{struct}
- Application to real problems

NP Coreference



- **Data**
  - MUC-6 task.
  - Features supplied by Ng and Cardie.
- **Approaches**
  - SVM^{light} trained on all pairs, then greedy correlation clustering, denoted **PCC** (pairwise classifier clustering).
  - SVM^{struct} with greedy correlation clustering.

News Story Clustering

- Trawled Google News to build our own dataset.
- Each day for 30 days, select at most 15 news articles from at most 70 related stories -- usually ~900 articles/day.
- Cluster ~900 articles in a day. The 70 related story groups formed the clusters.
- Train on first 15 day sets, test on last 15 day sets.

Building the pairwise feature vector ϕ

- 31 features of the following kind:
  1: cos sim. of unigrams in title
  2: cos sim. of bigrams in title
  3: cos sim. of trigrams in title
  4: cos sim. of unigrams in headline
  ⋮
  30: cos sim. of porter stemmed trigrams in quoted article text
  31: always 1!

SVM^{cluster} vs. PCC

- Entries are errors, either MITRE loss Δ_M or pairwise loss Δ_P, "Default" is "worst" error.
- Header training method. Left column testing clustering method and error rate.
- NP Coref: SVM^{cluster} significantly better for both MITRE and Pairwise loss training.
- News: SVM^{cluster} and PCC do not differ significantly.

Noun Phrase	SVM^{cluster} C_G	PCC	Def.
Test with C_G, Δ_M	41.3	51.6	51.0
Test with C_G, Δ_P	2.89	3.15	3.59

News	SVM^{cluster} C_G	SVM^{cluster} C_R	PCC	Def.
Test with C_G, Δ_P	2.36	2.43	2.45	9.45
Test with C_R^*, Δ_P	2.04	2.08	1.96	9.45

Optimizing to Right Δ

- SVM^{cluster} can optimize to an arbitrary clustering loss. Does this matter?

- Perverse test: Train and optimize model to one loss. Evaluate its performance with another loss.

- Not significantly different for evaluation on Δ_M, quite different for Δ_P. Even worse than default!!

Noun Phrase	*Opt. to* Δ_M	*Opt. to* Δ_P
Test on Δ_M	41.3	42.8
Test on Δ_P	4.06	2.89

Inclusion of Δ in Finding Constraint

- Recall portion of algorithm where the $\hat{\mathbf{y}}$ associated with most violated constraint is computed.

$$H(\mathbf{y}) \equiv \Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w}^T \Psi(\mathbf{x}_i, \mathbf{y}) - \mathbf{w}^T \Psi(\mathbf{x}_i, \mathbf{y}_i)$$
$$\text{compute } \hat{\mathbf{y}} = \text{argmax}_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y})$$

- What if we drop $\Delta(\mathbf{y}_i, \mathbf{y})$ from the cost function, and just maximize the objective?

- Note: when we introduce constraint, we still include $\Delta(\mathbf{y}_i, \hat{\mathbf{y}})$ in constraint!

- Never significant. Good news!

	with loss	*no loss*
NP-coreference, Δ_M	41.3	41.1
NP-coreference, Δ_P	2.89	2.81
News, train C_G, test C_G	2.36	2.42
News, train C_R, test C_R*	2.08	2.16

Real Relaxation versus Greedy Clustering

- What about underconstrained (with greedy C_G) versus overconstrained (with relaxed C_R) for training?
- Neither is significantly different from the other.

News	Train C_G	Train C_R
Test C_G	2.36	2.43
Test C_R^*	2.04	2.08

Conclusions

- **Content**
  - Adapted SVM^{struct} to clustering.
  - Advantages more obvious on problems with complex interactions among objects, or that use special performance measures.
- **Future Work**
  - Application to other problems.
  - Extend to semi-supervised clustering.