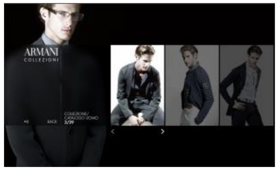
Robust Hierarchical Clustering

Maria-Florina Balcan
Georgia Institute of Technology

Joint work with Pramod Gupta

Clustering comes up everywhere

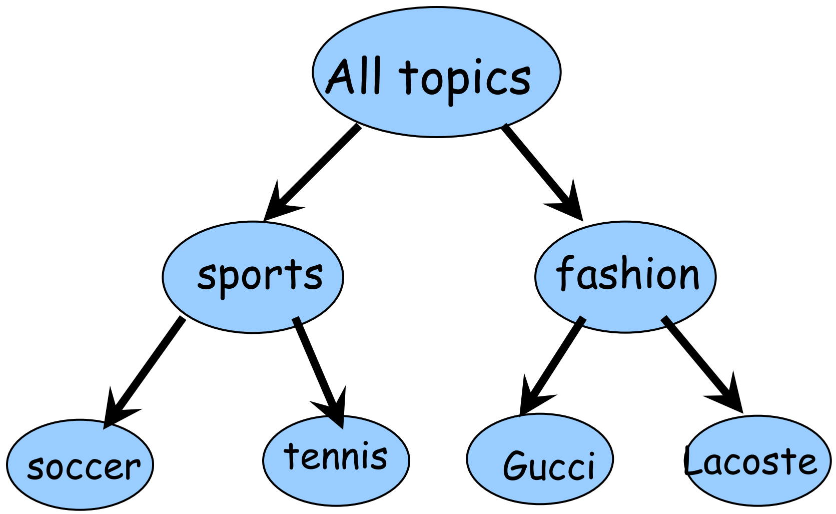
- Cluster news articles or web pages by topic



- Cluster protein sequences by function

Linkage Based Procedures

S set of n objects. [documents, web pages, protein seq.]



Have a similarity measure on pairs of objects.

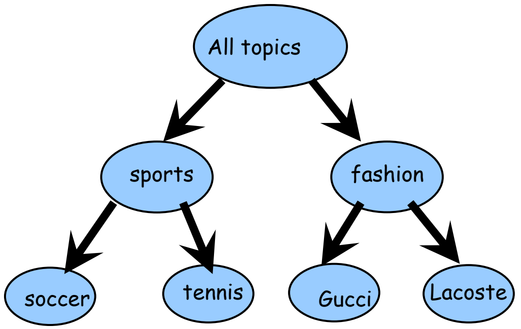
K(x,y) – similarity between x and y

E.g., # keywords in common, edit distance, wavelets coef., etc.

Classic Approach: Linkage Procedures

Have a similarity measure on pairs of objects.

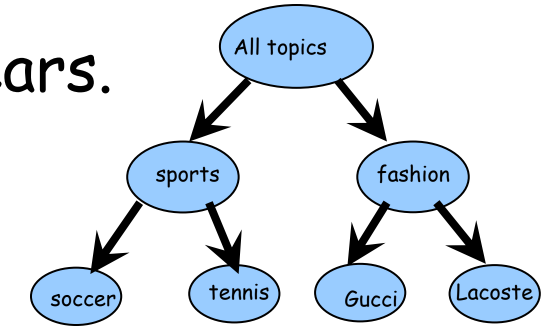
$K(x,y)$ – similarity between x and y



- Single linkage: $K(A,B) = \max_{x \text{ in } A,\, y \text{ in } B} K(x,y)$
- Average linkage: $K(A,B) = \text{avg}_{x \text{ in } A,\, y \text{ in } B} K(x,y)$
- Complete linkage: $K(A,B) = \min_{x \text{ in } A,\, y \text{ in } B} K(x,y)$

Linkage Procedures

Widely used across science for many years.



Simple, fast, easy to interpret.

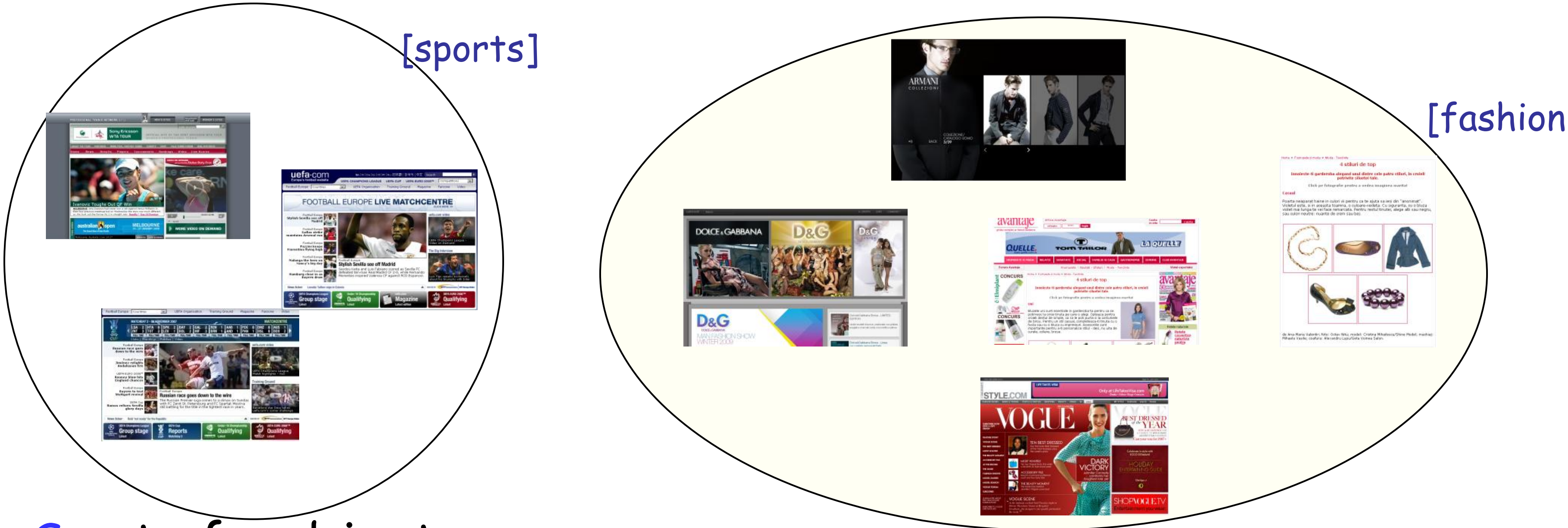
Problem: not robust to noise.

[e.g., Bilmes et. al, NIPS 2005]

Question: Can we design robust linkage procedures that are tolerant to noisy and incomplete similarity info?

We provide a robust aglomerative clustering algorithm.

Clustering: Formal Setup [BBV]



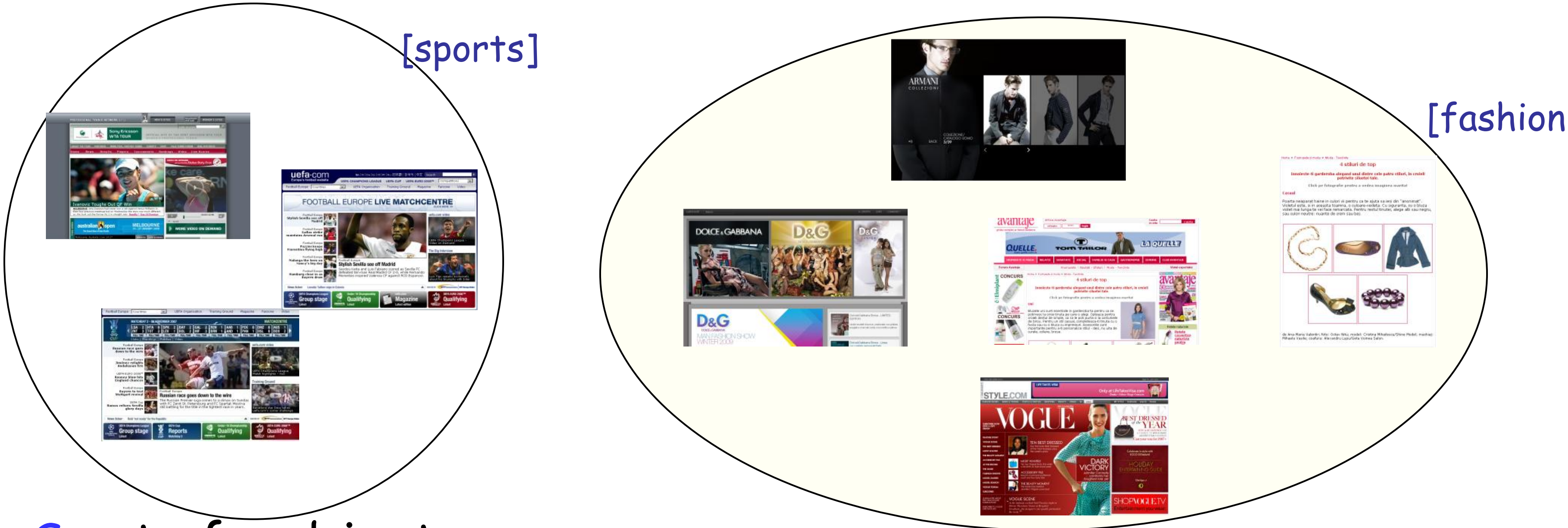
S set of n objects. [documents, web pages]

$\exists$ "ground truth" clustering $C_1, C_2, \ldots, C_k$. [true clustering by topic]

The error of clustering $C'_1, \ldots, C'_k$ is: $\text{error}(C') = \min_{\sigma \in S_k} \frac{1}{n} \sum_{i=1}^{k} |C_i - C'_{\sigma(i)}|$

error(C') = fraction of pts misclassified up to re-indexing of clusters

Clustering: Formal Setup [BBV]



S set of n objects. [documents, web pages]

$\exists$ "ground truth" clustering $C_1, C_2, \ldots, C_k$. [true clustering by topic]

The error of clustering $C'_1,\ldots,C'_k$ is: $$\text{error}(C') = \min_{\sigma \in S_k} \frac{1}{n} \sum_{i=1}^{k} |C_i - C'_{\sigma(i)}|$$

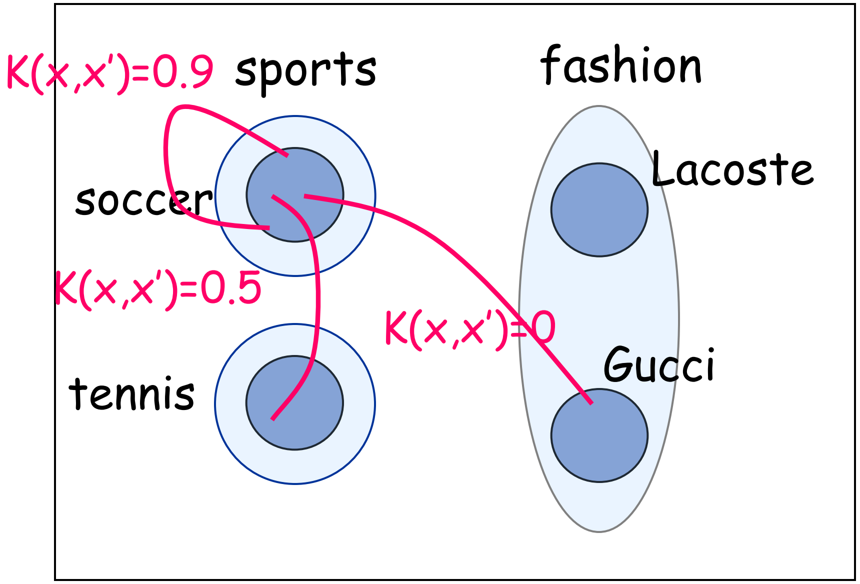
We are given a pairwise similarity function K.

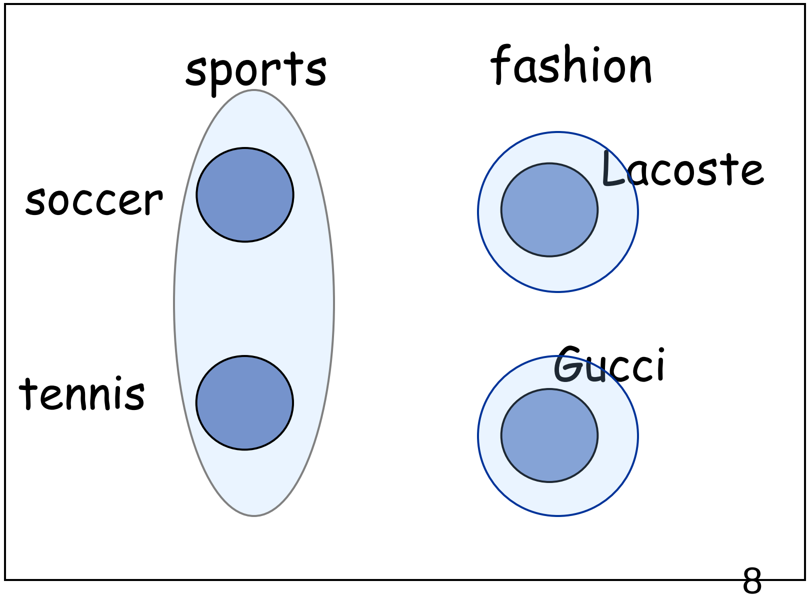
Goal: Produce hierarchy that has pruning of small error

Clustering: Formal Setup [BBV]

All x more similar to all y in own cluster than any z in any other cluster [strict separation property]

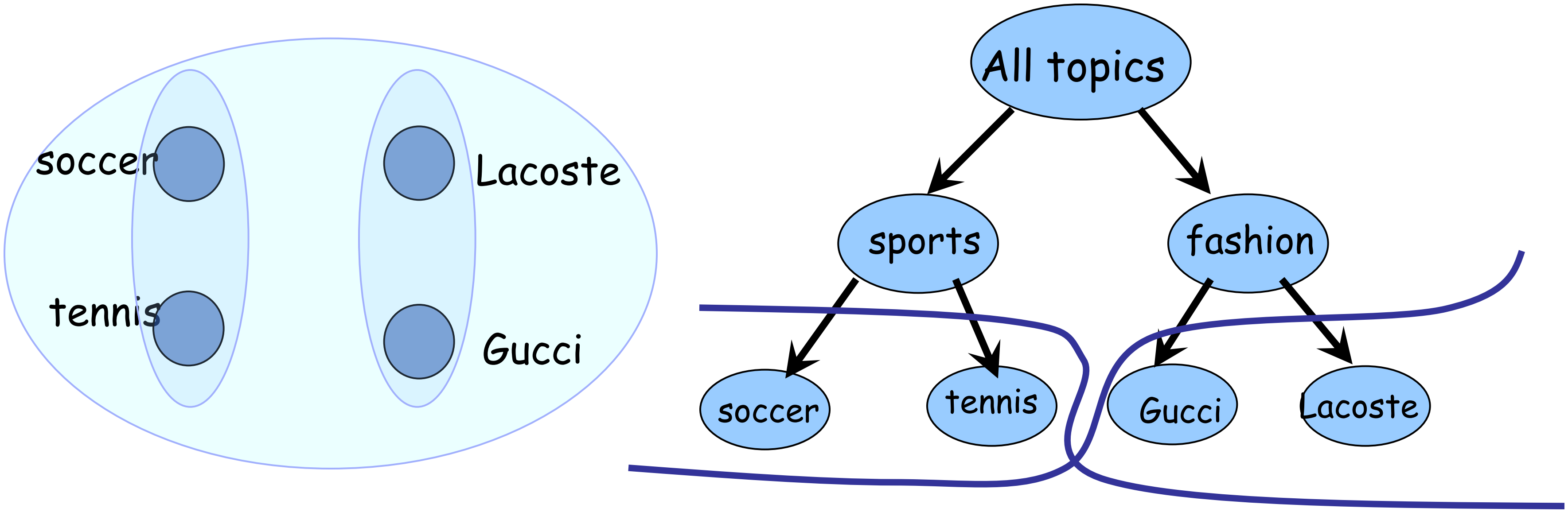
Note: same K can satisfy it for two very different, equally natural clusterings of the same data; so we can't hope to output a partition of the data.





Clustering: Formal Setup [BBV]

Produce a hierarchical clustering s.t. correct answer is approximately some pruning of it.



The user can then navigate it to determine his desired clustering.

Strict Separation Property

All x more similar to all y in own cluster than any z in any other cluster

Theorem Use Single-Linkage, construct a tree s.t. ground-truth clustering is a pruning of the tree.

Alternatively, for exploratory clustering:

Theorem All clusterings satisfying strict separation are prunnings of the tree.

All clusters satisfying strict separation are in the tree.

Good neighborhood property

α-good neighborhood property

For all points x, all but αn out of their $n_{C(x)}$ nearest neighbors belong to the cluster $C(x)$.

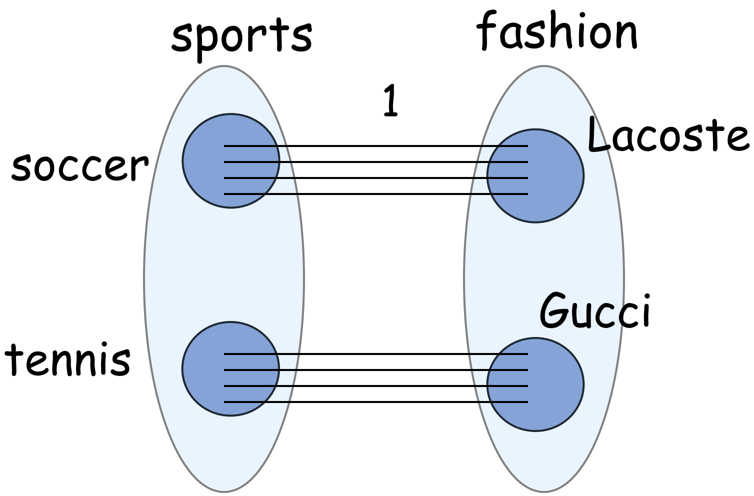
Note: strict separation is equivalent to 0-good neighborhood.

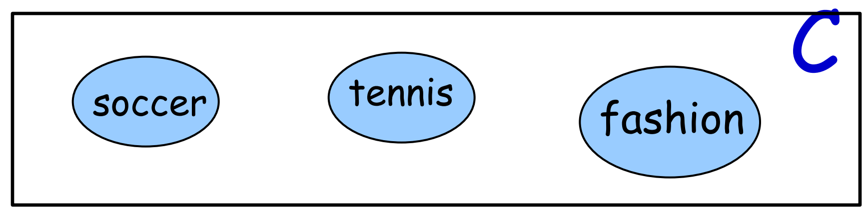
(α,ν)-good neighborhood property

For some $S' \subseteq S$ of size $(1-\nu)n$, K satisfies α-good neighborhood property on the instance induced by S'.

Notation: Points in S' are called good points. The rest, bad points.

Standard linkage algos fail even with α = 1/n





- (α,0)-good neighborhood, α=1/n.

- Not even ½ close to prunnings of tree produced by SL, AL, CL.

Can we design an alg that succeeds under the good neighborhood property?

A New Robust Hierarchical Clustering Algo

Our contribution: an efficient alg. for the good neighborhood property.

Theorem (Main Result)

K symmetric fnc satisfying (α,ν) good neighborhood.

If target clusters large (of size $\Omega((\alpha+\nu)n)$), then can produce a tree s.t. the target is ν-close to one of the prunings of the tree.

A New Robust Hierarchical Clustering Algo

Phase I: Generate list L of interesting blobs.

[not too large, not too small, almost pure]

Phase II: Run a robust linkage procedure on L to generate a hierarchy T.

Note: Robust in two ways

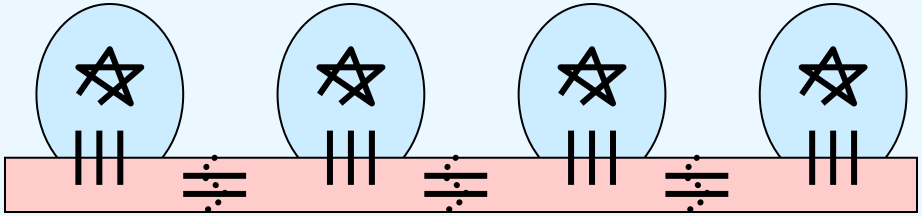
- Use global info to produce blobs.
- Use median info to link large enough blobs.

Use of blobs and median lends robustness since noisy similarities are outvoted.

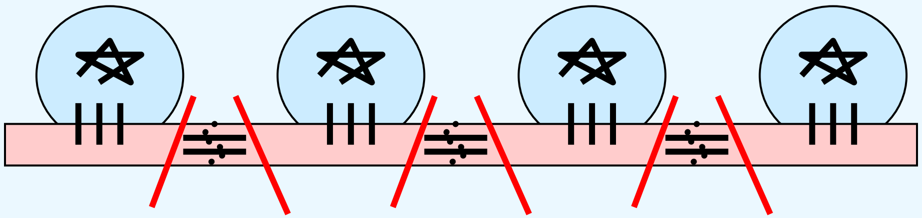
Phase I: Generate Interesting Blobs

Initial threshold $t = 6(v + \alpha)n + 1$, $L = \emptyset$, $A_S = S$

- Graph F_t : connect x & y in A_S if share $\geq t - 2(v + \alpha)n$ points in common out of their t nearest neighbors w.r.t. S.



- Graph H_t: connect x & y if share at least $3(v + \alpha)n$ neighbors in the graph F_t.



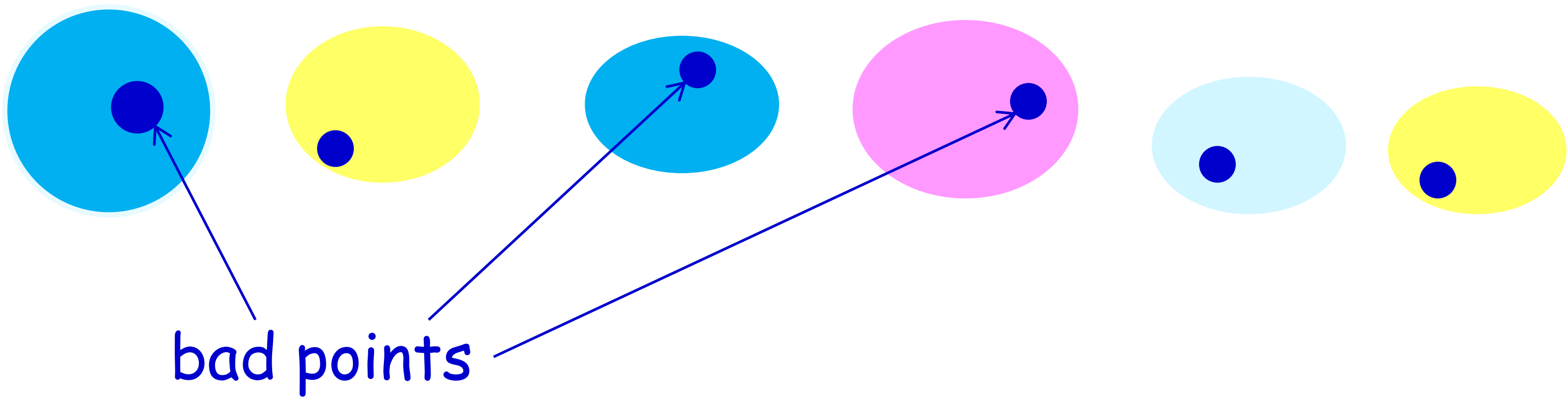
- Add to L all comp. C of H_t with $|C| \geq 3(v + \alpha)n$ and remove C from A_S.

- For each point $x \in A_S$, if $(v + \alpha)n$ out of its $5(v + \alpha)n$ nearest neighbors are in L, then assign x to a blob of highest median in L. Remove x from A_S.

- While $|A_s| \geq 3(v + \alpha)n$ and $t < n$, $t = t + 1$ and repeat.

Generate Interesting Blobs

Claim: The blobs in L form a partition of S, each blob is large enough, and it has good points from only one target cluster.

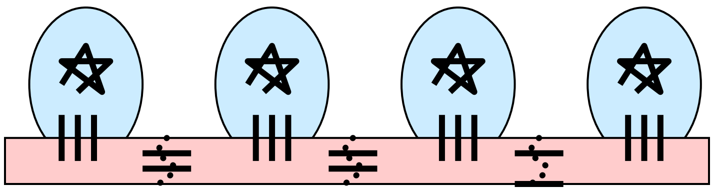


Generate Interesting Blobs

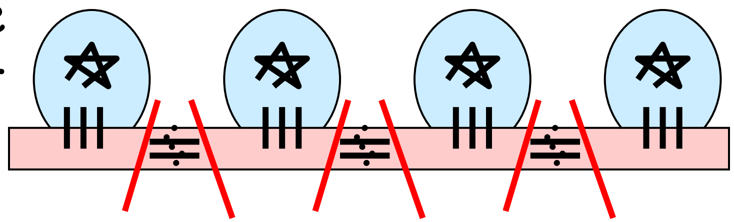
Claim: The blobs in L form a partition of S, each blob is large enough, and it has good points from only one target cluster.

Proof Idea: Assume all clusters have the same size, n_C

- At $t \leq n_C$, no two good points in two different clusters connected in F_t, a bad point only connected to a good set.



- At $t=n_C$, all good points in the same clusters forms cliques in F_t, a bad point only connected to a good set.



All components of H_t represent good blobs.

- We do not know n_C, but safe to start low and keep increasing it, and output large enough components.

Generate Interesting Blobs

Claim: Each blob in L is large enough and it has good points from only one target cluster.

Proof Idea:

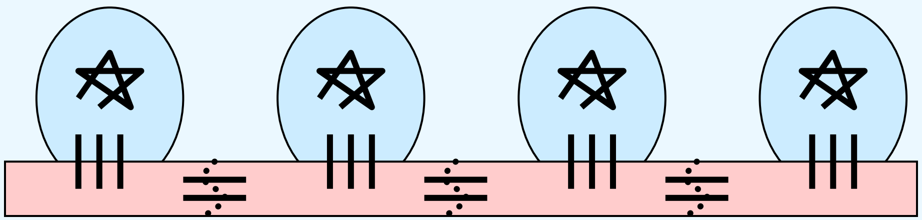
In general, different sizes

- assume $n_{C_1} \leq n_{C_2} \leq \ldots \leq n_{C_k}$
- we make sure by the time we pass n_{C_1} we have pulled out all good points in n_{C_1}; in general, by the time we passed n_{C_i} we have pulled all good points in clusters $1, 2, \ldots, i$.

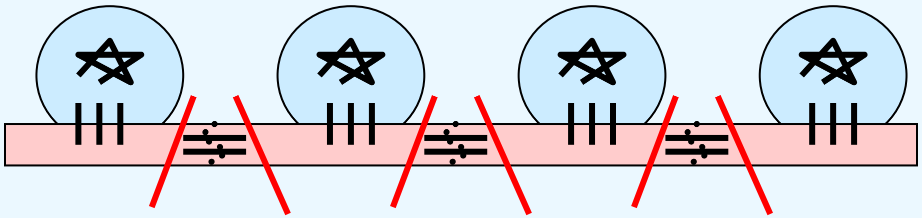
Generate Interesting Blobs

Initial threshold $t = 6(v + \alpha)n + 1$, $L = \emptyset$, $A_S = S$

- Graph F_t : connect x & y in A_S if share $\geq t - 2(v + \alpha)n$ points in common out of their t nearest neighbors w.r.t. S.



- Graph H_t: connect x & y if share at least $3(v + \alpha)n$ neighbors in the graph F_t.



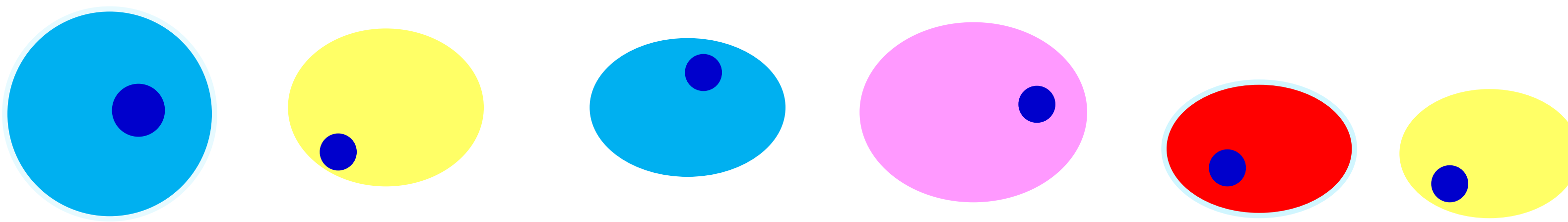
- Add to L all comp. C of H_t with $|C| \geq 3(v + \alpha)n$ and remove C from A_S.

- For each $x \in A_S$, if $(v + \alpha)n$ out of its $5(v + \alpha)n$ nearest neighbors are in L, then assign x to a blob of highest median in L. Remove x from A_S.

- While $|A_s| \geq 3(v + \alpha)n$ and $t < n$, $t = t + 1$ and repeat.

Robust Linkage Procedure

So, after the first phase:



Second phase:

- Find C, C' in the current list L which maximize score(C,C').
- Remove C and C' from L, merge them into C''; add C'' to L.
- Repeat till only one cluster remains in L.

Key point: define score s.t.:

$$\text{score}(A_i, A_j) > \text{score}(A_i, A_k)$$

$$\text{score}(A_i, A_j) > \text{score}(A_j, A_k)$$

Conclusions and Open Questions

A new robust algorithm for hierarchical clustering.

- provably works under the good neighborhood property.
- good neighborhood property is a “noisy” relaxation of the strict separation property.
- classic linkage algorithms (SL or AL) succeed under strict separation but fail badly under good neighborhood.

Open Questions

- Analyze other properties under which this alg succeeds.
- Design alg for a robust version of max stability, known to be a necessary and sufficient condition for single linkage.