Clustering with Constraints

Sugato Basu
SRI Intl. (AI Center)
Email: basu@ai.sri.com

Ian Davidson
SUNY – Albany (Dept. of Computer Science)
Email: davidson@cs.albany.edu

Acknowledgements

- Contribution of slides
  - Tomer Hertz
  - Sepandar Kamvar
  - Brian Kulis
- Insightful discussions
  - S.S. Ravi
  - Kiri Wagstaff
- Apologies
  - If we do not get around to covering your work or if you have work on constraints and clustering and we didn't include it in the bibliography (drop us an email).

Notation

- *S* : set of training data
- s_i : i^{th} point in the training set
- *L*: cluster labels on S
- l_i : cluster label of s_i
- C_j: centroid of j^{th} cluster
- *ML* : set of must-link constraints
- *CL* : set of cannot-link constraints
- CC_i : a connected component (sub-graph)
- *TC* : the transitive closure

Outline

- Introduction [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits of constraints [Ian]
- Feasibility and complexity [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Outline

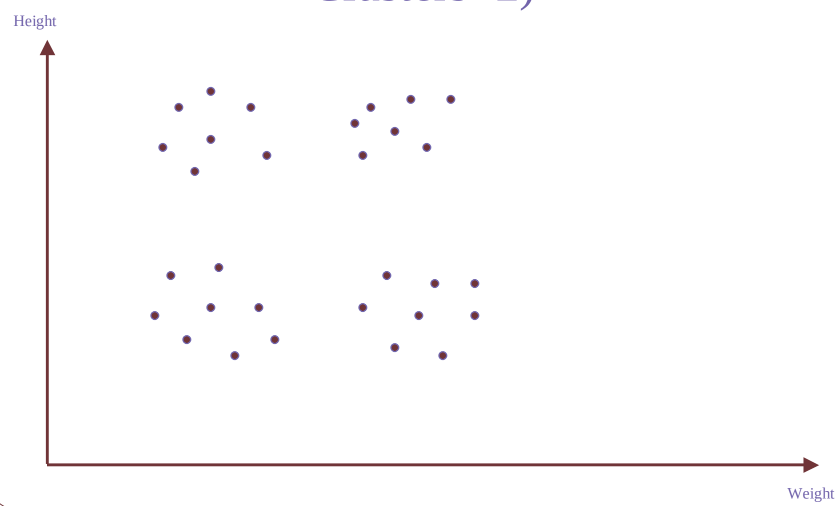
- Introduction [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits of constraints [Ian]
- Feasibility and complexity [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

A Motivating Example in Non-Hierarchical Clustering

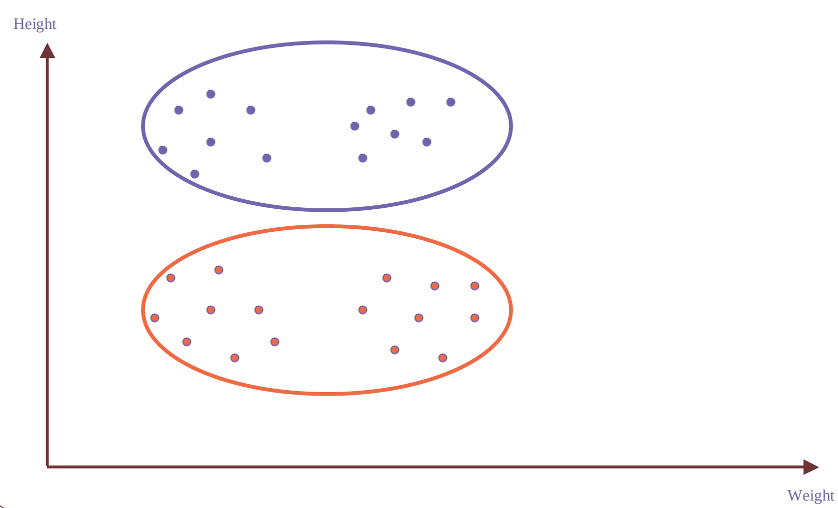
- Given a set of instances S
- Find the “best” set partition

  $S = \{S_1 \cup S_2 \cup \ldots S_k\}$
- Multitude of algorithms that define “best” differently
  - K-Means
  - Mixture Models
  - Self Organized Maps
- Aim is to find the **underlying** structure/patterns/groups in the data.

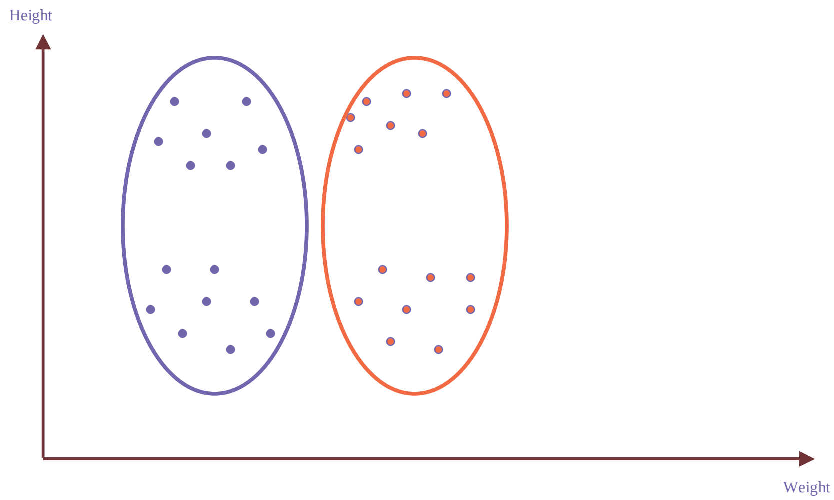
Clustering Example (Number of Clusters=2)



Horizontal Clusters



Vertical Clusters



K-Means Clustering

- Standard iterative partitional clustering algorithm
- Finds *k* representative centroids in the dataset
  - Locally minimizes the sum of distance (e.g., squared Euclidean distance) between the data points and their corresponding cluster centroids

$$\sum_{s_i \in S} D(s_i, C_{l_i})$$

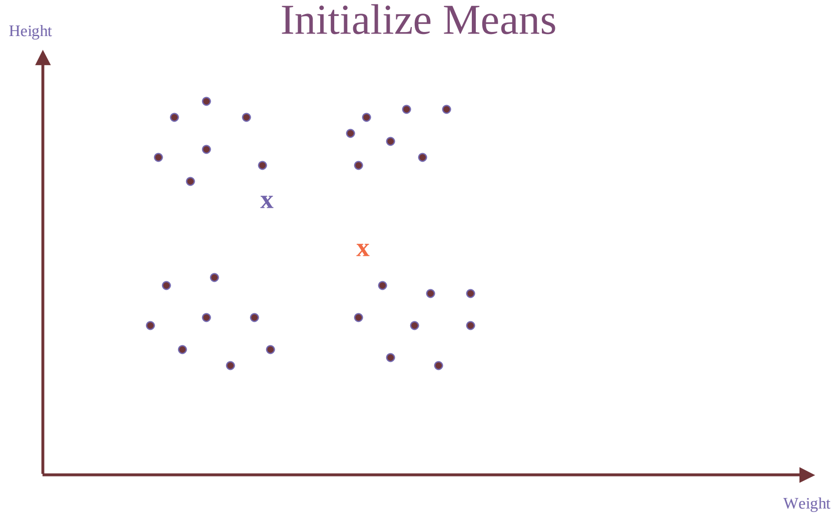
A Simplified Form of this Problem is intractable [Garey et al.'82]

K-Means Algorithm

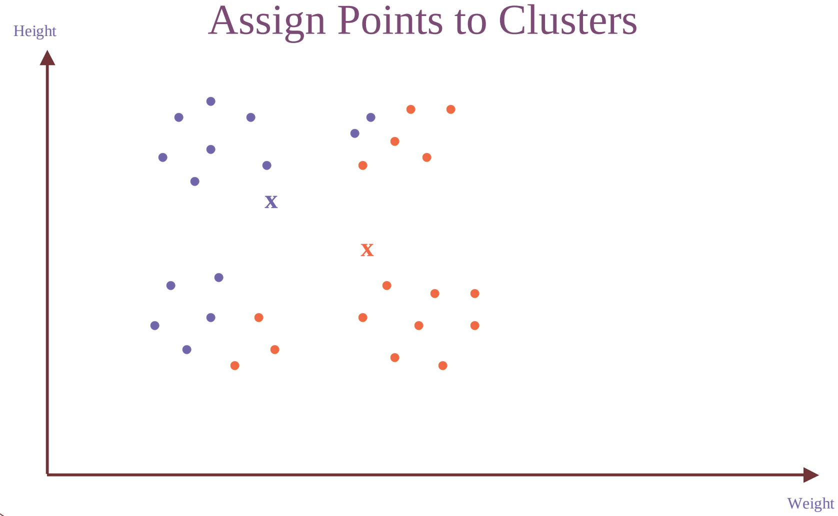
1. Randomly assign each instance to a cluster
2. Calculate the centroids for each cluster
3. For each instance
   - Calculate the distance to each cluster center
   - Assign the instance to the closest cluster
4. Goto 2 until distortion is small

K Means Example (k=2)

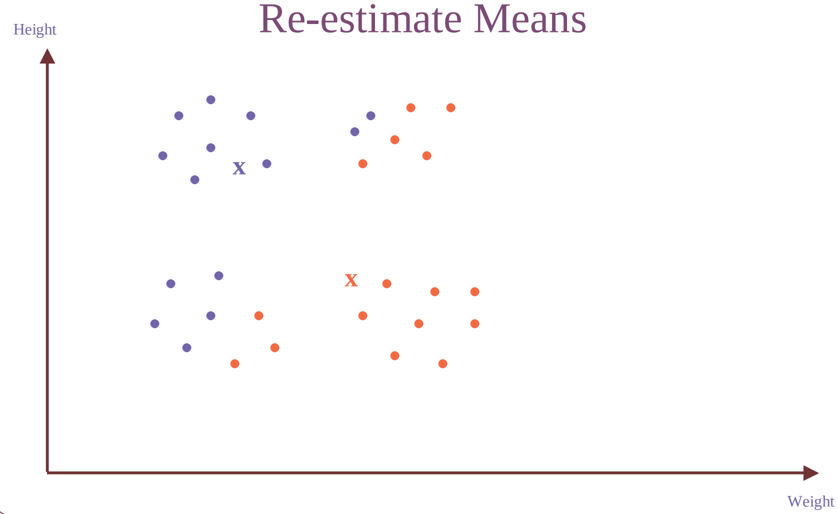
Initialize Means



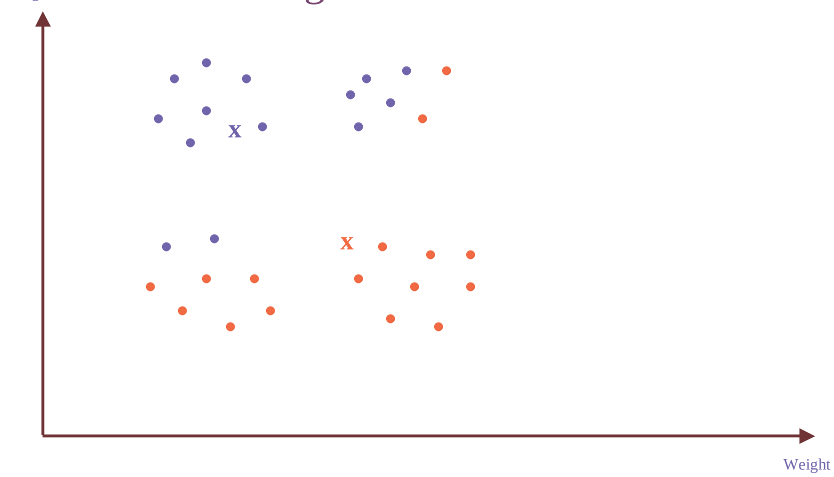
K Means Example
Assign Points to Clusters



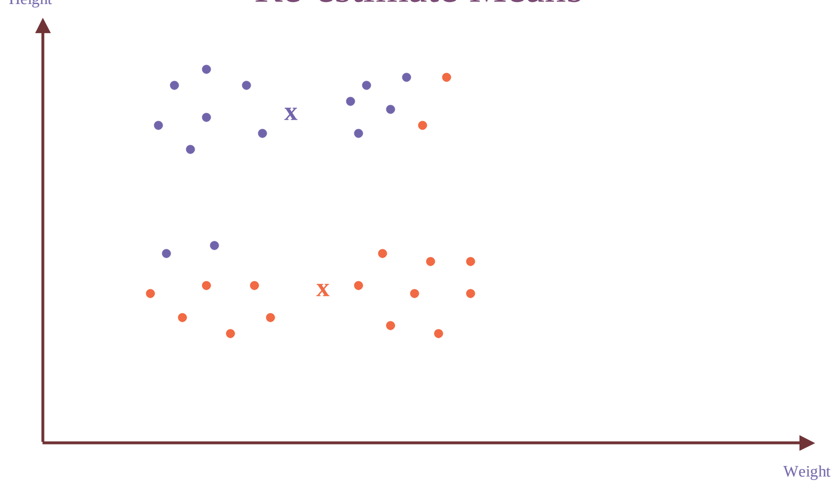
K Means Example
Re-estimate Means



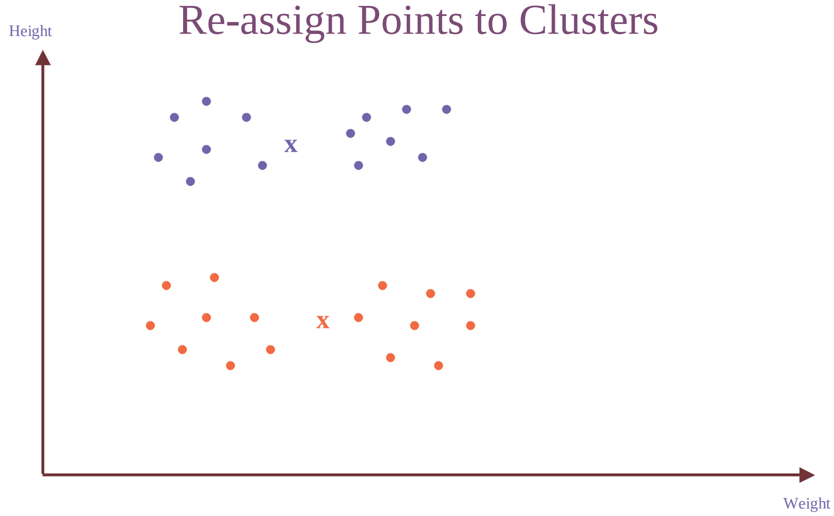
K Means Example
Re-assign Points to Clusters



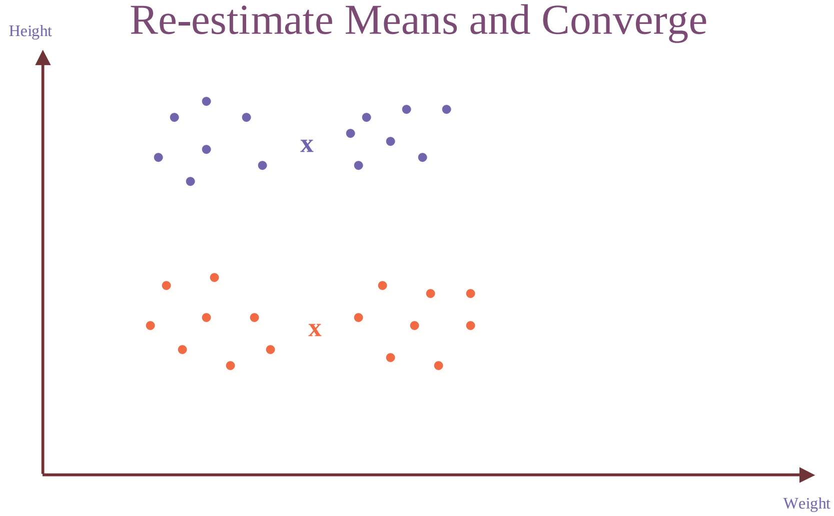
K Means Example
Re-estimate Means



K Means Example
Re-assign Points to Clusters

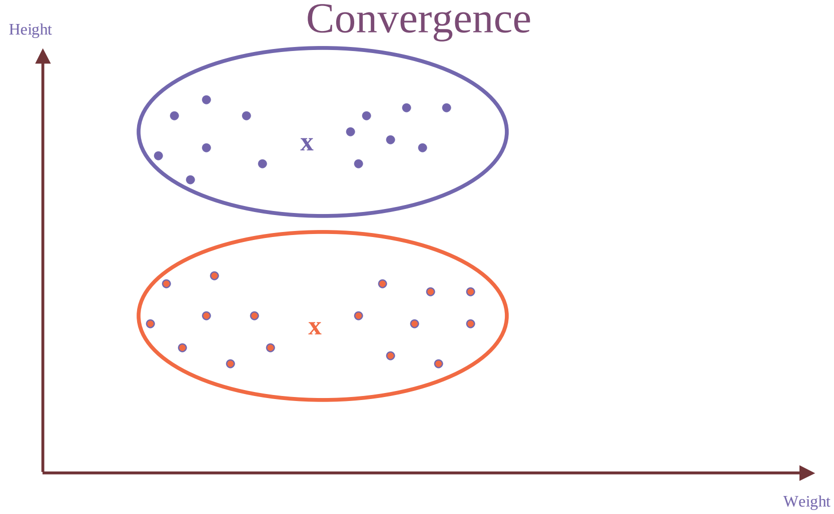


K Means Example
Re-estimate Means and Converge



K Means Example

Convergence



A Few Issues With K-Means Has Spawned Lots of Research

- Algorithm is typically restarted many times from random starting centroids
  - Due to sensitivity to initial centroids
    - i.e. Intelligently setting initial centroids [Bradley & Fayyad 2000]
- Convergence time of algorithm can be slow
  - Use KD-Trees to accelerate algorithms [Pelleg and Moore 1999]
- Clustering achieved may minimize VQE but has little practical value
- Which distance function should I use?
  - L1, L2, Mahalanobis etc.
- Constraints can help address these problems and more …

Automatic Lane Finding from GPS traces

[Wagstaff et al. ’01]

Lane-level navigation (e.g., advance notification for taking exits)

Lane-keeping suggestions (e.g., lane departure warning)

- **Constraints inferred from trace-contiguity (ML) & max-separation (CL)**

Mining GPS Traces (Schroedl et’ al)

- Instances are represented by the *x*, *y* location on the road. We also know when a car changes lane, but not what lane to.
- True clusters are very elongated and horizontally aligned with the lane central lines
- Regular k-means performs poorly on this problem instead finding spherical clusters.

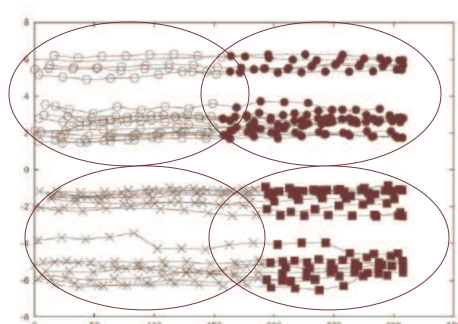
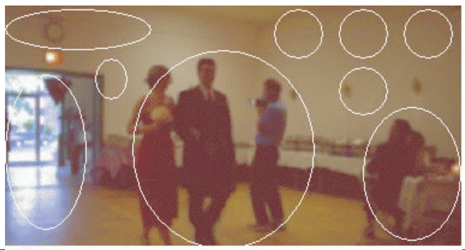


Figure 9. *k*-means output for data set 6, *k* = 4, with nearest clusters marked with different symbols.

Unconstrained K-Means Can Provide Not Useful Clusters

Object identification for Aibo robots



Only significant clusters shown

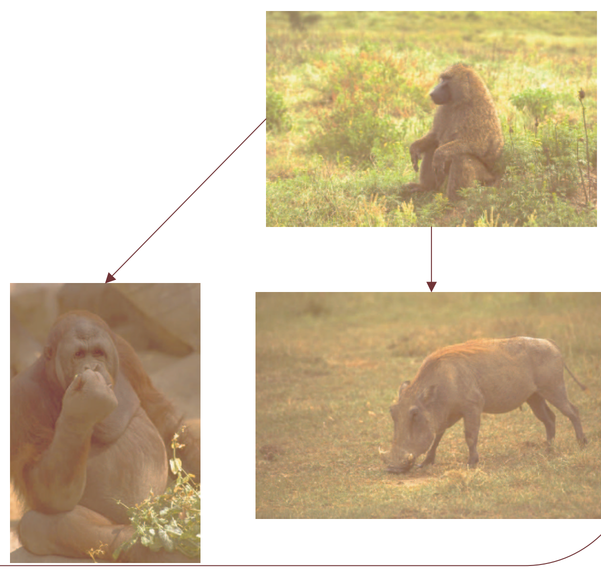
Why Learn Distance functions?

Nearest Neighbor

Image retrieval

Given a query image return the K-nearest neighbors of the image from the database.

Euclidean distance on Color Coherence Vectors returns both images as similar to query image



Basic Instance Level Constraints

- Historically, instance level constraints motivated by the availability of labeled data
  - i.e., Much unlabeled data and a little labeled data available generally as constraints, e.g., in web page clustering
- This knowledge can be encapsulated using instance level constraints [Wagstaff et al. '01]
  - Must-Link Constraints
    - A pair of points s_i and s_j $(i \neq j)$ must be assigned to the same cluster.
  - Cannot-Link Constraints
    - A pair of points s_i and s_j $(i \neq j)$ can not be assigned to the same cluster.

Properties of Instance Level Constraints

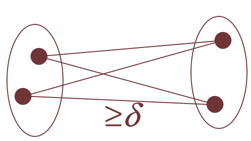
- Transitivity of Must-link Constraints
  - $ML(a,b)$ and $ML(b,c) \rightarrow ML(a,c)$
  - *Let X and Y be sets of ML constraints*
  - $ML(X)$ and $ML(Y)$, $a \in X$, $a \in Y$, $\rightarrow ML(X \cup Y)$
- The Entailment of Cannot link Constraints
  - $ML(a,b)$, $ML(c,d)$ and $CL(a,c) \rightarrow CL(a,d), CL(b,c), CL(b,d)$
  - *Let* $CC_1 \ldots CC_r$ be the groups of must-linked instances (i.e.. The connected components)
  - $CL(a \in CC_i, b \in CC_j) \rightarrow CL(x,y),\ \forall x \in CC_i,\ \forall y \in CC_j$

Complex Cluster Level Constraints

- δ-Constraint (Minimum Separation)
  - For any two clusters S_i, S_j $\forall$ i,j
  - For any two instances $s_p \in S_i$, $s_q \in S_j$ $\forall$ p,q
  - $D(s_p, s_q) \geq \delta$
- ε-Constraint
  - For any cluster S_i $|S_i| > 1$
  - $\forall p, s_p \in S_i, \exists s_q \in S_i : \varepsilon \geq D(s_p, s_q), s_p <> s_q$

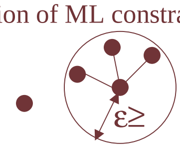
Converting Cluster Level to Instance Level Constraints

- Delta constraints?



For every point x, must-link all points y such that D(x,y) < δ, i.e. conjunction of ML constraints

- Epsilon constraints?
  - For every point x, must link to at least one point y such that D(x,y) <= ε, i.e. disjunction of ML constraints



- Will generate many instance level constraints

Other Constraint Types We Won't Have Time To Cover

- Balanced Clusters
  - Scalable model-based balanced clustering [Zhong et al. '03]
  - Frequency sensitive competitive learning [Galanopoulos et al. '96]
- Negative background information
  - Find another clustering that is quite different from a given set of clusterings [Gondek et al. '04]
- Clustering only with constraints
  - Use constraints to cluster the data, no underlying distance function
    - Correlation Clustering: [Bansal et al.'02]
    - Clustering with Qualitative Information: [Charikar et al. '03]

Outline

- Introduction [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits of constraints [Ian]
- Feasibility and complexity [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Big Picture

- Clustering with constraints:

  Partition unlabeled data into groups called clusters

  + use constraints to aid and bias clustering

- Goal:

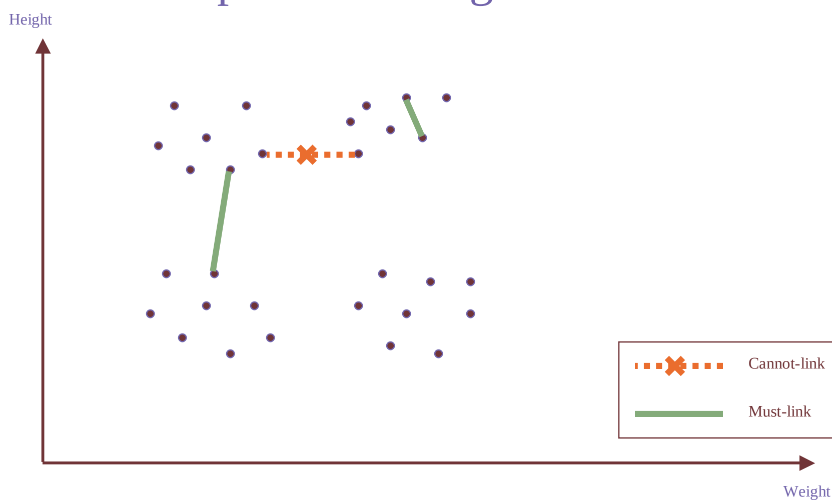
  Examples in same cluster similar, separate clusters different + constraints are maximally respected

Enforcing Constraints

- Clustering objective modified to enforce constraints
  - Strict enforcement: find "best" feasible clustering respecting all constraints
  - Partial enforcement: find "best" clustering maximally respecting constraints
- Uses standard distance functions for clustering

[Demiriz et al.'99, Wagstaff et al.'01, Segal et al.'03, Davidson et al.'05, Lange et al.'05]

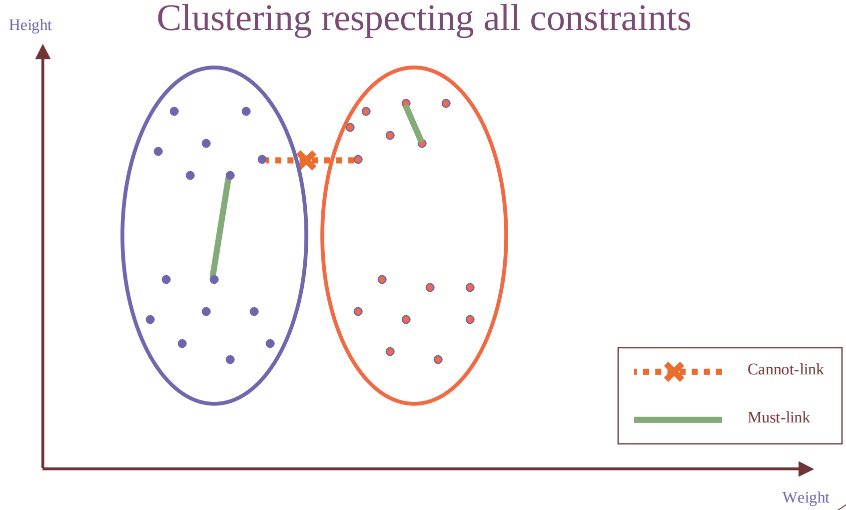
Example: Enforcing Constraints



© Basu and Davidson 2005 Clustering with Constraints 33

Example: Enforcing Constraints

Clustering respecting all constraints



© Basu and Davidson 2005 Clustering with Constraints 34

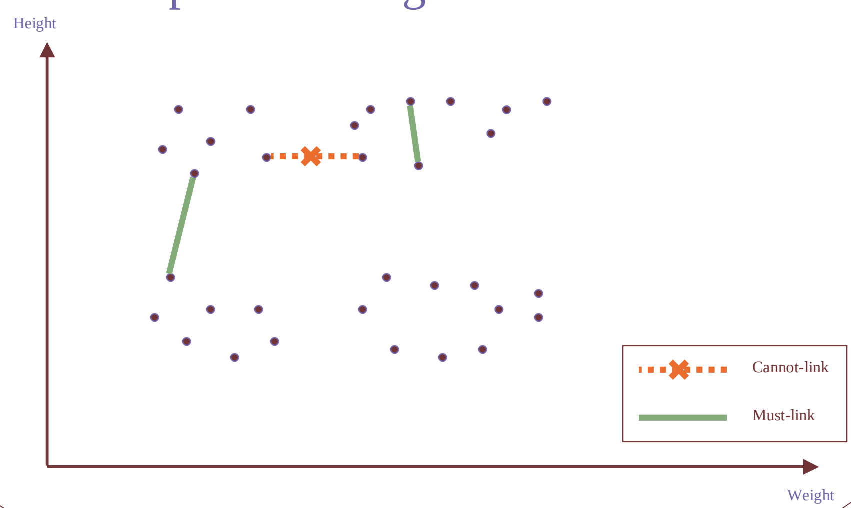
Learning Distance Function

- Constraints used to learn clustering distance function
  - *ML(a,b)* $\rightarrow$ *a* and *b* and surrounding points should be "close"
  - *CL(a,b)* $\rightarrow$ *a* and *b* and surrounding points should be "far apart"

- Standard clustering algorithm applied with learned distance function

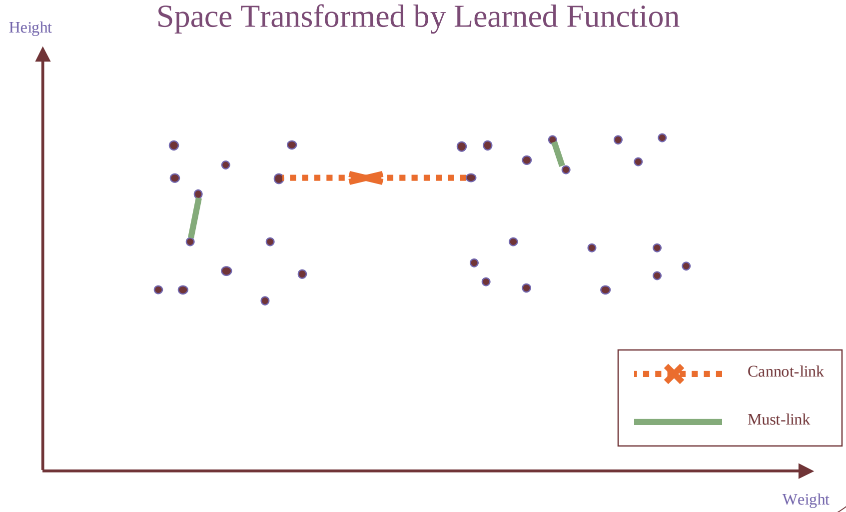
[Klein et al.'02, Cohn et al.'03, Xing et al.'03, Bar Hillel et al.'03, Bilenko et al.'03, Kamvar et al.'03, Hertz et al.'04, De Bie et al.'04]

Example: Learning Distance Function



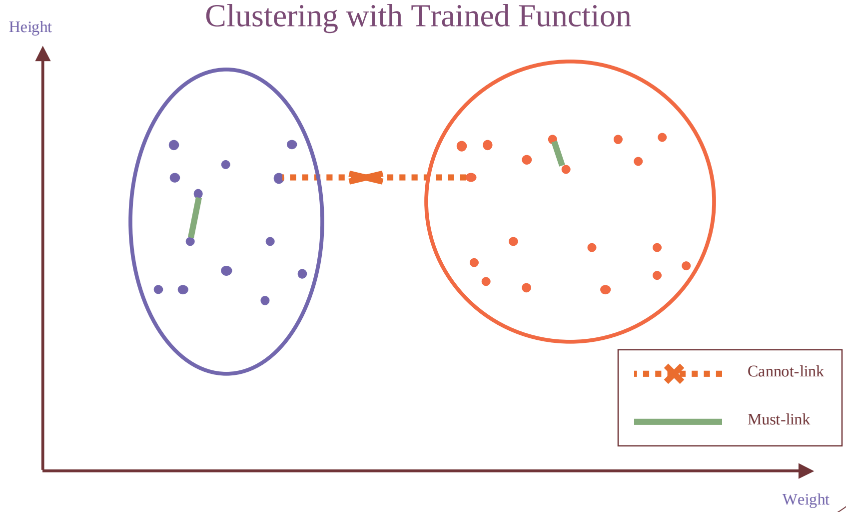
Example: Learning Distance Function

Space Transformed by Learned Function



Example: Learning Distance Function

Clustering with Trained Function

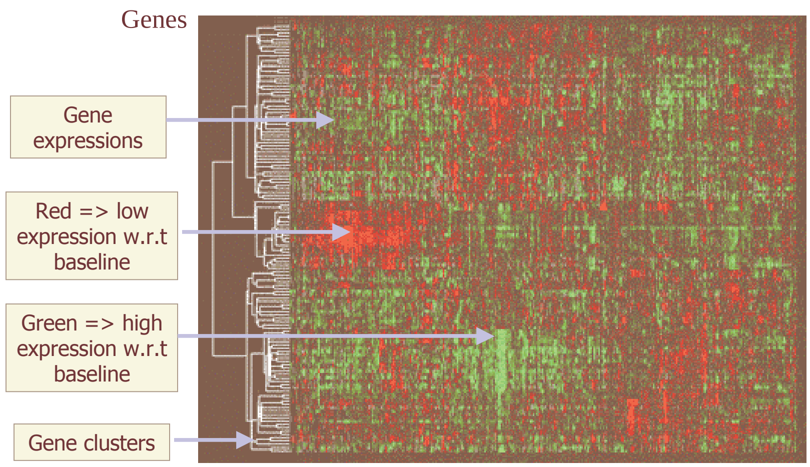


Enforce Constraints + Learn Distance

- Integrated framework [Basu et al.'04]
  - Respect constraints during cluster assignment
  - Modify distance function during parameter re-estimation
- Advantage of integration
  - Distance function can change the space to decrease constraint violations made by cluster assignment
  - Uses both constraints and unlabeled data for learning distance function

Outline

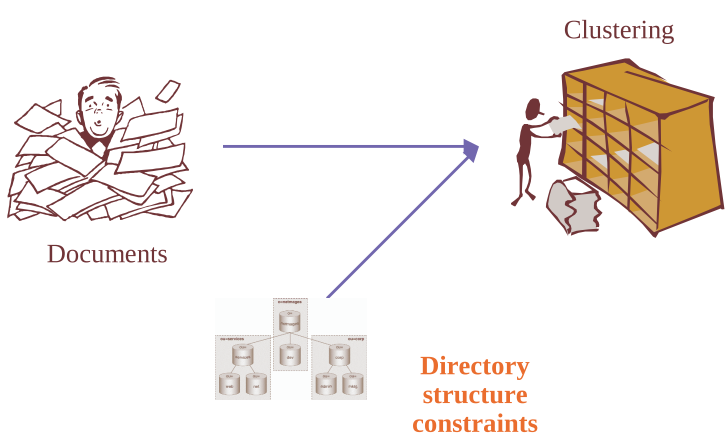
Gene Clustering Using Micro-array Data



Experiments

- **Constraints from gene interaction information in DIP**

Content Management: Document Clustering



Personalizing Web Search Result Clustering

Query: jaguar

Jaguar cars

Jaguar animal

Macintosh OS X (Jaguar)

- **Constraints mined from co-occurrence information in query web-logs**

Automatic Lane Finding from GPS traces

[Wagstaff et al. '01]

Lane-level navigation (e.g., advance notification for taking exits)

Lane-keeping suggestions (e.g., lane departure warning)

- **Constraints inferred from trace-contiguity (ML) & max-separation (CL)**

Outline

Summary of Benefits

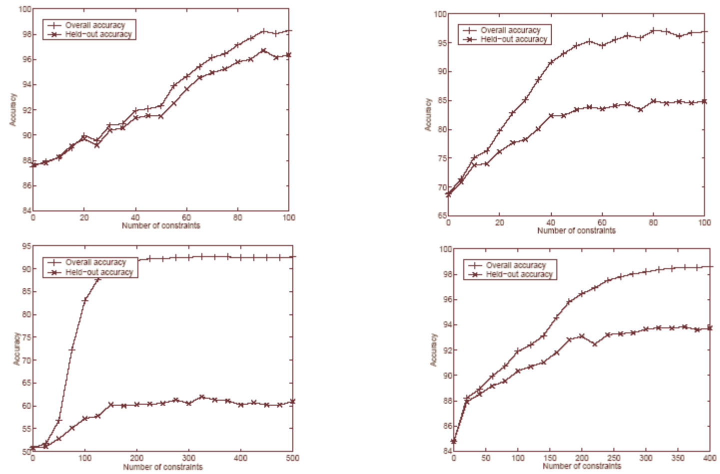
- Non-hierarchical Clustering
  - Find clusters where standard distance functions could not
  - Find solutions with given properties
  - Improve convergence time of algorithms
- Hierarchical Clustering
  - Improved quality of dendrogram
  - Use triangle inequality to speed up agglomerative algorithms
- Graphs
  - Clustering using constraints
  - Clustering graphs with real valued edges while respecting auxiliary constraint graph

Learning Distance Functions

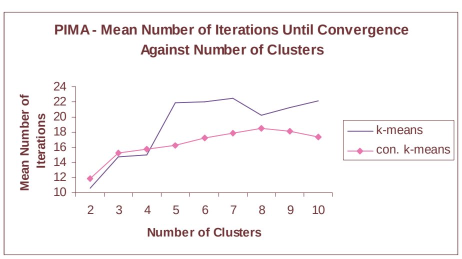
The Effects of Constraints on Clustering Solutions

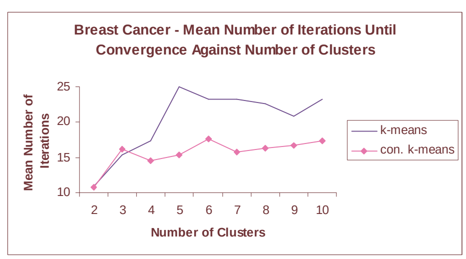
- Constraints divide the set of all plausible solutions into two sets: feasible and infeasible: $S = S_F \cup S_I$
- Constraints effectively reduce the search space to S_F
- S_F all have a common property
- So its not unexpected that we find solutions with a desired property and find them quickly.

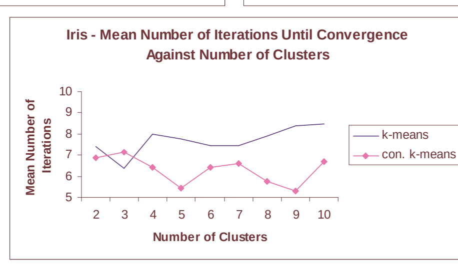
Effect of Constraints on Cluster Purity
[Wagstaff '02]



Effects of Constraints on Convergence Time







Outline

The Feasibility Problem

- We've seen that constraints are useful …
- But is there a catch?
- We are now trying to find a clustering under all sorts of constraints

Feasibility Problem

Given a set of data points *S*, a set of *ML* and *CL* constraints,

a lower (K_L) and upper bound (K_u) on the number of clusters,

is there **at least one** single set partition of *S* into *k* blocks, $K_U \geq k \geq K_L$

such that no constraints are violated?

i.e. CL(a,b), CL(b,c), CL(a,c), k=2?

Investigating the Feasibility Problem and Consequences?

- For a constraint type or combination:
  - P :construct a polynomial time algorithm
  - NP-complete : reduce from known NP-complete problem
- If the feasibility problem is in P then we can:
  - Use the algorithms to check if a single feasible solution exists before we even apply K-Means
  - Add feasibility checking as a step in K-Means.
- If feasibility problem is NP-complete then:
  - If we try to find a feasible solution at each iteration of K-Means, could take a long time as problem is intractable.

Summary of Feasibility Complexity Results

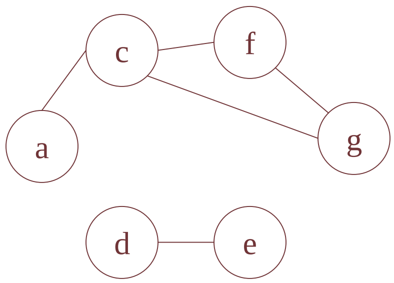
Constraint	Complexity
Must-Link	**P**
Cannot-Link	**NP-Complete**
δ-constraint	**P**
ϵ-constraint	**P**
Must-Link and δ	**P**
Must-Link and ϵ	**NP-complete**
δ and ϵ	**P**

Table 1: Results for Feasibility Problems

Cannot Link Example

Instances a thru z

Constraints: CL(a,c), CL(d,e), CL(f,g), CL(c,g), CL(c,f)



Graph K-coloring problem

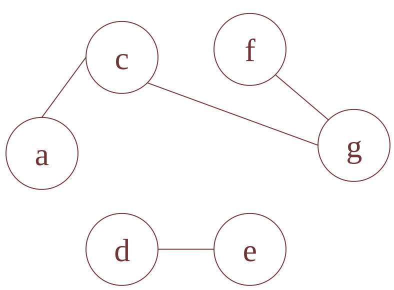
Graph K-coloring problem is intractable for all values of $K \geq 3$

See [Davidson and Ravi '05] for polynomial reduction from graph K-coloring problem.

Must Link Example

Instances a …z

ML(a,c), ML(d,e), ML(f,g), ML(c,g)



M1={a,c,f,g}
M2={d,e}

Let r be the size of the transitive closure (i.e. r=2 above), the number of connected components

Infeasible if k > (n-|TC|)– r
> 26-6 – 2
i.e., can't have too many clusters

New Results

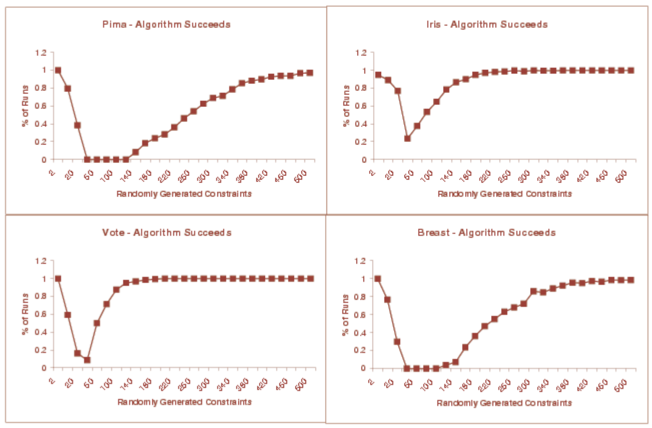
- Feasibility Problem for Disjunctions of ML and CL constraints are intractable
- But Feasibility Problem for Choice sets of ML and CL constraints are easy.
  - $ML(\mathbf{x}, y_1) \cup ML(\mathbf{x}, y_2) \ldots \cup ML(\mathbf{x}, y_n)$
  - i.e. x must-be linked with one of the y's.

Is The Feasibility Problem Really a Problem

- Wait! You said clustering under cannot link constraints was intractable.
- Worst case results say that there is one at least one "hard" problem instance so pessimistically we say the entire problem is hard.
- But when and how often does feasibility become a problem.
- Set k = # extrinsic clusters
- Randomly generated constraints by choosing two instances
- Run COP-k-means

Experimental Results

Figure 3: Graph of the proportion of times from 500 independent trials the algorithm in figure 2 gets stuck for various number of randomly chosen ML and CL constraints, k = number of instrinsic classes: Iris (3), Pima (2), Breast (2) and Vote (2).



Outline

- Introduction [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits of constraints [Ian]
- Feasibility and complexity [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Enforcing Constraints

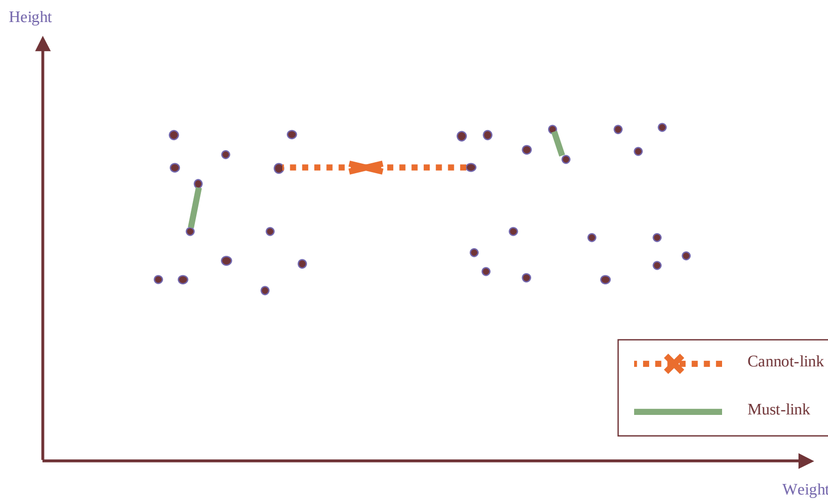
- Constraints are strong background information that should be satisfied.
- Two options
  - Satisfy all constraints, but we will run into infeasibility problems
  - Satisfy as many constraints as possible, but working out largest subset of constraints is also intractable (largest-color problem)

COP-k-Means – Nearest-"Feasible"-Centroid Idea

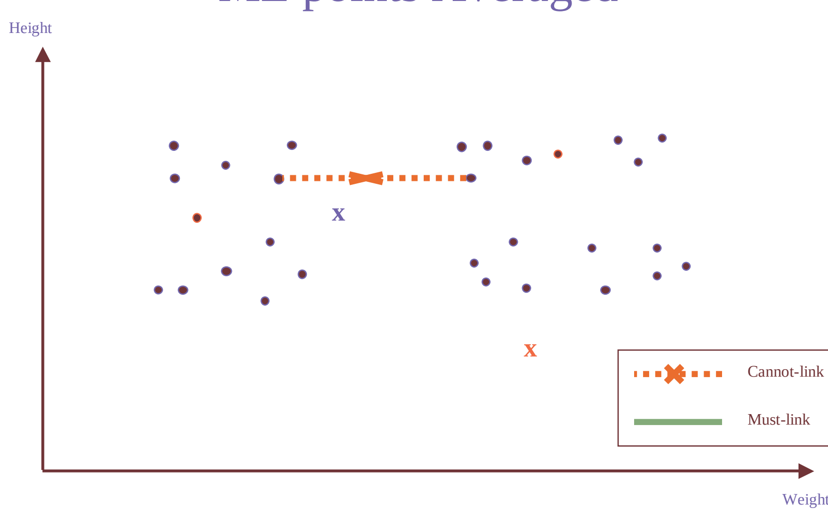
Input: S_u: unlabeled data, S_l: labeled data, k: the number of clusters to find, q: number of constraints to generate.
Output: A set partition of $S = S_u \cup S_l$ into k clusters so that all the constraints in $C = ML \cup CL$ are satisfied.

1. $ML = \emptyset, CL = \emptyset$
2. **loop** q times **do**
   (a) Randomly choose two distinct points x and y from S_l.
   (b) if(Label(x) = Label(y)) $ML = ML \cup \{x, y\}$ else $CL = CL \cup \{x, y\}$
3. Compute the transitive closure from ML to obtain the connected components $CC_1, ..., CC_r$.
4. For each i, $1 \le i \le r$, replace data points in CC_i with the average of the points in CC_i.
5. Randomly generate cluster centroids $C_1, \ldots, C_k$.
6. **loop** until convergence **do**
   (a) **for** $i = 1$ **to** $|S|$ **do**
      (a.1) Assign s_i to closest feasible cluster.
   (b) Recalculate $C_1, \ldots, C_k$.

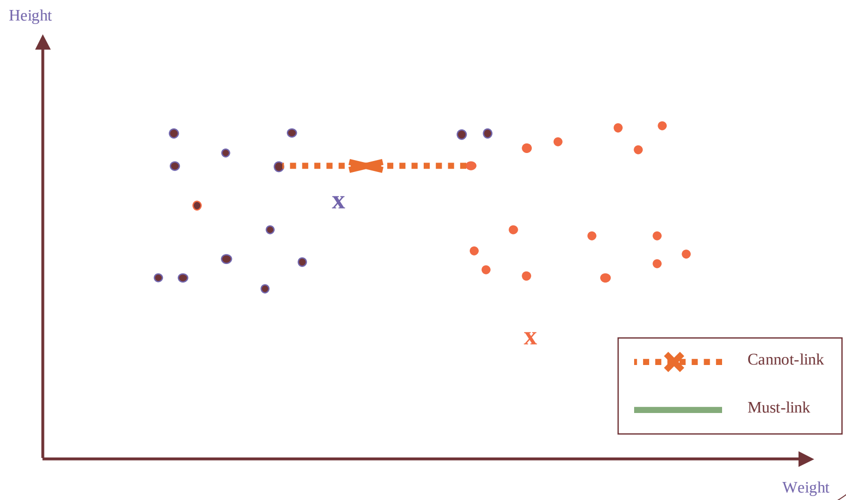
Example: COP-K-Means - 1



Example: COP-K-Means – 2
ML points Averaged



Example: COP-K-Means – 3
Nearest-Feasible-Assignment



Trying To Minimize VQE and Satisfy As Many Constraints As Possible

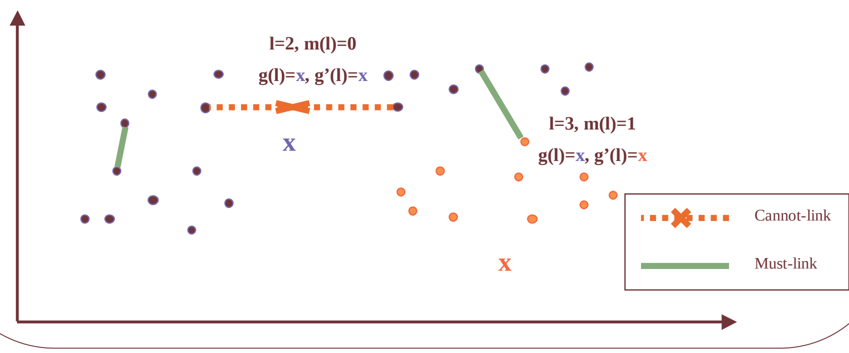
- Can't rely on expecting that I can satisfy all constraints at each iteration.
- Change aim of K-Means from:
  - Find a solution satisfying all the constraints and minimizing VQE

    TO

  - Find a solution satisfying most of the constraints (penalized if a constraint is violated) and minimizing VQE
- Two tricks
  - Need to express penalty term in same units as VQE/distortion
  - Need to rederive K-Means (as a gradient descent algorithm) from first principles.

An Approximation Algorithm – Notation

g(l), g'(l) and m(l) refer to the l^{th} constraint
g(l) : assigned cluster for first instance in constraint
g'(l) : assigned cluster for second instance in constraint
m(l) = 1 for must link, m(l) = 0 for cannot link



New Differentiable Objective Function

Satisfying a constraint may increase distortion
Trade-off between satisfying constraints and distortion requires measurement in the same units

$$(5.5) \quad CVQE_j = \frac{1}{2}\sum_{s_i \in Q_j} T_{j,1} + \frac{1}{2}\sum_{l=1, g(l)=j}^{s+r} (T_{j,2} \times T_{j,3})$$

where

$$T_{j,1} = (C_j - s_i)^2$$
$$T_{j,2} = \left[(C_j - C_{g'(l)})^2 \neg\Delta(g'(l), g(l))\right]^{m_l}$$
$$T_{j,3} = \left[(C_j - C_{h(g'(l))})^2 \Delta(g(l), g'(l))\right]^{1-m_l}$$

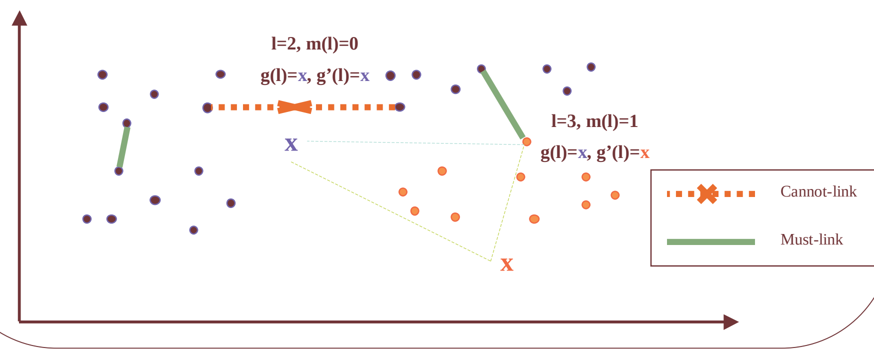
Only one is non-zero per constraint violation

If ML violated add distance between clusters

If CL violated add distance between cluster and nearest cluster

Visualizing the Penalties

Either satisfy the constraint, or
Assign to the "nearest" centroid but with a penalty



Constrained K-Means Algorithm

Algorithm aims to minimize CVQE and has a formal derivation

Randomly assign each instance to a cluster.

1. C_j = *Average of points assigned to j*

   + *Centroids of points that* ***should be*** *assigned to j*

   + *Nearest Centroids to points* ***that should not to be*** *assigned to j*

2. NN assignment for each instance using new distance

   Assign x to C_j iff $argmin_j\ CVQE\ (x,C_j)$

Goto 1 until $\Delta CVQE$ is small

Must Link Penalties

Cannot Link Penalties

Approximation Algorithm Experiments

- Binary class problems.
  - Use small amount of labeled data to generate ML between similar labeled instances and CL between different label instances
- As Wagstaff, Klein and Basu found cluster purity increases for k=2.
- For k>=2
  - The algorithm converged in fewer iterations than regular unconstrained k-means
  - On average vector quantization error was less than unconstrained k-means
  - Manages trade-off between satisfying constraints and minimizing VQE

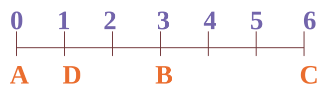
Outline

Hierarchical Clustering

Agglomerative Hierarchical Clustering

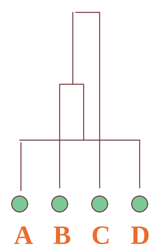
1. Initially, every instance is in its own cluster
2. Compute similarities between each cluster
3. Merge two most **similar** clusters into one.
4. Goto 2

Time Complexity $O(n^2)$



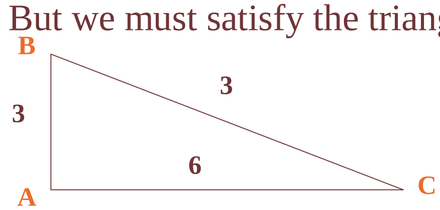
D =

	A	B	C	D
A	0	3	6	1
B	3	0	3	2
C	6	3	0	5
D	1	2	5	0



Modify the Distance Matrix (D) To Satisfy Instance Level Constraints (KKM02) - 1

- Metric spaces. Only changing the distance matrix not the distance function.
- But we must satisfy the triangle inequality



$d(x,y) \leq d(x,z) + d(z,y)$

$d(x,y) \geq | d(x,z) - d(z,y) |$

- If inequality did not hold then shortest distance between two points wouldn't be a line.

Modify the Distance Matrix (D) To Satisfy Instance Level Constraints (KKM02) - 2

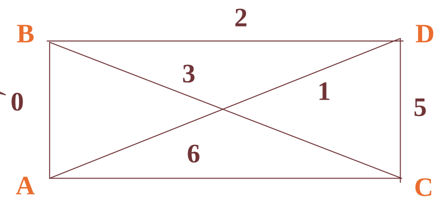
D =

	A	B	C	D
A	0	0	6	1
B	0	0	3	2
C	6	3	0	5
D	1	2	5	0

Causes Violation

ML(A,B)

CL(A,D)



$d(x,y) \le d(x,z) + d(z,y)$

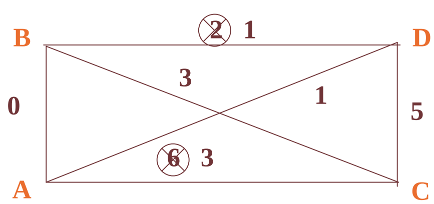
$d(x,y) \ge | d(x,z) - d(z,y) |$

Algorithm

- 1): Change ML distance instance entries in D to 0
- 2): Calculate D' from D using all pairwise shortest path algorithms, takes $O(n^3)$
- 3): D'' = D' Except Change CL distance entries to be max(D)+1

Modify the Distance Matrix (D) To Satisfy Instance Level Constraints (KKM02) - 3

STEP 2



$d(x,y) \le d(x,z) + d(z,y)$

$d(x,y) \ge | d(x,z) - d(z,y) |$

D' =

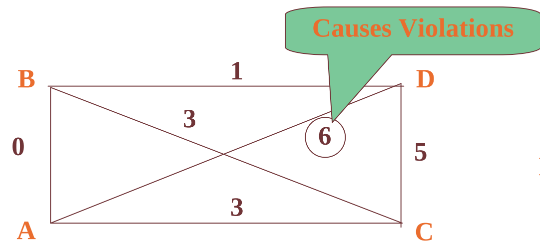
	A	B	C	D
A	0	0	3	1
B	0	0	3	1
C	3	3	0	5
D	1	1	5	0

Algorithm

- 1): Change ML distance instance entries in D to 0
- 2): Calculate D' from D using all pairwise shortest path algorithms, takes $O(n^3)$
- 3): D'' = D' Except Change CL distance entries to be max(D)+1

Modify the Distance Matrix (D) To Satisfy Instance Level Constraints (KKM02) - 4

S T E P 3



D'' =

	A	B	C	D
A	**0**	**0**	**3**	**6**
B	**0**	**0**	**3**	**1**
C	**3**	**3**	**0**	**5**
D	**6**	**1**	**5**	**0**

But Because of entailment property of CL we "maintain" the triangle inequality

Join(A,B)

Can't Join((A,B),D) instead Join((A,B),C) and then stop

Indirectly made d(B,D) and d(A,C) >> 6 and make inequality indirectly hold.

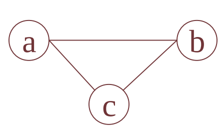
Feasibility, Dead-ends and Speeding Up Agglomerative Clustering

Feasibility Problem

Instance: Given a set S of points, a (symmetric) distance function $d(x,y) \geq 0\ \forall x,y$ and a collection of *C* constraints.

Problem: Can *S* be partitioned into **at least one** single subsets (clusters) so that all constraints are satisfied?

CL(a,b),
CL(b,c),
CL(a,c)
(k=3, k=2, k=1)?



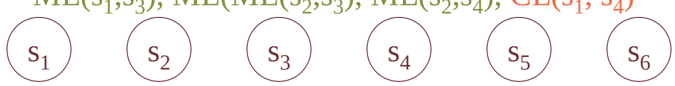
For fixed *k* equivalent to graph coloring so NP-complete

Feasibility Results [11,12]

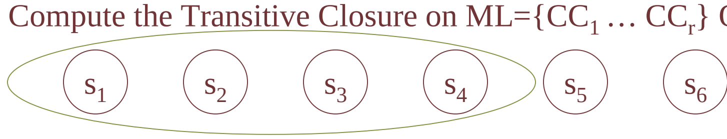
Constraint	Given *k*	Unspecified *k*
ML	**P**	**P**
CL	**NP-complete**	**P**
δ	**P**	**P**
ε	**P**	**P**
ML and ε	**NP-complete**	**P**
ML and δ	**P**	**P**
δ and ε	**P**	**P**
ML, CL and ε	**NP-complete**	**NP-complete**

Feasibility under ML and CL

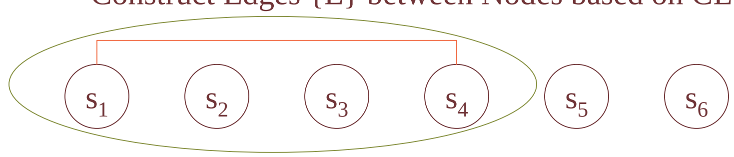
$ML(s_1,s_3)$, $ML(ML(s_2,s_3)$, $ML(s_2,s_4)$, $CL(s_1, s_4)$



Compute the Transitive Closure on ML={CC_1 … CC_r} $O(n+m_{ML})$



Construct Edges {E} between Nodes based on CL: $O(m_{CL})$



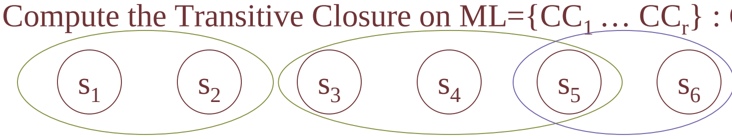
Infeasible: iff $\exists h,k : e_h(s_i, s_j) : s_i, s_j \in CC_k$: $O(m_{CL})$

Feasibility under ML and ε

$S' = \{x \in S : x$ *does* ***not*** *have an* ε *neighbor*$\} = \{s_5, s_6\}$
Each of these should be in their own cluster

$ML(s_1,s_2)$, $ML(s_3,s_4)$, $ML(s_4,s_5)$

Compute the Transitive Closure on $ML=\{CC_1 \ldots CC_r\}$: O(n+m)



Infeasible: iff $\exists i,j : s_i \in CC_j, s_i \in S'$: $O(|S'|)$

An Algorithm for ML and CL Constraints

ConstrainedAgglomerative(S,ML,CL) **returns** $Dendrogram_i$, $i = k_{min} \ldots k_{max}$

Notes: In Step 5 below, the term "mergeable clusters" is used to denote a pair of clusters whose merger does not violate any of the given CL constraints. The value of t at the end of the loop in Step 5 gives the value of $k_{\min}$.

1. Construct the transitive closure of the ML constraints (see [4] for an algorithm) resulting in r connected components $M_1, M_2, \ldots, M_r$.
2. If two points $\{x, y\}$ are both a CL and ML constraint then output "No Solution" and stop.
3. Let $S_1 = S - (\bigcup_{i=1}^{r} M_i)$. Let $k_{\max} = r + |S_1|$.
4. Construct an initial feasible clustering with $k_{\max}$ clusters consisting of the r clusters $M_1, \ldots, M_r$ and a singleton cluster for each point in S_1. Set $t = k_{\max}$.
5. **while** (there exists a pair of mergeable clusters) **do**
   (a) Select a pair of clusters C_l and C_m according to the specified distance criterion.
   (b) Merge C_l into C_m and remove C_l. (The result is $Dendrogram_{t-1}$.)
   (c) $t = t - 1$.
   endwhile

Fig. 2. Agglomerative Clustering with ML and CL Constraints

Empirical Results

Data Set	Distortion		Purity	
	Unconstrained	Constrained	Unconstrained	Constrained
Iris	3.2	2.7	58%	66%
Breast	8.0	7.3	53%	59%
Digit (3 vs 8)	17.1	15.2	35%	45%
Pima	9.8	8.1	61%	68%
Census	26.3	22.3	56%	61%
Sick	17.0	15.6	50%	59%

Table 2. Average Distortion per Instance and Average Percentage Cluster Purity over Entire Dendrogram

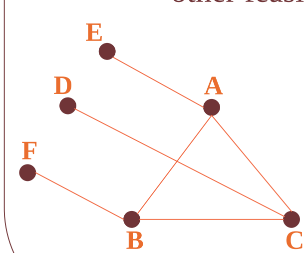
Data Set	Unconstrained	Constrained
Iris	22,201	3,275
Breast	487,204	59,726
Digit (3 vs 8)	3,996,001	990,118
Pima	588,289	61,381
Census	2,347,305,601	563,034,601
Sick	793,881	159,801

Table 3. The Rounded Mean Number of Pair-wise Distance Calculations for an Unconstrained and Constrained Clustering using the δ constraint

Dead-end Clusterings

Definition 3. *A feasible clustering* $C = \{C_1, C_2, \ldots, C_k\}$ *of a set* S *is* ***irreducible*** *if no pair of clusters in* C *can be merged to obtain a feasible clustering with* $k-1$ *clusters.*

A *k* cluster clustering is a dead-end if it is irreducible, even though other feasible clusterings with <*k* clusters exist



The Greedy Closest Join Algorithm:

Join (F,D) Join (FD,E)

But then get stuck

Alternative is:

Join(F,C), Join(D,A), Join(E,B)

Why Are Dead-Ends a Problem?

- Theorem (in technical report)
  - Let $k_{min} < k_{max}$, then if there is a feasible clustering with k_{max} clusters and a "coarsening" with k_{min} clusters there exists a feasible clustering **for every value** between k_{min} and k_{max}
- But you can't always go from a clustering with k_{max} to one with k_{min} clusters if you perform closest cluster merge.
- That is if you use traditional agglomerative algorithms your dendrogram can end prematurely.

Dead-End Results

- For dead-end situations, you can't use agglomerative clustering algorithms, otherwise you'll prematurely terminate the dendrogram.

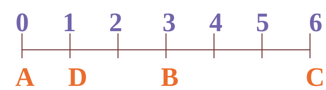
Constraint	Dead-end Solutions?
ML	No [PKDD05]
CL	Yes [PKDD05]
δ	No [PKDD05]
ϵ	No [PKDD05]

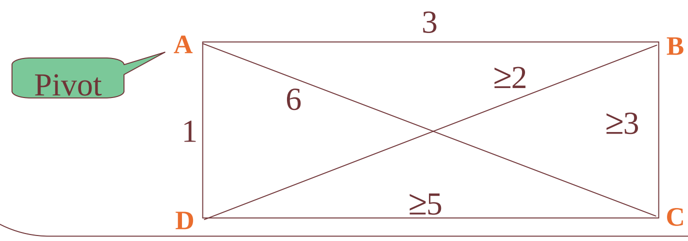
Constraint	Dead-end Solutions?
ML and ϵ	No [PKDD05]
ML and δ	No [PKDD05]
δ and ϵ	No [PKDD05]
ML, CL & ϵ	Yes [PKDD05]

Speeding Up Agglomerative Clustering Using the Triangle Inequality - 1

Definition 2. *(The γ Constraint For Hierarchical Clustering) Two clusters whose geometric centroids are separated by a distance greater than γ cannot be joined.*

Calculate distance between a pivot and all other points
Bound distances on remaining pairs of points





Speeding Up Agglomerative Clustering Using the Triangle Inequality - 2

Let $\gamma = 2$

D =

	A	B	C	D
A	0	3	6	1
B	3	0	≥3	≥2
C	6	≥3	0	≥5
D	1	≥2	≥5	0

Data Set	Unconstrained	Using γ Constraint
Iris	22,201	19,830
Breast	487,204	431,321
Digit (3 vs 8)	3,996,001	3,432,021
Pima	588,289	501,323
Census	2,347,305,601	1,992,232,981
Sick	793,881	703,764

Mean number of distance calculations

Calculate: D(a,b)=1, D(a,c) = 3, D(a,d) = 6
Save D(b,d)≥5 D(c,d)≥3
Calculate D(b,c)≥2,

Algorithm

IntelligentDistance $(\gamma,\ C = \{C_1, \ldots, C_k\})$
returns $d(i,j)\ \forall i, j.$

1. **for** $i = 2$ **to** $n-1$ $\quad d_{1,i} = D(C_1, C_i)$ endloop
2. **for** $i = 2$ to $n-1$
   - **for** $j = i+1$ to $n-1$ $\quad \hat{d_{i,j}} = |d_{1,i} - d_{1,j}|$
     - **if** $\hat{d_{i,j}} > \gamma$ then $d_{i,j} = \gamma + 1$; *do not join* **else** $d_{i,j} = D(x_i, x_j)$
   - endloop

   endloop
3. return $d_{i,j},\ \forall i, j.$

Fig. 3. Function for Calculating Distances Using the γ Constraint and the Triangle Inequality.

- Worst case result $O(n^2)$ distance calculations
- Best case calculated bound **always** exceeds γ : $O(n\text{-}1)$
- Average case using the Markov inequality: save *1/2c* distance calculations where $\gamma = c\rho$ and ρ is the average distance between two points

Outline

Distance Learning as Convex Optimization [Xing et al. '02]

- Learns a parameterized Mahalanobis distance

$$\min_A \sum_{(s_i,s_j)\in ML} \| s_i - s_j \|_A^2 = \min_A \sum_{(s_i,s_j)\in ML} (s_i - s_j)^T A (s_i - s_j)$$

$$\sum_{(s_i,s_j)\in CL} \| s_i - s_j \|_A \geq 1$$

$$s.t. \quad A \phi 0$$

Alternate formulation

- Equivalent optimization problem

$$\max_A \quad g(A) = \sum_{(s_i,s_j)\in CL} \| s_i, s_j \|_A$$

$$f(A) = \sum_{(s_i,s_j)\in ML} \| s_i, s_j \|_A^2 \leq 1 \quad \longrightarrow C_1$$

$$s.t. \quad A \phi 0 \quad \longrightarrow C_2$$

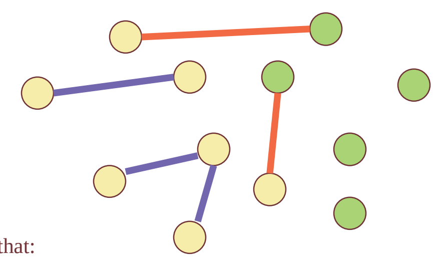
Optimization Algorithm

- Solve optimization problem using combination of
  - gradient ascent: to optimize the objective
  - iterated projection algorithm: to satisfy the constraints

Iterate
  Iterate
    $A := \arg\min_{A'} \{||A' - A||_F : A' \in C_1\}$
    $A := \arg\min_{A'} \{||A' - A||_F : A' \in C_2\}$
  until A converges
  $A := A + \alpha(\nabla_A g(A))_{\perp \nabla_A f}$
until convergence

Distance Learning in Product Space

[Hertz et al. '04]



- Input:
  - Data set X in R^n.
  - Equivalence constraints
- Output: function D: $\underbrace{X \times X}_{\text{product space}} \longrightarrow [0,1]$ such that:
  - points from the same class are close to each other.
  - points from different classes are very far from each other.
- Basic Observation:
  - *Equivalence constraints* ⇔ Binary labels in product space
  - Use boosting on product space to learn function

Boosting in a nutshell

A standard ML method that attempts to boost the performance of "weak" learners

Basic idea:

1. Initially, weights are set **equally**
2. **Iterate:**
   - **i.** **Train** weak learner on weighted data
   - **ii.** **Increase** weights of **incorrectly** classified examples (force weak learner to focus on difficult examples)
3. Final hypothesis: **combination of weak hypotheses**

EM on Gaussian Mixture Model

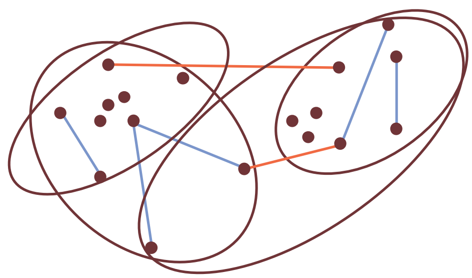
- GMM: Standard data representation that models data using a number of Gaussian sources
- The parameters of the sources are estimated using the EM algorithm:
  - E step: Calculate Expected log-likelihood of the data over all possible assignments of data-points to sources
  - M step: Differentiate the Expectation w.r.t. the **parameters**

The Weak Learner: Constrained EM

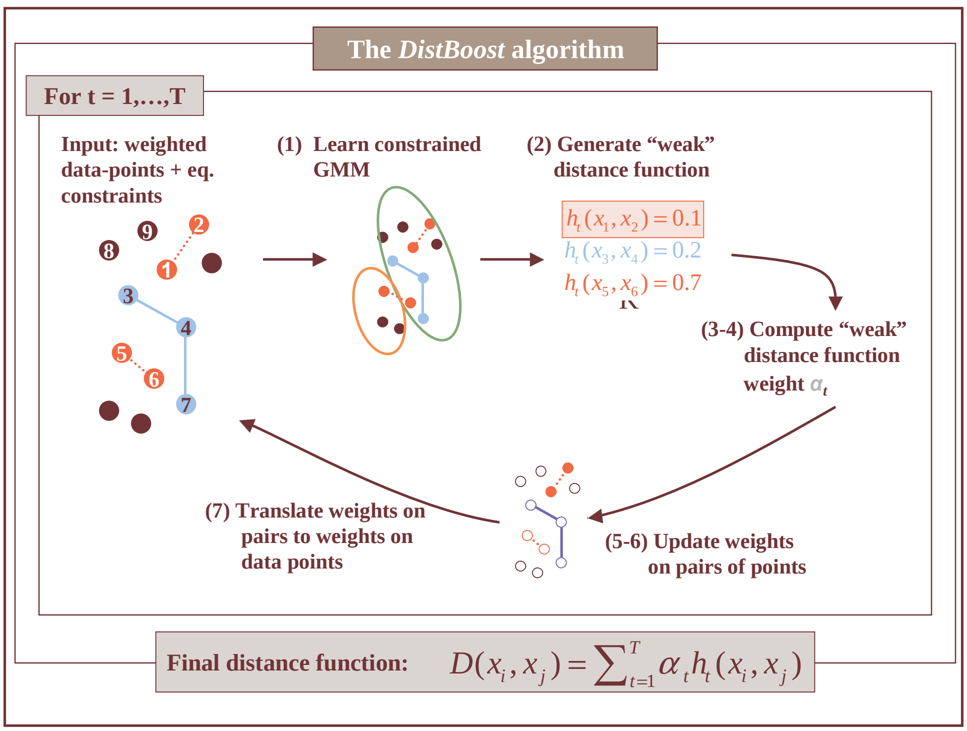
Constrained EM algorithm: fits a mixture of Gaussians to unlabeled data given a set of equivalence constraints.

Modification in case of equivalence constraints:

E step: sum only over assignments which comply with the constraints

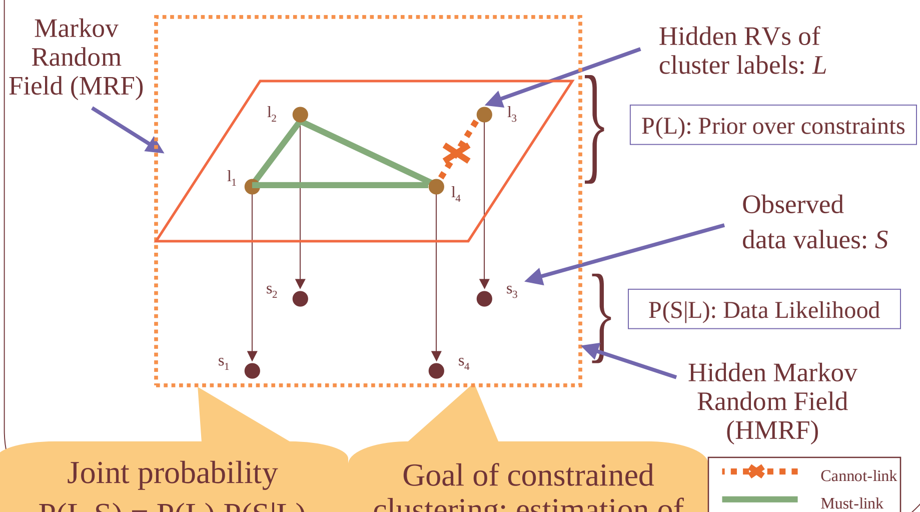


 Clustering with Constraints 97



Integrated Approach: HMRF

[Basu et al. '04]



Joint probability
P(L,S) = P(L).P(S|L)

Goal of constrained clustering: estimation of P(L,S) on HMRF

Constrained Clustering on HMRF

Constraint potentials

$$\Pr(L) \propto \exp[-\sum_{i,j} V(s_i, s_j, l_i, l_j)]$$

Cluster distortion

$$\Pr(S \mid L) \propto \exp[-\sum_{s_i} D(s_i, C_{l_i})]$$

$$\Downarrow$$

Joint probability

$$\Pr(L, S) = \Pr(S \mid L) \Pr(L)$$

Overall objective of constrained clustering

$$-\log \Pr(L, S) \propto \left(\sum_{s_i} D(s_i, C_{l_i}) + \sum_{i,j} V(s_i, s_j, l_i, l_j) \right)$$

MRF potential

- Generalized Potts (Ising) potential:

$$V(s_i, s_j, l_i, l_j) = \begin{cases} w_{ij} D_A(s_i, s_j) & if \quad l_i \neq l_j, (s_i, s_j) \in ML \\ \overline{w_{ij}} \left[D_{A,\max} - D_A(s_i, s_j) \right] & if \quad l_i = l_j, (s_i, s_j) \in CL \\ 0 & else \end{cases}$$

HMRF-KMeans: Objective Function

$$J_{HMRF} = \underbrace{\sum_{s_i \in S} D_A(s_i, C_{l_i})}_{\text{KMeans distortion}} + \underbrace{\sum_{\substack{(s_i, s_j) \in ML \\ s.t.\, l_i \neq l_j}} w_{ij} D_A(s_i, s_j)}_{\text{ML violation: constraint-based}} + \underbrace{\sum_{\substack{(s_i, s_j) \in CL \\ s.t.\, l_i = l_j}} \overline{w_{ij}} \left(D_{A,\max} - D_A(s_i, s_j) \right)}_{\text{CL violation: constraint-based}}$$

KMeans distortion

ML violation: constraint-based

CL violation: constraint-based

Penalty function: distance-based

-log P(S|L)

-log P(L)

HMRF-KMeans: Algorithm

Initialization:

- Use neighborhoods derived from constraints to initialize clusters

Till *convergence*:

1. **Point assignment:**

- Assign each point s to cluster h^* to minimize both distance and constraint violations

2. **Mean re-estimation:**

- Estimate cluster centroids C as means of each cluster
- Re-estimate parameters A of D_A to minimize constraint violations

HMRF-KMeans: Convergence

Theorem:

HMRF-KMeans converges to a local minima of J_{HMRF} for for Bregman divergences D (e.g., KL divergence, squared Euclidean distance) or directional distances (e.g., Pearson's distance, cosine distance)

Ablation/Sensitivity Experiment

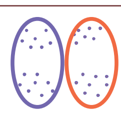
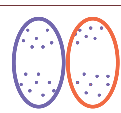
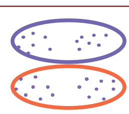
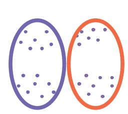
- MPCK-Means: both constraints and distance learning
- MK-Means: only distance learning
- PCK-Means: only constraints
- K-Means: purely unsupervised

Evaluation Measure

- Compare cluster partitioning to class labels on the dataset
- Mutual Information measure calculated only on test set

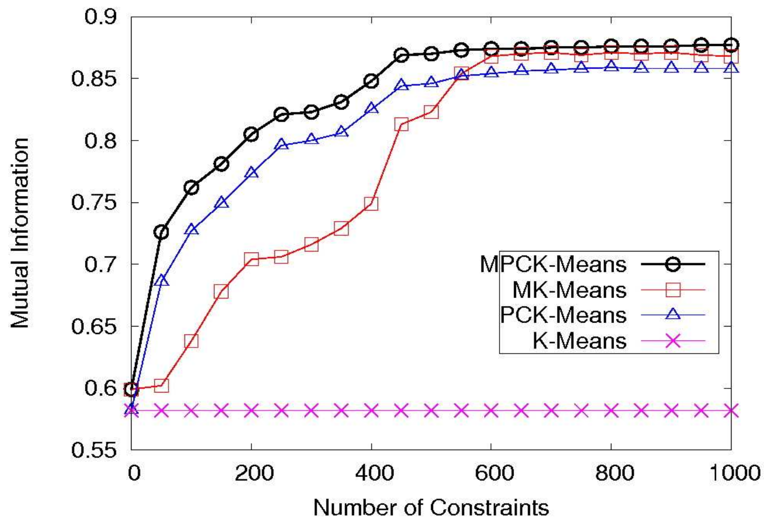
[Strehl et al. '00]

$$MI = \frac{I(C;K)}{[H(C)+H(K)]/2}$$

Cluster partitions	Underlying classes	MI value
		High
		Low

Experiment Results: PenDigits subset

(squared Euclidean distance)



Outline

- Introduction [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits of constraints [Ian]
- Feasibility and complexity [Ian]
- Algorithms for constrained clustering
    - Enforcing constraints [Ian]
    - Hierarchical [Ian]
    - Learning distances [Sugato]
    - Initializing and pre-processing [Sugato]
    - Graph-based [Sugato]

Finding Informative Constraints given a quota of Queries

- Active learning for constraint acquisition [Basu et al.'04]:
  - In interactive setting, constraints obtained by queries to a user
  - Need to get **informative** constraints to get better clustering

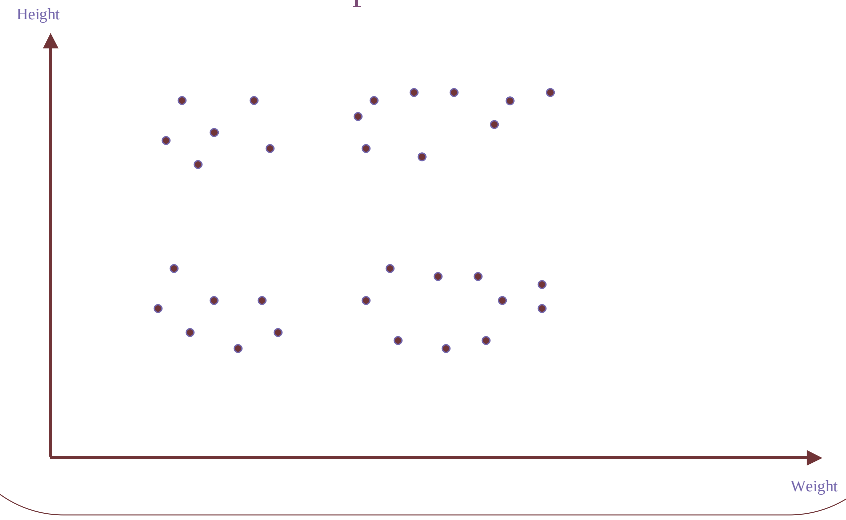
- Two-phase active learning algorithm:
  - Explore: Use *farthest-first* traversal [Hochbaum et al.'85] to explore the data and find K pairwise-disjoint neighborhoods (cluster skeleton) rapidly
  - Consolidate: Consolidate basic cluster skeleton by getting more points from each cluster, within max $(K-1)$ queries for any point

Algorithm: Explore

- Pick a point s at random, add it to neighborhood N_1, $\lambda = 1$
- While queries are allowed and ($\lambda < k$)
  - Pick point s farthest from existing λ neighborhoods
  - If by querying s is *cannot-linked* to all existing neighborhoods, then set $\lambda = \lambda+1$, start new neighborhood N_λ with s
  - Else, add s to neighborhood with which it is *must-linked*

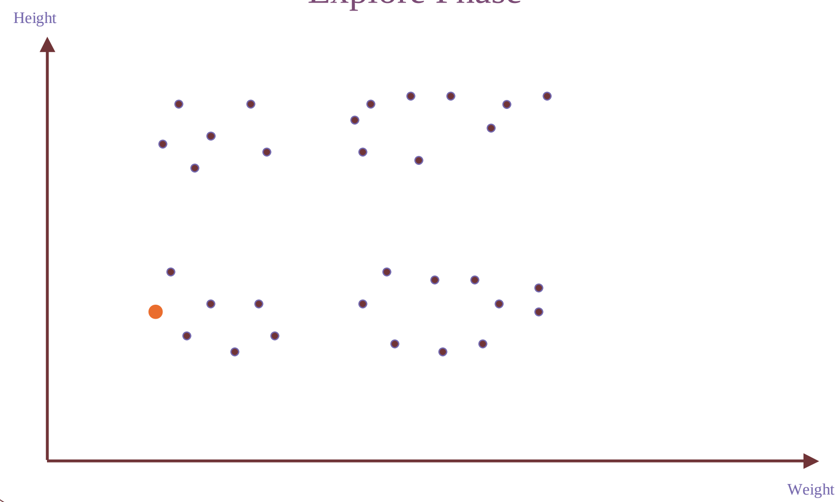
Active Constraint Acquisition for Clustering

Explore Phase



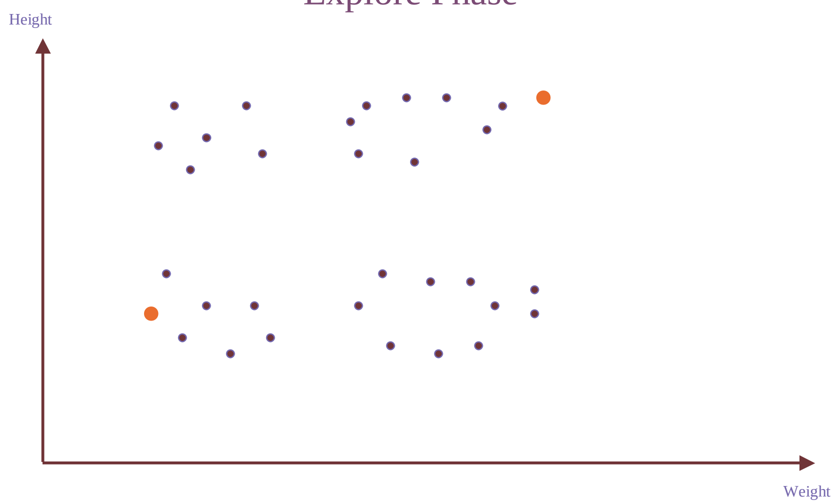
Active Constraint Acquisition for Clustering

Explore Phase



Active Constraint Acquisition for Clustering

Explore Phase



Active Constraint Acquisition for Clustering

Explore Phase

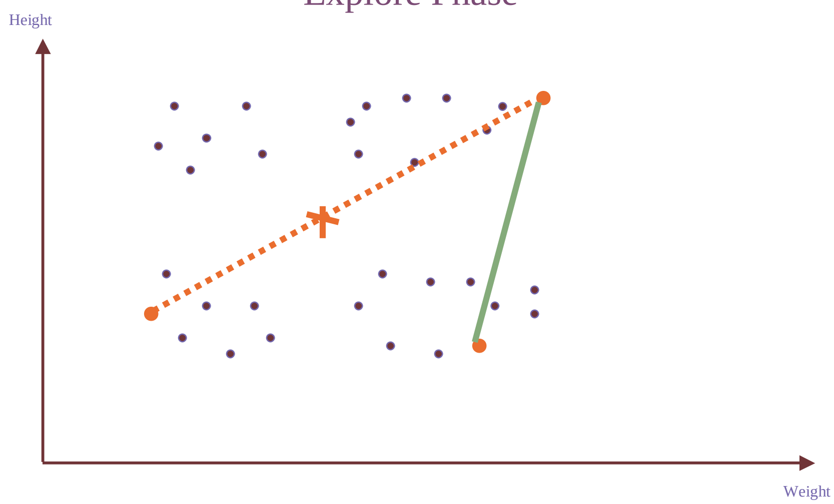
Active Constraint Acquisition for Clustering

Explore Phase

 Clustering with Constraints 115

Active Constraint Acquisition for Clustering

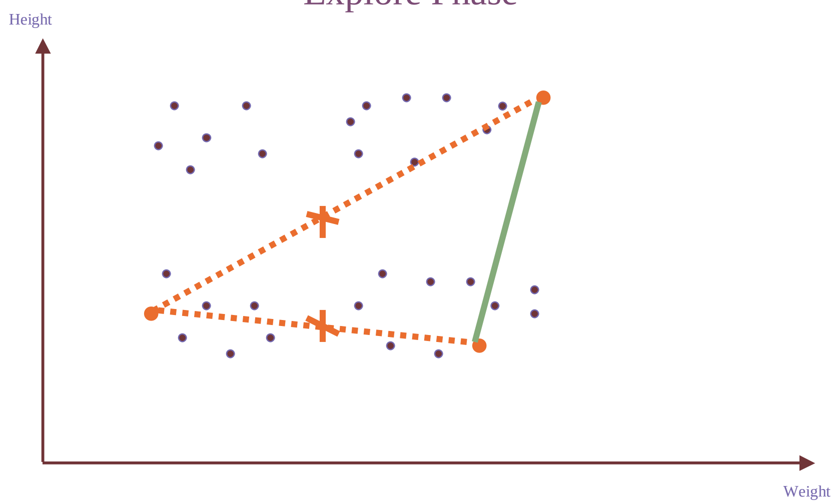
Explore Phase



 Clustering with Constraints 116

Active Constraint Acquisition for Clustering

Explore Phase

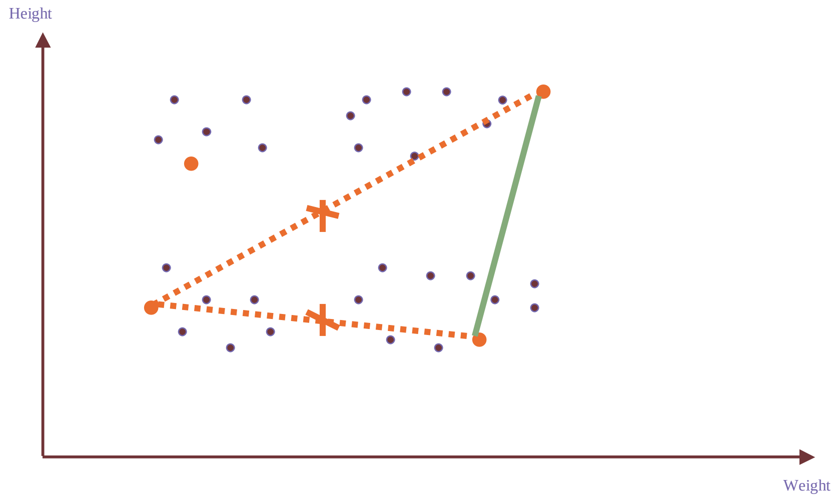


Algorithm: Consolidate

- Estimate centroids of each of the λ neighborhoods
- While queries are allowed
  - Randomly pick a point *s* not in the existing neighborhoods
  - Query *s* with each neighborhood (in sorted order of decreasing distance from *s* to centroids) until *must-link* is found
  - Add *s* to that neighborhood to which it is *must-linked*

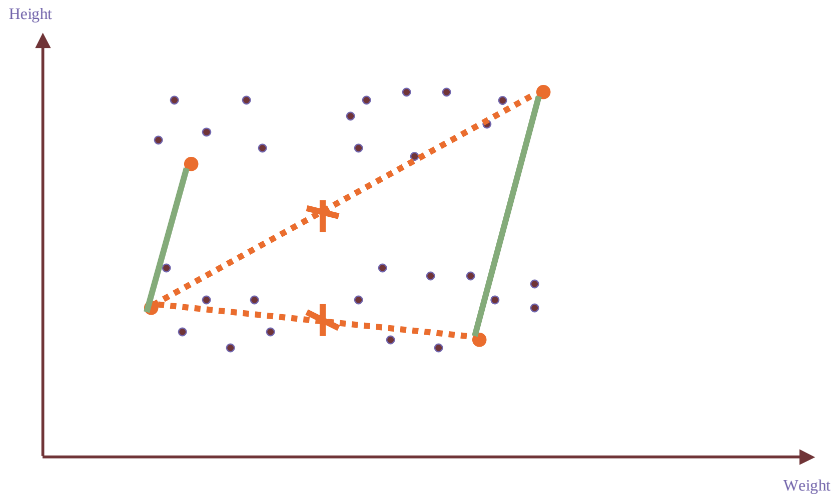
Active Constraint Acquisition for Clustering

Consolidate Phase



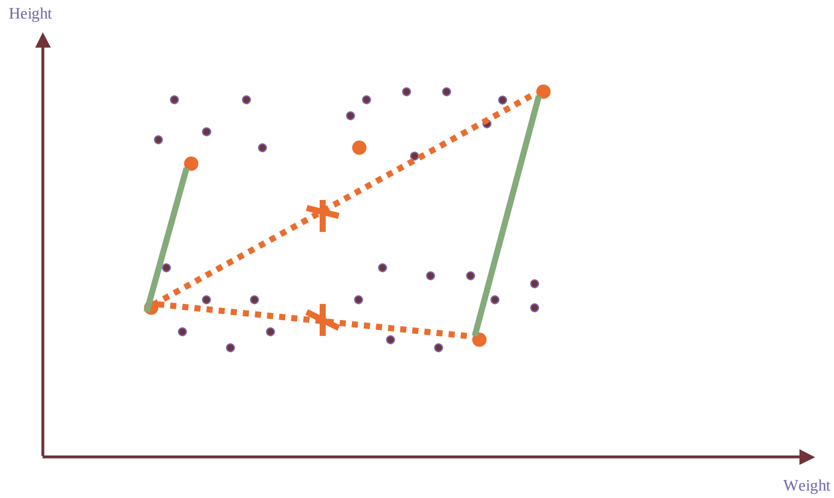
Active Constraint Acquisition for Clustering

Consolidate Phase



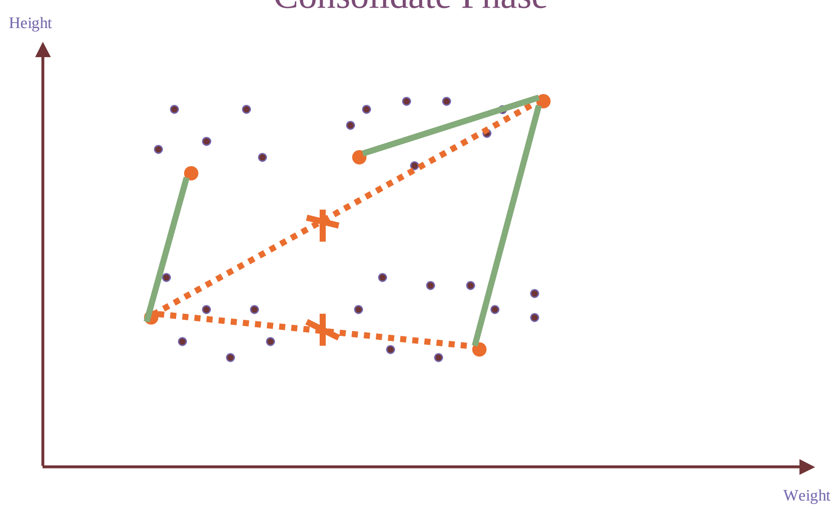
Active Constraint Acquisition for Clustering

Consolidate Phase



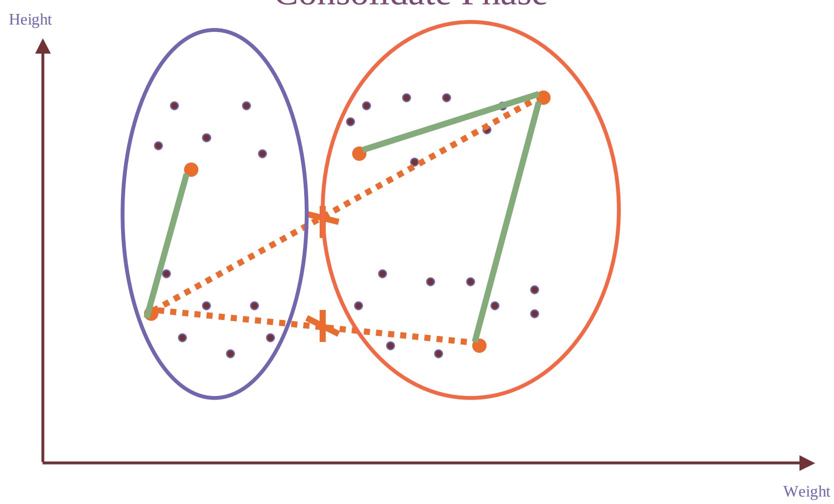
Active Constraint Acquisition for Clustering

Consolidate Phase

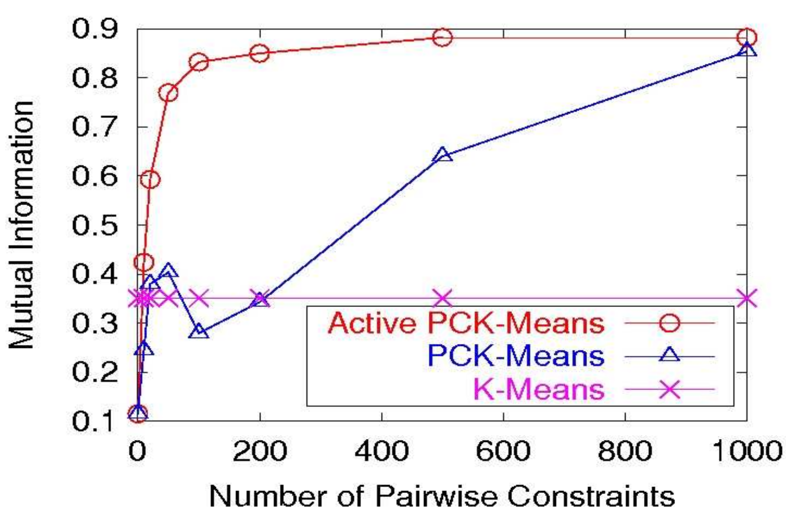


Active Constraint Acquisition for Clustering

Consolidate Phase



Experiments: 20-Newsgroups subset



Confusion Matrices

No constraints

	Cluster1	Cluster2	Cluster3
Misc	**71**	12	17
Guns	25	**61**	14
Mideast	12	36	**52**

20 queries

	Cluster1	Cluster2	Cluster3
Misc	**84**	7	9
Guns	5	**91**	4
Mideast	7	7	**86**

Algorithms to Seed K-Means When Feasibility Problem is in P [Davidson et al. '05]

- Each algorithm will find a feasible solution.
- You can build upon each to make them minimize the vector quantization error (or what-ever objective function your algorithm has) as well.

Finding a Feasible Clustering for ML Constraints

Note: Whenever a feasible solution exists, the following algorithm outputs a collection of K_ℓ clusters satisfying all the must-link constraints.

1. Compute the transitive closure of the constraints in C. Let this computation result in r sets of points, denoted by $M_1, M_2, \ldots, M_r$.
2. Let $S' = S - \bigcup_{i=1}^{r} M_i$. ($S'$ denotes the subset of points that are not involved in any must-link constraint.)
3. **if** $r \geq K_\ell$ **then**
   (a) Let $A = (\bigcup_{i=K_\ell}^{r} M_i) \cup S'$.
   (b) Output $M_1, \ldots, M_{K_\ell - 1}, A$.

   else

   if $|S'| < K_\ell - r$ **then**
   Output "There is no solution."

   else
   (a) Let $t = K_\ell - r$. Partition S' into t clusters $A_1, \ldots, A_t$ arbitrarily.
   (b) Output $M_1, \ldots, M_r, A_1, \ldots, A_t$.

Figure 1: Algorithm for the ML-Feasibility Problem

Must link constraints are Transitive: ML(a,b), ML(b,c) implies ML(a,c). Replace with ML(a,b,c)

See paper for an algorithm
r = # connected components

S' are those points not part of ML constraints

Too many connected components merge some: doesn't violate ML constraints

Too many clusters to find.

$r < K_l <= n-r$

Finding a Feasible Clustering Under the δ Constraint

1. for each point s_i do
   (a) Determine the set $X_i \subseteq S - \{s_i\}$ of points such that for each point $x_j \in X_i$, $d(s_i, x_j) < \delta$.
   (b) For each point $x_j \in X_i$, create the must-link constraint $\{s_i, x_j\}$.
2. Let C denote the set of all the must-link constraints created in Step 1. Use the algorithm for the ML-feasibility problem (Figure 1) with point set S, constraint set C and the values K_ℓ and K_u.

Figure 2: Algorithm for the δ-Feasibility Problem

Finding a Feasible Clustering Under the ε Constraint

1. Find the set $S_1 \subseteq S$ such that no point in S_1 has an ϵ-neighbor. Let $t = |S_1|$ and $S_2 = S - S_1$.
2. Construct the auxiliary graph $G(V, E)$ for S_2 (see Definition 3.1). Let G have r connected components (CCs) denoted by $G_1, G_2, \ldots, G_r$.
3. Let $N^* = t + \min\{1, r\}$. (**Note:** To satisfy the ϵ-constraint, at least N^* clusters must be used.)
4. **if** $N^* > K_u$ **then** Output "No feasible solution" and **stop**.
5. Let $C_1, C_2, \ldots, C_t$ denote the singleton clusters corresponding to points in S_1. Let $X_1, X_2, \ldots, X_r$ denote the clusters corresponding to the CCs of G.
6. **if** $t + r \geq K_u$

   then /* We may have too many clusters. */

   (a) Merge clusters $X_{K_u-t}, X_{K_u-t+1}, \ldots, X_r$ into a single new cluster X_{K_u-t}.

   (b) Output the K_u clusters $C_1, C_2, \ldots, C_t, X_1, X_2, \ldots, X_{K_u-t}$.

   else /* We have too few clusters. */

   (a) Let $N = t + r$. Construct spanning trees $T_1, T_2, \ldots, T_r$ corresponding to the CCs of G.

   (b) **while** $(N < K_\ell)$ **do**

      (i) Find a tree T_i with at least two nodes. If no such tree exists, output "No feasible solution" and **stop**.

      (ii) Let v be a leaf in tree T_i. Delete v from T_i.

      (iii) Delete the point corresponding to v from cluster X_i and form a new singleton cluster X_{N+1} containing that point.

      (iv) $N = N + 1$.

   (c) Output the K_ℓ clusters $C_1, C_2, \ldots, C_t, X_1, X_2, \ldots, X_{K_\ell-t}$.

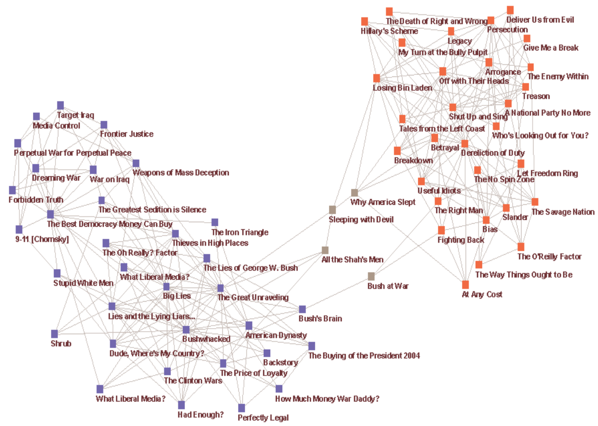
Outline

- Introduction [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits of constraints [Ian]
- Feasibility and complexity [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Graph-based Clustering

- Data input as graph:

  real valued edges between pairs of points denotes similarity



Constrained Graph-based Clustering

- Clustering criterion:

  minimize normalized cut

- Possible solution:

  Spectral Clustering

  [Kamvar et al. '03]

- Constrained graph clustering:

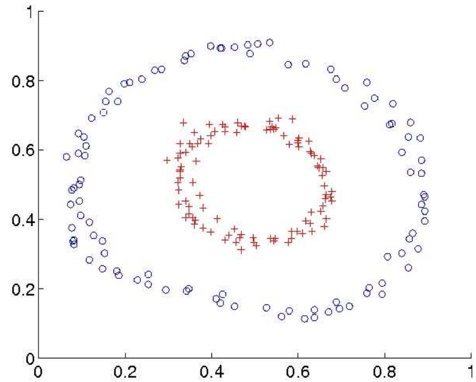
  minimize cut in input graph while maximally respecting constraints in auxilliary constraint graph

Kernel-based Clustering

- 2-circles data not linearly separable
- transform to high-D using kernel

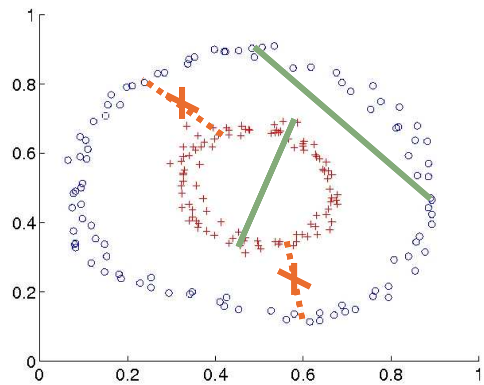
  $e.g., < s_1, s_2 >= e^{-\|s_1 - s_2\|^2}$

- cluster kernel similarity matrix using weighted kernel K-Means



Constrained Kernel-based Clustering

- Use the data and the specified constraints to create appropriate kernel



SS-Kernel-KMeans [Kulis et al.'05]

- Contributions:
  - Theoretical equivalence between constrained graph clustering and weighted kernel KMeans
  - Unifies vector-based and graph-based constrained clustering using kernels
- Algorithm:
  - Forms a kernel matrix from data and constraints
  - Runs weighted kernel KMeans
- Benefits:
  - HMRF-KMeans and Spectral Clustering are special cases
  - Fast algorithm for constrained graph-based clustering
  - Kernels allow constrained clustering with non-linear cluster boundaries

Kernel for HMRF-KMeans with squared Euclidean distance

$$J_{HMRF} = \sum_{c=1}^{k} \sum_{s_i \in S_c} \| s_i - C_c \|^2 - \sum_{\substack{(s_i, s_j) \in ML \\ s.t. l_i = l_j}} \frac{w_{ij}}{|S_{l_i}|} + \sum_{\substack{(s_i, s_j) \in CL \\ s.t. l_i = l_j}} \frac{w_{ij}}{|S_{l_i}|}$$

$$K = S + W,$$

$$\text{where} \begin{cases} S_{ij} = s_i . s_j, \\ W_{ij} = \begin{matrix} + w_{ij} \text{ if } (s_i, s_j) \in ML \\ - w_{ij} \text{ if } (s_i, s_j) \in CL \end{matrix} \end{cases}$$

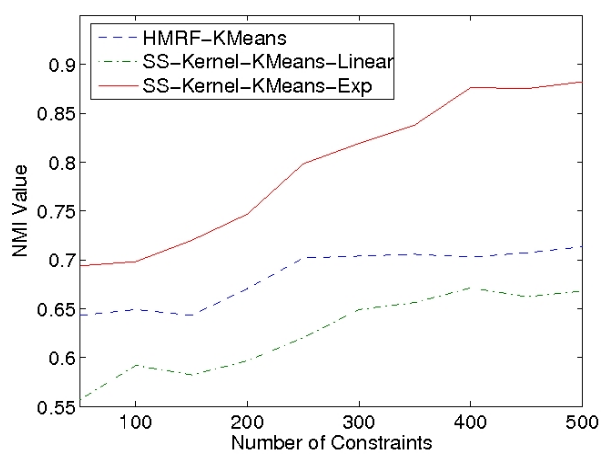
Kernel for Constrained Normalized-Cut Objective

$$J_{NormCut} = \sum_{c=1}^{k} \frac{\text{links}(V_c, V \setminus V_c)}{\deg(V_c)} - \sum_{\substack{(s_i,s_j)\in ML \\ s.t. l_i = l_j}} \frac{w_{ij}}{\deg(V_{l_i})} + \sum_{\substack{(s_i,s_j)\in CL \\ s.t. l_i = l_j}} \frac{w_{ij}}{\deg(V_{l_i})}$$

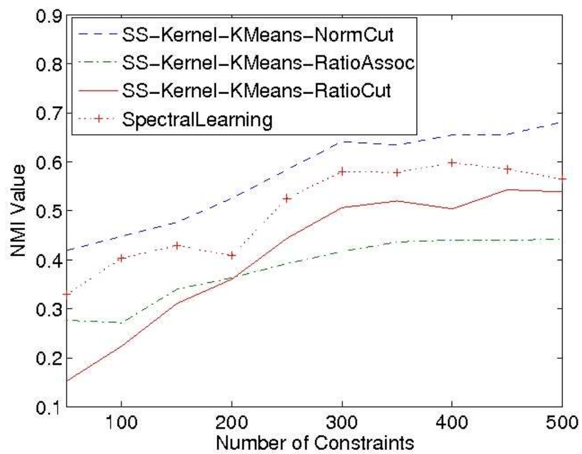
$$K = D^{-1}AD + D^{-1}WD,$$

$$\text{where} \begin{cases} A_{ij} = \text{graph affinity}(i, j), \\ D = \text{diagonal degree matrix} \\ W_{ij} = \begin{matrix} +w_{ij} \text{ if } (s_i, s_j) \in ML \\ -w_{ij} \text{ if } (s_i, s_j) \in CL \end{matrix} \end{cases}$$

Experiment: PenDigits subset



Experiment: Yeast Gene network



Today we talked about …

- Introduction [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits of constraints [Ian]
- Feasibility and complexity [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Thanks for Your Attention.
We Hope You Learnt a Few Things

Sugato will be available until Tuesday morning
Ian will be available until Monday afternoon

References - 1

[1] N. Bansal, A. Blum and S. Chawla, "Correlation Clustering", 43rd Symposium on Foundations of Computer Science (FOCS 2002), pages 238-247.

[2] S. Basu, A. Banerjee and R. J. Mooney, "Semisupervised Learning by Seeding", Proc. 19th Intl. Conf. on Machine Learning (ICML-2002), Sydney, Australia, July 2002.

[3] S. Basu, M. Bilenko and R. J. Mooney, "A Probabilistic Framework for Semi-Supervised Clustering", Proc. 10th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD-2004), Seattle, WA, August 2004. Best Paper Award.

[4] S. Basu, M. Bilenko and R. J. Mooney, "Active Semi-Supervision for Pairwise Constrained Clustering", Proc. 4th SIAM Intl. Conf. on Data Mining (SDM-2004).

[5] K. Bennett, P. Bradley and A. Demiriz, "Constrained K-Means Clustering", Microsoft Research Technical Report 2000-65, May 2000.

[6] De Bie T., Momma M., Cristianini N., "Efficiently Learning the Metric using Side-Information", in Proc. of the 14th International Conference on Algorithmic Learning Theory (ALT2003), Sapporo, Japan, Lecture Notes in Artificial Intelligence, Vol. 2842, pp. 175-189, Springer, 2003. (pdf)(bib)

[7] A. Blum, J. Lafferty, M.R. Rwebangira, R. Reddy, "Semi-supervised Learning Using Randomized Mincuts", International Conference on Machine Learning, 2004.

[8] M. Charikar, V. Guruswami and A. Wirth, "Clustering with Qualitative Information", Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003.

References - 2

[9] H. Chang, D.Y. Yeung. Locally linear metric adaptation for semi-supervised clustering. Proceedings of the Twenty-First International Conference on Machine Learning (ICML), pp.153-160, Banff, Alberta, Canada, 4-8 July 2004.

[10] D. Cohn, R. Caruana, and A. McCallum, "Semi-supervised clustering with user feedback", Technical Report TR2003-1892, Cornell University, 2003.

[11] I. Davidson, S.S. Ravi, Clustering under Constraints: Feasibility Results and the K-Means Algorithm, SIAM Data Mining Conference 2005. Best Paper Award.

[12] I. Davidson, S.S. Ravi, Hierarchical Clustering with Constraints: Theory and Practice, 9th European Principles and Practice of KDD, PKDD 2005.

[13] A. S. Galanopoulos and S. C. Ahalt. Codeword distribution for frequency sensitive competitive learning with one-dimensional input data. IEEE Transactions on Neural Networks, 7(3):752-756, 1996.

[14] j M. R. Garey and D. S. Johnson and H. S. Witsenhausen. The complexity of the generalized Lloyd-Max problem. IEEE Transactions on Information Theory, 28(2):255-256, 1982 j

[15] David Gondek, Shivakumar Vaithyanathan, and Ashutosh Garg Clustering with Model-level Constraints, SIAM International Conference on Data Mining (SDM), 2005.

[16] David Gondek and Thomas Hofmann Non-Redundant Data Clustering, 4th IEEE International Conference on Data Mining (ICDM), 2004. Best Paper Award

[17] T. F. Gonzalez, "Clustering to Minimize the Maximum Intercluster Distance", Theoretical Computer Science, Vol. 38, No. 2-3, June 1985, pp. 293-306.

 Clustering with Constraints

References - 3

[18] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin-based distance functions for clustering. ICML 2004.

[19] Aharon Bar Hillel. Tomer Hertz. Noam Shental. Daphna Weinshall Learning Distance Functions using Equivalence Relations ICML 2003.

[20] S. D. Kamvar, D. Klein, and C. Manning, "Spectral Learning," IJCAI, 2003.

[21] D. Klein, S. D. Kamvar and C. D. Manning, "From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering", *Proc. 19th Intl. Conf. on Machine Learning* (ICML 2002).

[22] B. Kulis, S. Basu, I. Dhillon, R. J. Mooney, "Semi-supervised Graph Clustering: A Kernel Approach", ICML 2005.

[23] M. Law, Alexander Topchy, Anil K. Jain, Model-based Clustering With Probabilistic Constraints, SDM 2005.

[24] Z. Lu and T. Leen, Semi-supervised Learning with Penalized Probabilistic Clustering. NIPS 2005.

[25] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, Computing Gaussian Mixture Models with EM using Side-Information. In Proc. of workshop *The Continuum from labeled to unlabeled data in machine learning and data mining*, ICML 2003.

[26] M. Schultz and T. Joachims, Learning a Distance Metric from Relative Comparisons, Proceedings of the Conference on Advance in Neural Information Processing Systems (NIPS), 2003.

 Clustering with Constraints

References - 4

[27] Segal, E., Wang, H., and Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. Bioinformatics, 19.

[28] A. Strehl, J. Ghosh, R. Mooney. Impact of similarity measures on web-page clustering. AAAI Workshop on AI for Webpage Search, Austin, pp. 58-64, 2000.

[29] K. Wagstaff and C. Cardie, "Clustering with Instance- Level Constraints", Proc. 17th Intl. Conf. on Machine Learning (ICML 2000), Stanford, CA, June-July 2000, pp. 1103-1110.

[30] K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl, "Constrained K-means Clustering with Background Knowledge", *ICML 2001*.

[31] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. NIPS 15, 2003

[32] Z. Zhang, J.T. Kwok, D.Y. Yeung. Parametric distance metric learning with label information. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03), pp.1450-1452, Acapulco, Mexico, August 2003.

[33] S. Zhong and J. Ghosh. Scalable, model-based balanced clustering. In SIAM International Conference on Data Mining (SDM-03), pp.71-82, San Francisco, CA, 2003.