Τεχνικές Συσταδοποίησης με βάση περιορισμούς

Μαρία Χαλκίδη
Τμήμα Ψηφιακών Συστημάτων
Παν. Πειραιά

Introduction to Semi-supervised learning

- **Clustering (unsupervised learning)** is applicable in many real life scenarios
  - there is typically a large amount of **unlabeled data** available.

- The notion of **good clustering** is strictly ***related to the application domain*** and the ***users perspectives***.

- The use of **user input** is critical for
  - the success of the clustering process
  - the evaluation of the clustering accuracy.

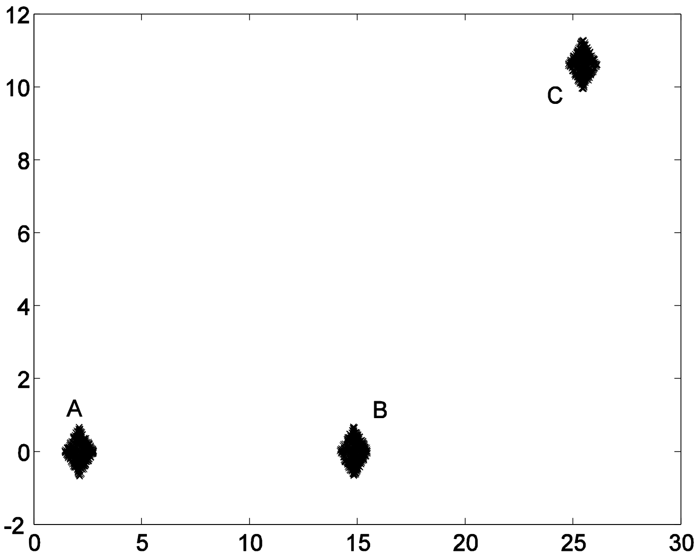
- **User input** is given as
  - Labeled data or Constraints

Motivating semi-supervised learning (I)

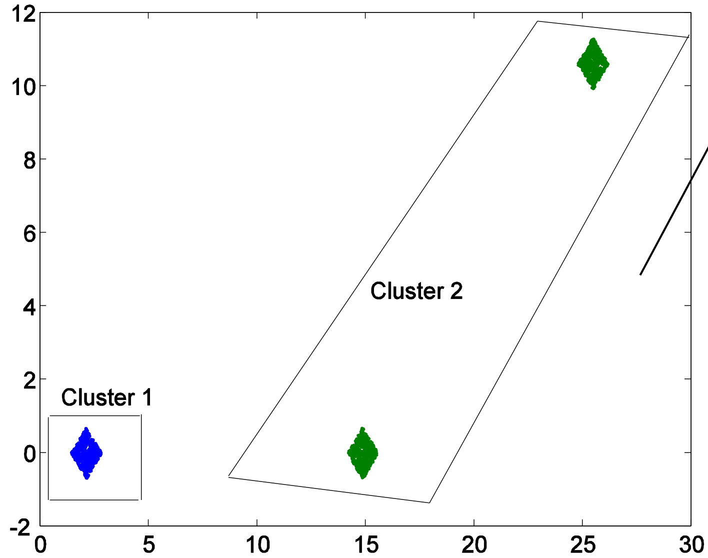
- **Data are correlated.** To recognize clusters, a distance function should reflect such correlations.

- **Traditional clustering methods** fail leading to meaningless results in the case of high-dimensional data
  - **lack of clustering tendency** in a part of the defined subspaces or
  - the **irrelevance of some data dimensions** (i.e. attributes) to the application aspects and user requirements

Learning approaches that use
labeled data/constraints + ***unlabeled*** data
have recently attracted the interest of researchers

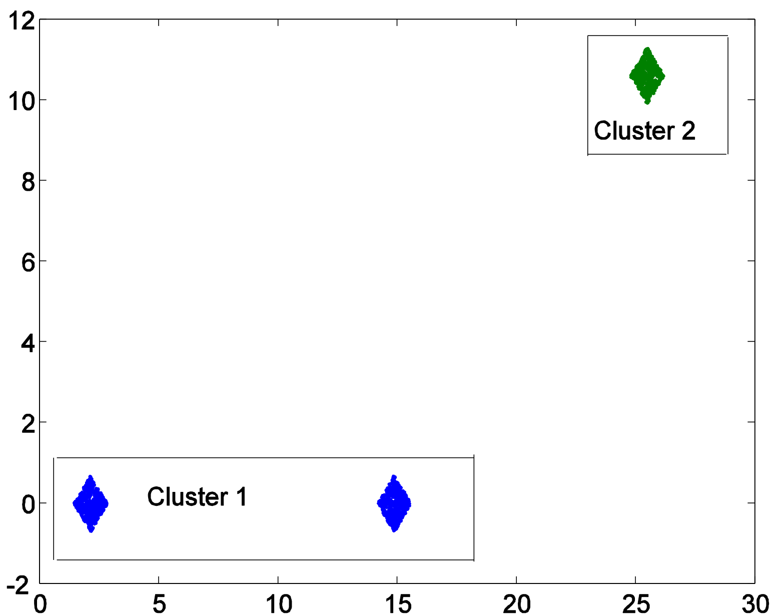


(a)



(b)

a user may want the points in B and C to belong to the same cluster



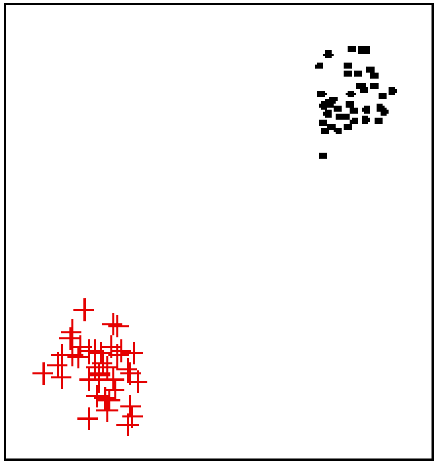
(c)

→ The right clustering may depend on the user's perspective.

→ Fully automatic techniques are very limited in tackling this problem

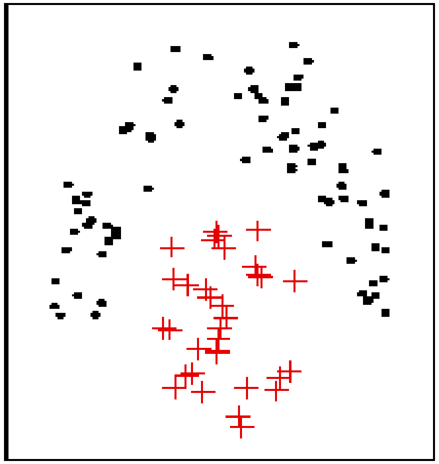
Patterns in Feature Space

- When can we use constraints?



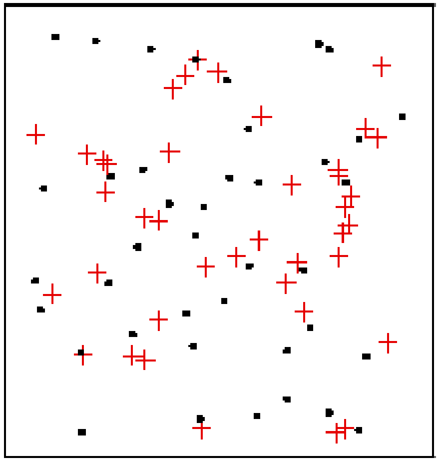
TOO EASY

Don't need constraints



JUST RIGHT

Constraints effective



TOO HARD

Can't use constraints

Clustering under constraints

- Use **constraints** to
  - learn a distortion/distance function
    - Points surrounding a pair of **must-link/cannot-link** points should be close to/far from each other
  - guide the algorithm to a useful solution
    - Two points should be in the same/different clusters

Defining the constraints

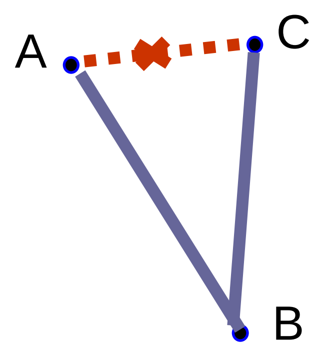
- A set of points $X = \{x_1, \ldots, x_n\}$ on which sets of ***constraints*** have been defined.

- **Must-link constraints**
  - **S:** $\{(x_i, x_j)$ in $X\}$: x_i and x_j **should belong** to the same cluster
- **Cannot-link constraints**
  - **D:** $\{(x_i, x_j)$ in $X\}$: x_i and x_j **cannot belong** to the same cluster

- **Conditional constraints**
  - **δ-constraint:** the distance between any pair of points in two different clusters to be at least ***δ***
  - **ε-constraint:** Any node x should have an ***ε***-neighbor in its cluster

Clustering with constraints: Feasibility issues

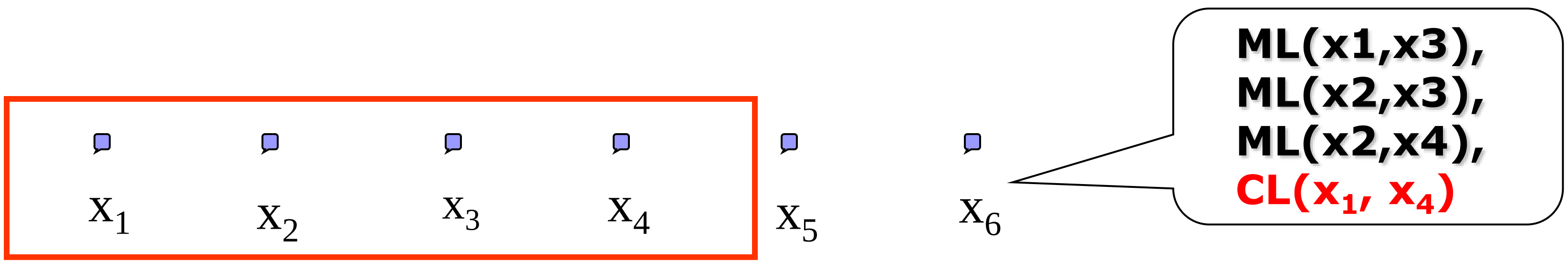
- **Constraints** provide information that should be satisfied.
- Options for **constraint-based clustering**
  - **Satisfy all constraints**
    - **Not always possible:** A with B, B with C, C not with A.



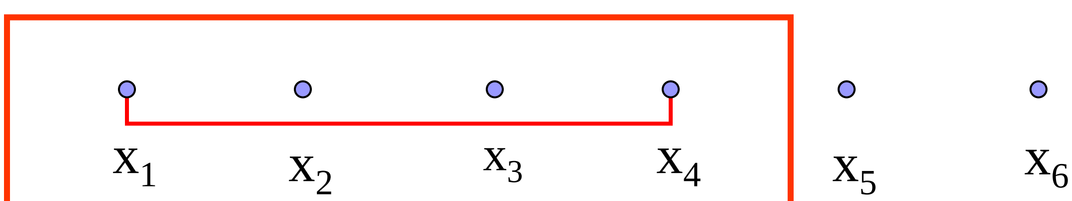
    - **Satisfy as many constraints as possible**
  - Any combination of constraints involving <u>cannot-link constraints</u> is generally computationally intractable (Davidson & Ravi, ISMB 2000),

Feasibility under *Must-link(ML)* and *Cannot-link(CL)* *constraints*

Form the clusters implied by the ML={CC_1 … CC_r} constraints → Transitive closure of the ML constraints



Construct Edges {E} between Nodes based on CL



Infeasible: iff $\exists h, k : e_h(x_i, x_j) : x_i, x_j \in CC_k$

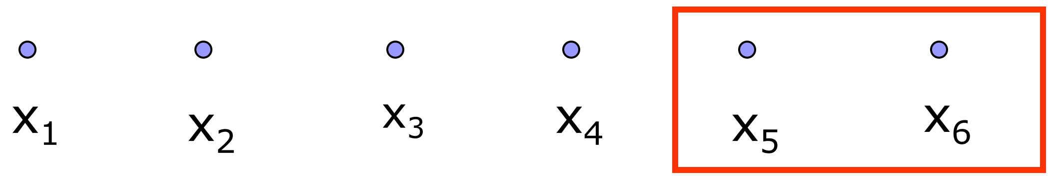
*S. Basu, I. Davidson, tutorial ICDM 2005

Feasibility under *ML* and ε

ε-constraint: Any node **x** should have an **ε**-neighbor in its cluster (another node y such that **D(x,y)≤ ε**)

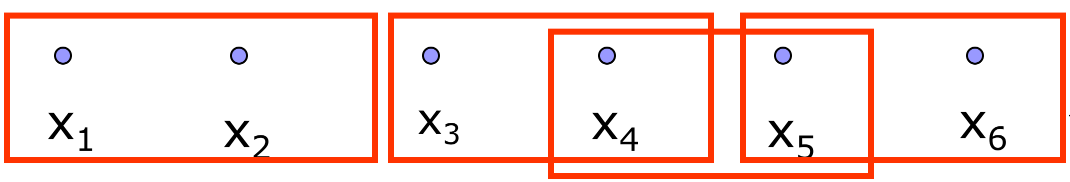
$S' = \{x \in S : x$ *does* ***not*** *have an* ε *neighbor*$\} = \{x_5, x_6\}$

Each of these should be in their own cluster



Compute the ***Transitive Closure*** on ML=$\{CC_1 \dots CC_r\}$

ML(x_1,x_2), ML(x_3,x_4), ML(x_4,x_5)



Infeasible: iff $\exists i, j : x_i \in CC_j, x_i \in S'$

*S. Basu, I. Davidson,turorial ICDM 2005

Clustering based on constraints

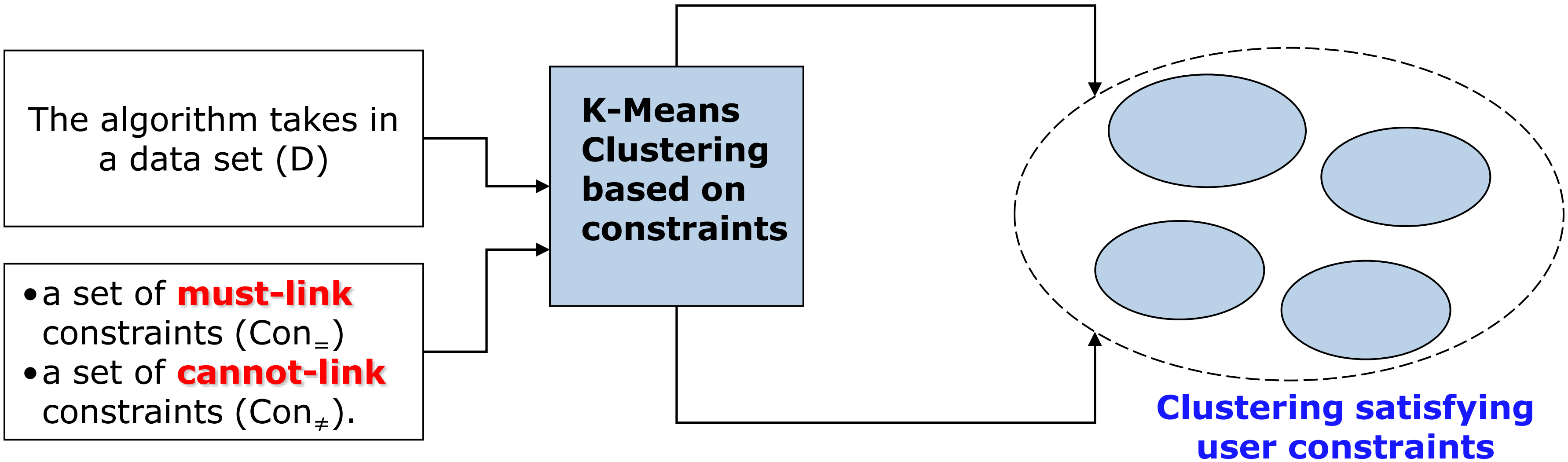
- **Algorithm specific approaches**
  - **Incorporate constraints into the clustering algorithm**
    - COP K-Means **(Wagstaff et al, 2001)**
    - Hierarchical clustering **(I. Davidson, S. Ravi, 2005)**
  - **Incorporate metric learning into the algorithm**
    - MPCK-Means **(Basu et al 2003)**
    - MPCK-Means with local weights **(Bilenko et al 2004)**
    - HMRF K-Means **(Basu et al 2004)**
- **Learning a distance metric** **(Xing et al. '02)**
- **Kernel-based constrained clustering** **(Kulis et al.'05, Yan et al. 2006)**

COP K-Means (I)
[Wagstaff et al, 2001]

- Semi-supervised variant of K-Means
- **Constraints:** Initial background knowledge
- **Must-link** & **Cannot-link** constraints are used in the clustering process
  - Generate a partition that satisfies all the given constraints

K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584, 2001.

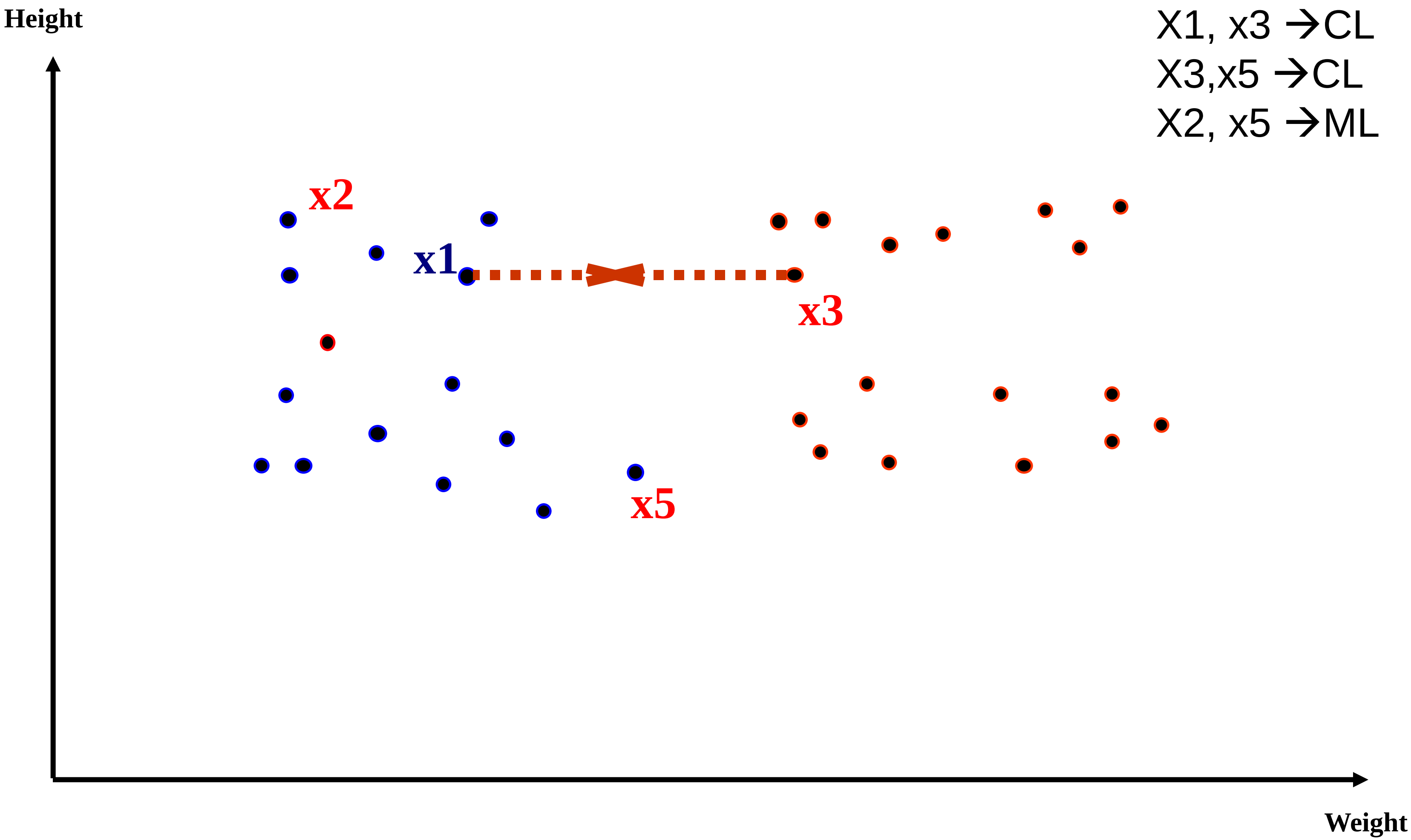
COP K-Means (II)



- **When updating cluster assignments,**
  - we ensure that none of the specified constraints are violated.
- **Assign each point d_i to its closest cluster C_j.** This will succeed unless a constraint would be violated.
  - If there is another point $d_{=}$ that must be assigned to the same cluster as d_i, but that is already in some other cluster, or
  - there is another point $d_{\neq}$ that cannot be grouped with d_i but is already in *C*, then d_i cannot be placed in *C*.
- **Constraints are never broken.** If a legal cluster cannot be found for d_i, the empty partition (f_g) is returned.

Example: COP-K-Means

X1, x2 → ML
X1, x3 →CL
X3,x5 →CL
X2, x5 →ML



Hierarchical Clustering based on constraints
[I. Davidson, S. Ravi, 2005]

Instance: A set S of nodes, the (symmetric) distance **d(x,y)≥0** for each pair of nodes x and y and a collection C of constraints

- **Question:** Can we create a dendrogram for S so that all the constraints in C are satisfied?

Davidson I. and Ravi, S. S. "Hierarchical Clustering with Constraints: Theory and Practice", ***In PKDD 2005***

Constraints and Irreducible Clusterings

- 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.

If mergers are not done correctly, the dendrogram may stop prematurely

- $X=\{x_1, x_2, \ldots, x_k\}$, $Y=\{y_1, y_2, \ldots, y_k\}$, $Z=\{z_1, z_2, \ldots, z_k\}$, $W=\{w_1, w_2, \ldots, w_k\}$

- **CL-constraints**
  - $\forall\{x_i, x_j\}, i \neq j$
  - $\forall\{w_i, w_j\}, i \neq j$
  - $\forall\{y_i, z_j\}, i \leq j, j \leq k$

- Feasible clustering with 2k clusters:

$\{x_1, y_1\}, \{x_2, y_2\}, \ldots, \{x_k, y_k\}, \{z_1, w_1\}, \{z_2, w_2\}, \ldots, \{z_k, w_k\}$

But then get stuck

- **Alternative is:**

$\{x_1, w_1, y_1, y_2, \ldots, y_k\}, \{x_2, w_2, z_1, z_2, \ldots, z_k\}, \{x_3, w_3\}, \ldots, \{x_k, w_k\}$

Using constraints for hierarchical clustering

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

MPCK-Means
[Basu et al 2003]

- **Incorporate metric learning directly into the clustering algorithm**
  - Unlabeled data influence the metric learning process
- **Objective function**
  - <u>Sum of total square distances</u> between the points and cluster centroids
  - <u>Cost of violating</u> the pair-wise constraints

S. Basu, M. Bilenko, R. Mooney. "Comparing and Unifying Search-Based and Similarity-Based Approaches to Semi-Supervised Clustering". *Proceedings of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining Systems, 2003*

Unifying constraints and Metric learning

Generalized K-means distortion function

$$J_{mpckm} = \sum_{x_i \in X} \left\| x_i - \mu_{l_i} \right\|_A^2 - \log(\det(A)) +$$

$$\sum_{(x_i, x_j) \in M} w_{ij} f_M(x_i, x_j) 1[l_i \neq l_j] + \sum_{(x_i, x_j) \in C} \overline{w}_{ij} f_C(x_i, x_j) 1[l_i = l_j]$$

Violation must-link constraints

$$f_M(x_i, x_j) = \left\| x_i - x_j \right\|^2$$

Penalty functions

Violation cannot-link constraints

$$f_C(x_i, x_j) = \left\| x' - x'' \right\|^2 - \left\| x_i - x_j \right\|^2$$

(x', x'') is the maximally separated pair of points in the dataset

MPCK-Means approach

Initialization:

- Use neighborhoods derived from constraints to initialize clusters

Repeat until convergence:

1. **E-step:**
   - **Assign** each point *x* to a cluster *to minimize*
     - **distance of x from the cluster centroid + constraint violations**
2. **M-step:**
   - **Estimate** cluster centroids μ_{li} as means of each cluster
   - **Re-estimate** parameters *A (dimension weights)* of D_A to minimize constraint violations

$$\frac{\partial J_{mpckm}}{\partial A} = 0 \longrightarrow A = \Big(\sum_{x_i \in X} (x_i - \mu_{li})(x_i - \mu_{li})^T - \sum_{(x_i, x_j) \in ML} w_{ij} (x_i - x_j)(x_i - x_j)^T 1(l_i \neq l_j) + \sum_{(x_i, x_j) \in CL} \overline{w}_{ij} (x_i - x_j)(x_i - x_j)^T 1(l_i = l_j) \Big)^{-1}$$

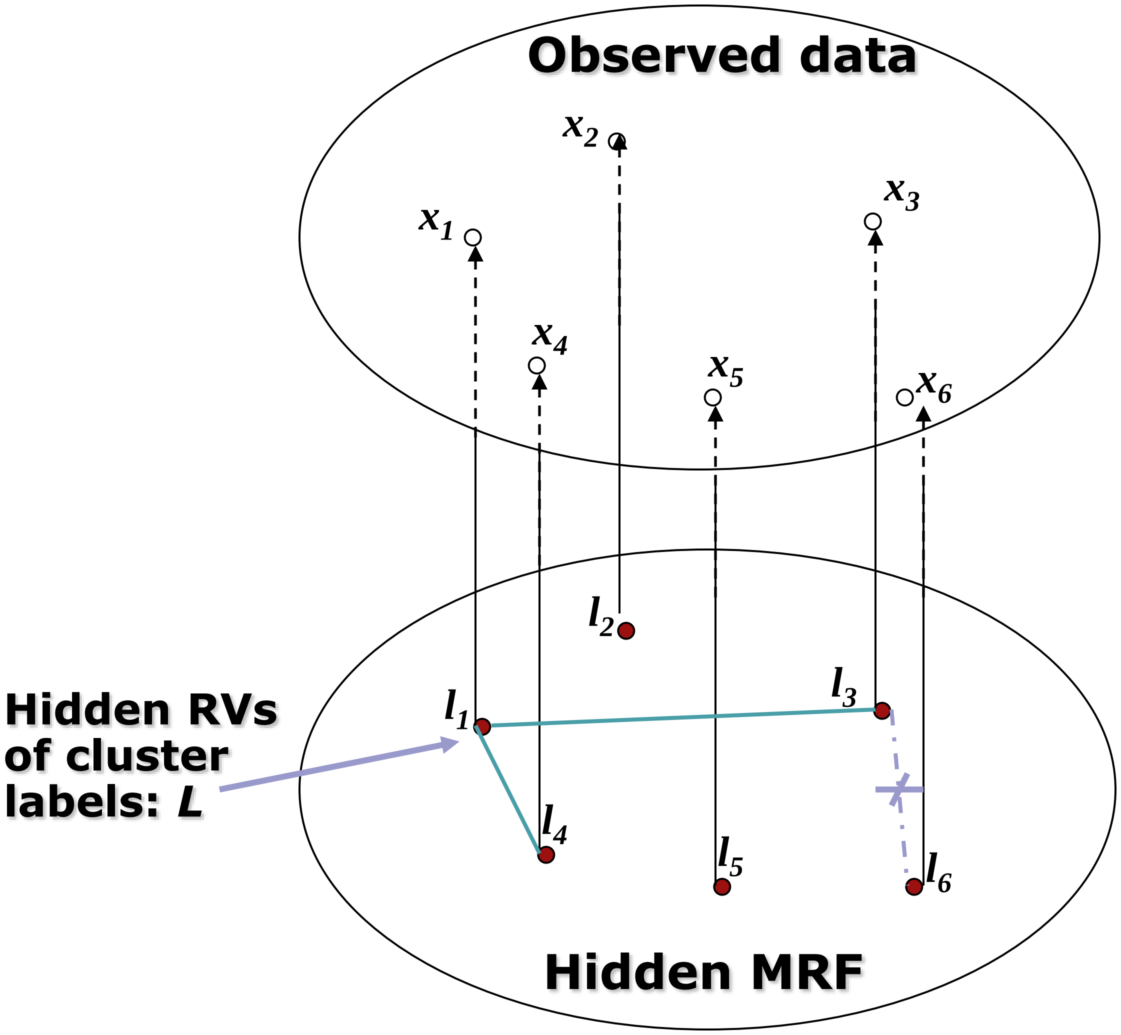
Probabilistic framework for Semi-Supervised Clustering [Basu et al 2004]

- **Hidden Markov Random Fields:**

  Unified probabilistic model that

  - **incorporate pair-wise constraints** along with an underlying distortion measure

Bayesian Approach: HMRF



P(X/L): Conditional probability of the observation set for a given configuration

P(L): Probability distribution of hidden variables

- **Goal of constrained clustering: maximize P(L,X) on HMRF**
- **P(L,X) = P(L)·P(X|L)**

S. Basu, M. Bilenko, R. Mooney. "A Probabilistic Framework for Semi-Supervised Clustering". in Proceedings of the 22th KDD Conference, August 2004

Constrained Clustering on HMRF [Basu et al 2004]

$$Pr(L) = \frac{1}{Z_1} exp[-\sum_i \sum_j V(i,j)]$$

normalizing constant

overall label configuration

Constraint potentials

$$Pr(X \mid L) = \frac{1}{Z_3} exp[-\sum_{x_i} D(x_i, \mu_{l_i})]$$

Cluster distortion

$$\Downarrow$$

Joint probability

$$Pr(L, X) = Pr(X \mid L) \cdot Pr(L)$$

$$-log\, Pr(L, X) = \left(\sum_{x_i} D(x_i, \mu_{l_i}) + \sum_i \sum_j V(i,j) \right)$$

Overall objective of constrained clustering

MRF potential

- Generalized Potts potential:

Cost of violating must/cannot link constraint

$$V(i, j) = \begin{cases} w_{ij} D_A(x_i, x_j) & \text{if} \quad l_i \neq l_j, (x_i, x_j) \in ML \\ \overline{w_{ij}} \left[D_{A,\max} - D_A(x_i, x_j) \right] & \text{if} \quad l_i = l_j, (x_i, x_j) \in CL \\ 0 & \text{otherwise} \end{cases}$$

HMRF-KMeans: Objective Function

K-Means distortion

Must Link violation: constraint-based

$$J_{HMRF} = \sum_{s_i \in S} D_A(x_i, \mu_{l_i}) + \sum_{\substack{(x_i, x_j) \in ML \\ s.t.\, l_i \neq l_j}} w_{ij} D_A(x_i, x_j)$$

$$+ \sum_{\substack{(x_i, x_j) \in CL \\ s.t.\, l_i = l_j}} \overline{w_{ij}} (D_{A,max} - D_A(x_i, x_j))$$

Cannot Link violation: constraint-based

Penalty scaling function: distance-based

-log P(X|L)

-log P(L)

Learning a distance metric based on user constraints

- In **semi-supervised clustering** the requirement is :
  - **learn the distance measure** to satisfy user constraints.

- **Learning a distance** measure → different weights are assigned to different dimensions
  - **Map data to a new space** where user constraints are satisfied

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

- **Goal: Learn a distance metric** between the points in X that satisfies the given constraints
- The problem reduces to the following **optimization problem :**

$$\boxed{\min_A \sum_{(x_i, x_j) \in ML} \left\| x_i - x_j \right\|_A^2}$$

given that

$$\sum_{(x_i, x_j) \in CL} \left\| x_i - x_j \right\|_A \geq 1 \quad A \geq 0$$

E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, December 2002.

Learning Mahalanobis distance

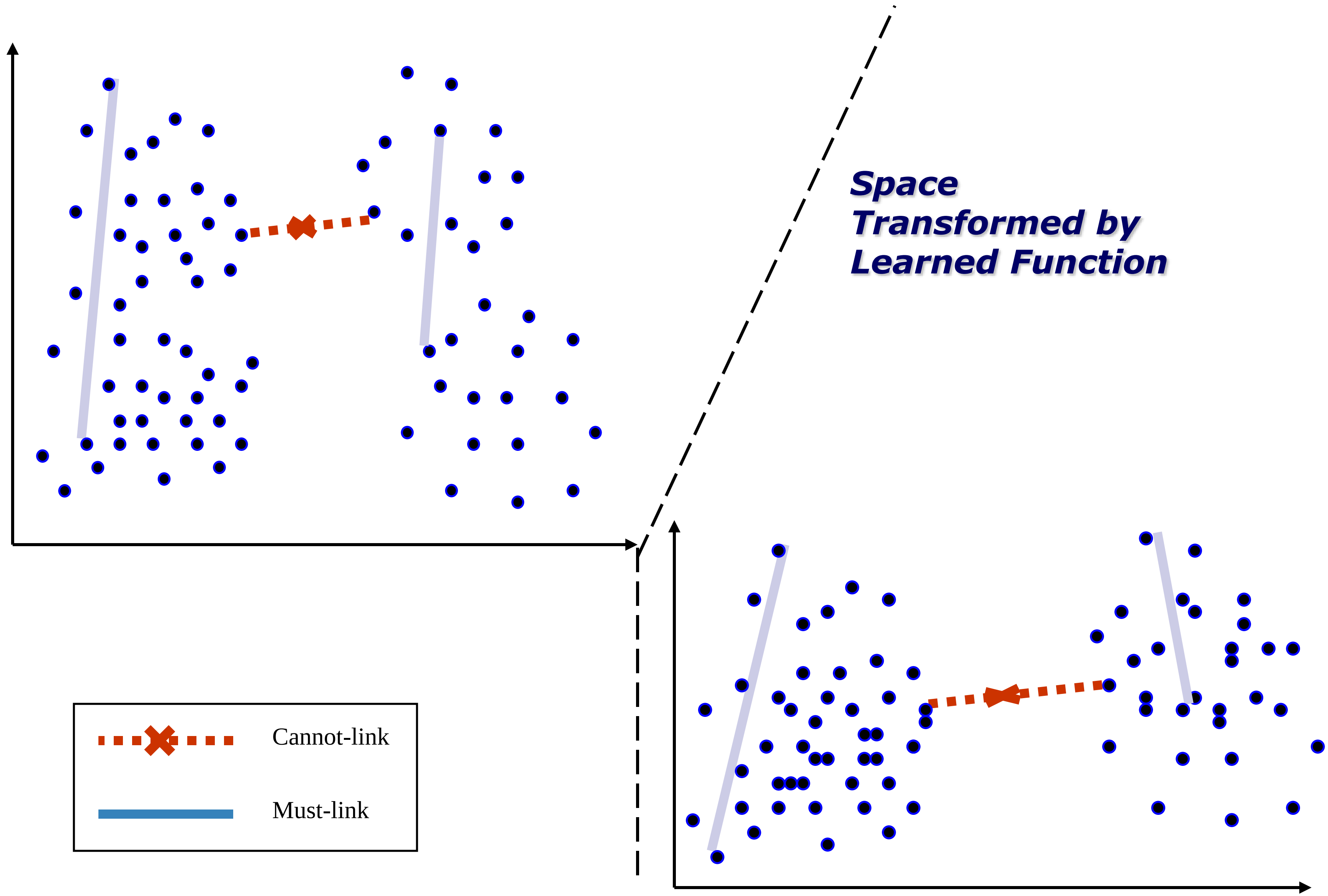
Mahalanobis distance =

Euclidean distance parameterized by matrix A

$$\| x - y \|_A^2 = (x - y)^T A (x - y)$$

Typically ***A*** is the covariance matrix, but we can also learn it given constraints

Example: Learning Distance Function



The Diagonal *A* Case

- Considering the case of learning ***a diagonal*** A
- we can solve the original ***optimization problem*** using Newton-Raphson to efficiently optimize the following

$$\boxed{g(A) = \sum_{(x_i, x_j) \in ML} \|x_i - x_j\|_A^2 - \log\left(\sum_{(x_i, x_j) \in CL} \|x_i - x_j\|_A \right)}$$

Use **Newton Raphson Technique**:

$$x' = x - g(x)/g'(x)$$

$$g(A') = A - g(A).J^{-1}(A)$$

Full *A* Case: Alternative Formulation

- Equivalent **optimization problem**

$$\max_{A} g(A) = \sum_{(s_i, s_j) \in CL} \| x_i - x_j \|_A$$

$$\text{s.t.} \quad f(A) = \sum_{(s_i, s_j) \in ML} \| x_i - x_j \|_A^2 \leq 1 \quad : \quad C_1$$

$$A \geq 0 \quad : \quad C_2$$

Optimization Algorithm - Full *A* Case

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

Space of all positive semi definite matrices

Minimizing a quadratic objective subject to single linear constraint → O(n²)

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

Project A into sets C_1 and C_2

Gradient step

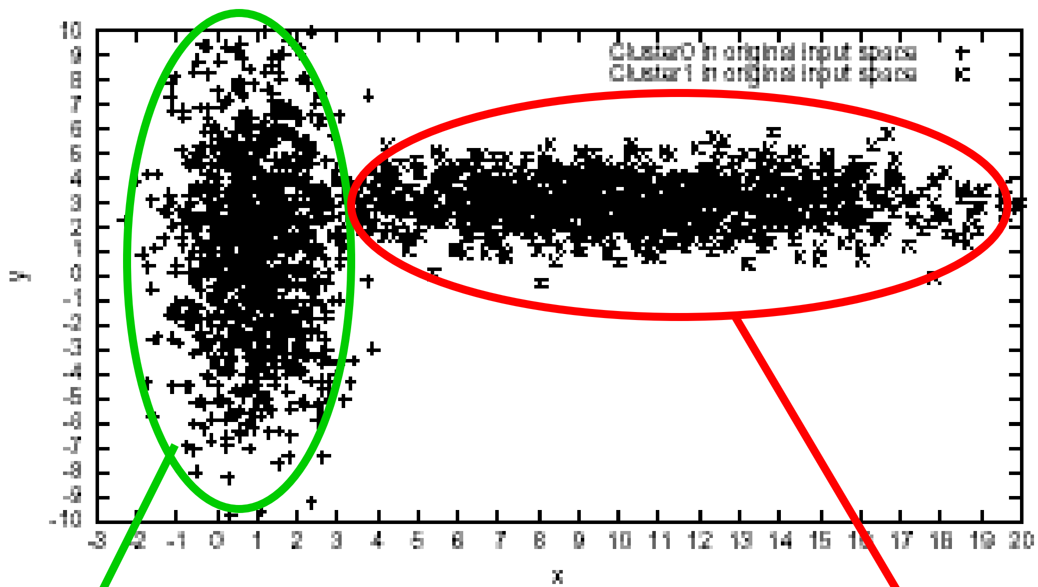
Semi-supervised clustering:
Global vs local weights learning

- **Weights of dimensions** are trained to
  - minimize the distance between must-linked instances and maximize cannot-linked instances

- **Limitation:**
  - Assume a single metric for all clusters
  - preventing clusters from having different shapes

Locally Adaptive Clustering

Each cluster is characterized by different attribute weights (Friedman and Meulman 2002, Domeniconi 2002)



$$\boxed{(w_{1x}, w_{1y}), \;\; w_{1x} > w_{1y}}$$

$$\boxed{(w_{2x}, w_{2y}), \;\; w_{2y} > w_{2x}}$$

Semi-supervised clustering using local weights

- **Solution:**
  - Allow a separate weight matrix, $\mathbf{A_h}$, for each cluster **h**
  - Cluster **h** is generated by a Gaussian with covariance matrix $\mathbf{A_h}^{-1}$

$$J_{mkmeans} = \sum_{x_i \in X} \left(\left\| x_i - \mu_{l_i} \right\|_{A_{l_i}}^2 - \log\left(\det\left(A_{l_i}\right)\right) \right)$$

- Generalized version of K-Means using different weights per cluster:

MPC-KMeans with local weights

K-Means distortion

Must Link violation: constraint-based

$$J_{MPCKM} = \sum_{s_i \in S} \left(\|x_i - \mu_{l_i}\|^2_{A_{l_i}} - \log\left(\det\left(A_{l_i}\right)\right) \right) + \sum_{(x_i, x_j) \in M} w_{ij}\, f_M(x_i, x_j)\, \mathbb{1}[l_i \neq l_j]$$

$$+ \sum_{\substack{(x_i, x_j) \in CL \\ s.t.\, l_i = l_j}} \overline{w_{ij}}\, f_C(x_i, x_j)\, \mathbb{1}[l_i = l_j]$$

Cannot Link violation: constraint-based

Penalty function: distance-based

$$f_M(x_i, x_j) = \frac{1}{2}\|x_i - x_j\|^2_{A_{l_i}} + \frac{1}{2}\|x_i - x_j\|^2_{A_{l_j}}$$

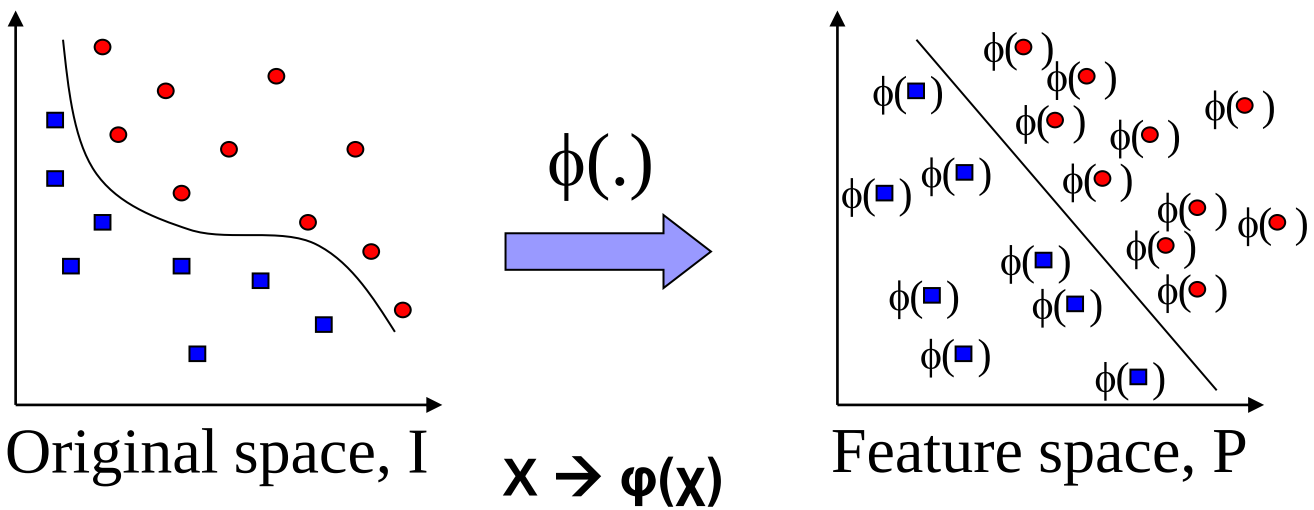
$$f_C(x_i, x_j) = \|x'_{l_i} - x''_{l_i}\|^2_{A_{l_i}} - \|x_i - x_j\|^2_{A_{l_i}}$$

(x'_{li}, x''_{li}) is the maximally separated pair of points in the dataset according to the l_i–metric metric.

Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In p*roceedings of the 21st International Conference on Machine Learning* (ICML-2004), Banff, Canada, July 2004.

Kernel-based learning methods- Main Idea

- **Kernel Methods** work by:
  - embedding data in a vector space, P
  - looking for (linear) relations in such space
- Much of the geometry of the data in the embedding space (relative positions) is contained in all pairwise inner products
- **Kernel trick:** $K(x, y) = \varphi(x) \cdot \varphi(y)$
  - The distance computation in P can be efficiently performed in input space, I.



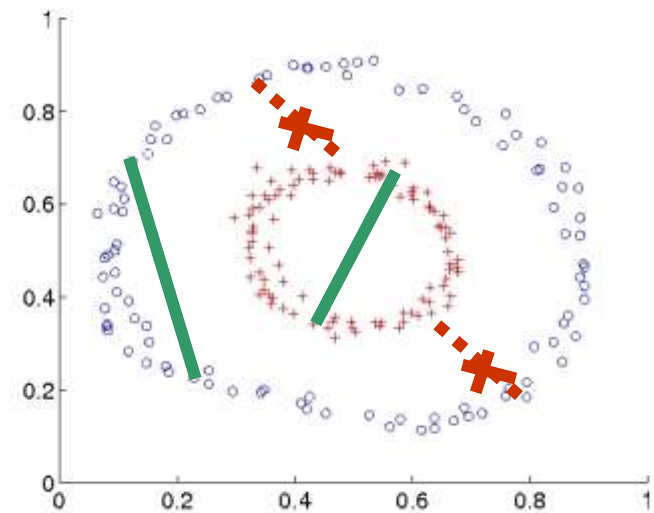
Kernel based Semi-supervised clustering
[Kulis et al.'05]

A non-linear transformation, φ

- maps data to a high dimensional space
- the data are expected to be more separable
- a kernel function **k (x, y)** computes **φ(x)·φ(y)**

The user gives constraints

The appropriate kernel is created based on constraints



$$J\left(\{\pi\}_{c=1}^{k}\right)=\sum_{c=1}^{k}\sum_{x_i\in\pi_c}\left\|\phi(x_i)-m_c\right\|^2-\sum_{\substack{x_i,x_j\in ML\\ l_i=l_j}}w_{ij}+\sum_{\substack{x_i,x_j\in CL\\ l_i=l_j}}w_{ij}$$

$$\left\|\phi(x_i)-m_c^{\phi}\right\|=A_{ii}+B_{cc}-D_{ic}$$

$$A_{ii}=\phi(x_i)\cdot\phi(x_i)=1$$

$$B_{cc}=\frac{1}{|\pi_C|^2}\sum_{x_j,x_{j'}\in\pi_C}\phi(x_j)\cdot\phi(x_{j'})=\frac{1}{|\pi_C|^2}\sum_{x_j,x_{j'}\in\pi_C}K(x_j,x_{j'})$$

$$D_{ic}=\frac{2}{|\pi_C|}\sum_{x_j\in\pi_C}\phi(x_i)\cdot\phi(x_j)=\frac{2}{|\pi_C|}\sum_{x_j\in\pi_C}K(x_i,x_j)$$

Kernel for HMRF-KMeans with squared Euclidean distance

Center of cluster *c*

Points in cluster l_i

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

Input similarity matrix

Constraint similarity matrix

Cluster size weighted penalties

$$K = S + W,$$

$$\text{where} \begin{cases} S_{ij} = x_i \cdot x_j\,, \ \text{input similarity matrix}, \\ W_{ij} = \begin{matrix} + w_{ij} \ \text{if} \ (x_i, x_j) \in ML \\ - w_{ij} \ \text{if} \ (x_i, x_j) \in CL \end{matrix} \end{cases}$$

Trace($ZT^K Z$),

Now define a matrix Z such that $Z_{.c}$, the column c of ˜ Z, is equal to $z_c/(z_c^T z_c)1/2$.

Semi-Supervised Kernel-KMeans

[Kulis et al.'05]

- **Algorithm:**
  - Constructs the appropriate kernel matrix from data and constraints
  - Runs weighted kernel K-Means
- **Input of the algorithm:** Kernel matrix
  - Kernel function on vector data or
  - Graph affinity matrix
- **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

Adaptive Kernel-based Semi-supervised Clustering [Yan, Domeniconi, ECML06]

- **Kernel function** affects the **quality of clustering results**
- **Critical problem:**
  - learn kernel's parameter based on the data and the given constraints (must- and cannot-link)
  - **Integrate constraints into the clustering objective function**
  - Optimize the kernel parameter iteratively during the clustering process.

B. Yan, C. Domeniconi. An Adaptive Kernel Method for Semi-Supervised Clustering. ECML 2006, Berlin, Germany

Adaptive-SS-Kernel-KMeans

Distance from the cluster centroid

$$J_{kernel\_obj} = \sum_{s_i \in S} \left(\left\| \phi(x_i) - m_c^{\phi} \right\|^2 \right) + \sum_{(x_i, x_j) \in M} w_{ij} \left\| \phi(x_i) - \phi(x_j) \right\|^2$$

Must Link violation: constraint-based

$$+ \sum_{\substack{(x_i, x_j) \in CL \\ s.t. l_i = l_j}} \overline{w_{ij}} \left((D_{max}^{\phi})^2 - \left\| \phi(x_i) - \phi(x_j) \right\|^2 \right)$$

Cannot Link violation: constraint-based

Penalty function: distance-based

$$\left\| \phi(x_i) - m_c^{\phi} \right\| = A_{ii} + B_{cc} - D_{ic}$$

$$A_{ii} = \phi(x_i) \cdot \phi(x_i) = 1 \qquad B_{cc} = \frac{1}{|\pi_C|^2} \sum_{x_j, x_{j'} \in \pi_C} \phi(x_j) \cdot \phi(x_{j'}) = \frac{1}{|\pi_C|^2} \sum_{x_j, x_{j'} \in \pi_C} K(x_j, x_{j'})$$

$$D_{ic} = \frac{2}{|\pi_C|} \sum_{x_j \in \pi_C} \phi(x_i) \cdot \phi(x_j) = \frac{2}{|\pi_C|} \sum_{x_j \in \pi_C} K(x_i, x_j)$$

Gaussian Kernel: $k(x, x') = e^{-\|x - x'\|^2 / \sigma^2}$

Algorithm: Adaptive-SS-Kernel-KMeans

- **Initialize clusters** using the given constraints;

  t=0

- **E-step:** Assign each data point x_i to a cluster $\pi_c^{(t)}$ so that J_{kernel_obj} is minimized
- **M-step(1):** Re-compute $B_{cc}^{(t)}$

$$B_{cc} = \frac{1}{|\pi_C|^2} \sum_{x_j, x_{j'} \in \pi_C} K(x_j, x_{j'})$$

- **M-step(2):** Optimize the kernel parameter using the gradient descent according to the rule:

$$\sigma^{(new)} = \sigma^{(old)} - \rho \frac{\partial J_{kernel\_obj}}{\partial \sigma}$$

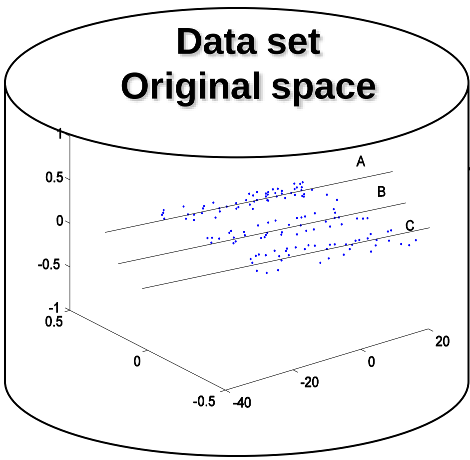
- t=t+1

Clustering based on constraints & cluster validity criteria

- Different distance metrics may satisfy the same number of constraints
- One solution is to apply a different criterion that evaluates the resulting clustering to choose the right distance metric
- A general approach should:
  - Learn an appropriate distance metric to satisfy the constraints
  - Determine the best clustering w.r.t the defined distance metric.

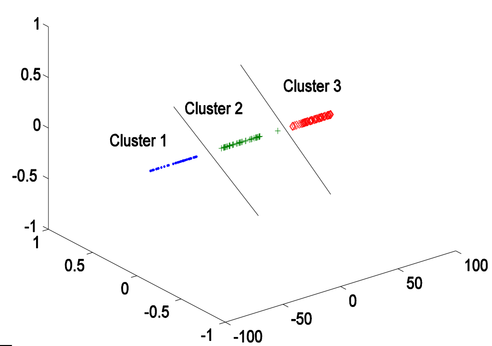
Semi-supervised learning framework

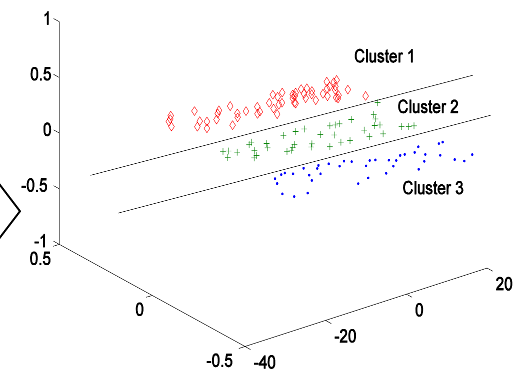
[Halkidi et.al, IEEE ICDM 2005]

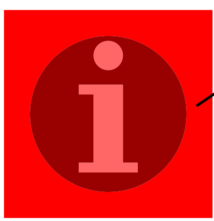


Semi-supervised learning Framework

Learn the space where the best partitioning according to the user constraints can be defined







Constraints **Must-link constraints**

S: {(x_i, x_j) in X }: x_i and x_j should belong to the same cluster

Cannot-link constraints

D: {(x_i, x_j) in X} : x_i and x_j cannot belong to the same cluster

Initializing dimension weights based on user constraints

- Learn the distance measure to satisfy user constraints (must-link and cannot-link).
- Different weights are assigned to different dimensions
- Learn ***a diagonal*** matrix A using Newton-Raphson to efficiently optimize the following equation [Xing et al, 2002]

$$g(A) = \sum_{(x_i, x_j) \in S} \|x_i - x_j\|_A^2 - \log\left(\sum_{(x_i, x_j) \in D} \|x_i - x_j\|_A \right)$$

Best weighting of data dimensions

- **W:** set of different weightings defined for a set of d data dimensions.
- $W_j \in W$ best weighting for a given dataset
  - **if** the clustering of data in the ***d*−dimensional** space defined by

$$W_j = [w_{j1}, \ldots, w_{jd}] \; (w_{ji} > 0)$$

optimizes the quality measure:

$$QoC_{constr}(C_j) = optim_{i=1,\ldots,m}\{QoC_{constr}(C_i)\}$$

given that C_j is the clustering for the W_j weighting vector.

Defining dimension weights

- **Clustering quality criterion (measure) :** evaluates a clustering, C_i, of a dataset in terms of
  - its **accuracy w.r.t. the user constraints** (*ML & CL*)
  - its **validity based on well-defined cluster validity criteria**.

$$\mathbf{QoC_{constr}(C_i) = w \cdot Accuracy_{ML\&CL}(C_i) + ClusterValidity(C_i)}$$

significance of the user constraints w.r.t. the cluster validity criteria

% of constraints satisfied in C_i

***Ci*'s cluster validity.**

Cluster Validity criteria

- ***S_Dbw*** → validity of clustering results in terms of objective criteria

$$\mathbf{S\_Dbw(c) = Scat(c) + Dens\_bw(c)}$$

$$\boxed{ClusterValidity(C_i) = (1+S\_Dbw(C_i))^{-1}}$$

Our approach aims to optimize the following form:

$$QoC_{constr}(C_i) = w \cdot AccuracyS\&D(C_i) + (1+S\_Dbw(C_i))^{-1})$$

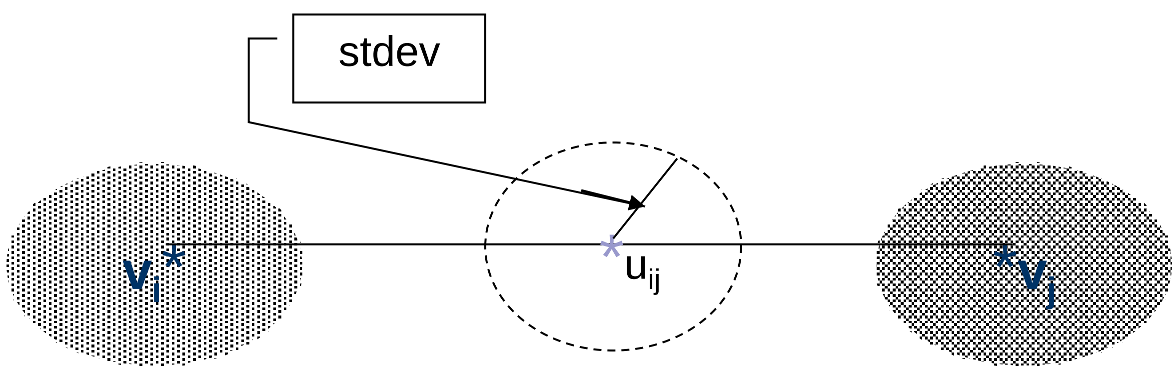
S_Dbw definition: Inter-cluster Density (ID)

Dens_bw: Average density in the area among clusters in relation with the density of the clusters

$$\mathrm{Dens\_bw}(c) = \frac{1}{c \cdot (c-1)} \sum_{i=1}^{c} \left(\sum_{\substack{j=1 \\ i \neq j}}^{c} \frac{\mathrm{density}(u_{ij})}{\max\{\mathrm{density}(v_i), \mathrm{density}(v_j)\}} \right),$$

$$\mathrm{density}(u\ ) = \sum_{l=1}^{n_{ij}} f(x_l, u\ ), \quad f(x,u) = \begin{cases} 0, & \text{if } d(x,u) > \mathrm{stdev} \\ 1, & \text{otherwise} \end{cases}$$

where $n_{ij} =$ *number of tuples that belong to the clusters* c_i *and* c_j, i.e., $x_l \in c_i \cup c_j \subseteq S$



S_Dbw definition: Intra-cluster variance

Average scattering of clusters:

$$\mathrm{Scat}(c) = \frac{\frac{1}{c}\sum_{i=1}^{c}\left\|\sigma(v_i)\right\|}{\left\|\sigma(X)\right\|}$$

where $$\sigma_x^p = \frac{1}{n}\sum_{k=1}^{n}\left(x_k^p - \bar{x}^p\right)^2$$

where $\bar{x}^p$ *is the* p_{th} *dimension of* $\bar{X} = \frac{1}{n}\sum_{k=1}^{n} x_k, \forall x_k \in X$

$$\sigma_{v_i}^p = \sum_{k=1}^{n_i}\left(x_k^p - v_i^p\right)^2 \Big/ n_i$$

Hill climbing procedure: Defining dimension weights

- Initialize dimension weights to satisfy ***S*** and ***D***,

$$W_{cur} = \{W_i \mid i = 1, . . . , d\}$$

- Cl_{cur} ← clustering of data in space defined by W_{cur}.
- **For each dimension i**
  1. Updated W_{cur}. ← Increase or decrease the *i*-th dimension of W_{cur}
  2. Cl_{cur} ← Cluster data in new space defined by W_{cur}.
  3. Quality(W_{cur}) ← $QoC_{constr}(Cl_{cur})$
     - If there is improvement to Quality(W_{cur}) **Go to step** 1
- W_{best}, ← weighting resulting in 'best' clustering (correspond to maximum $QoC_{constr}(Cl_{cur})$)

An iterative semi-supervised learning approach

[Halkidi et.al, ICDM 2005]

Original data

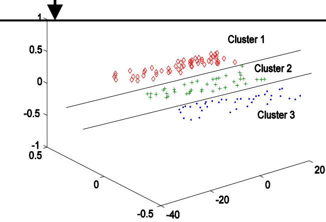
User constraints

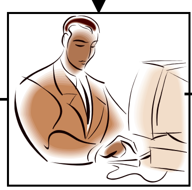
Define dimension weights, W, based on constraints

Optimize weights based on user constraints and validity criteria (Hill climbing)

Cluster data in the new space

Present results to user

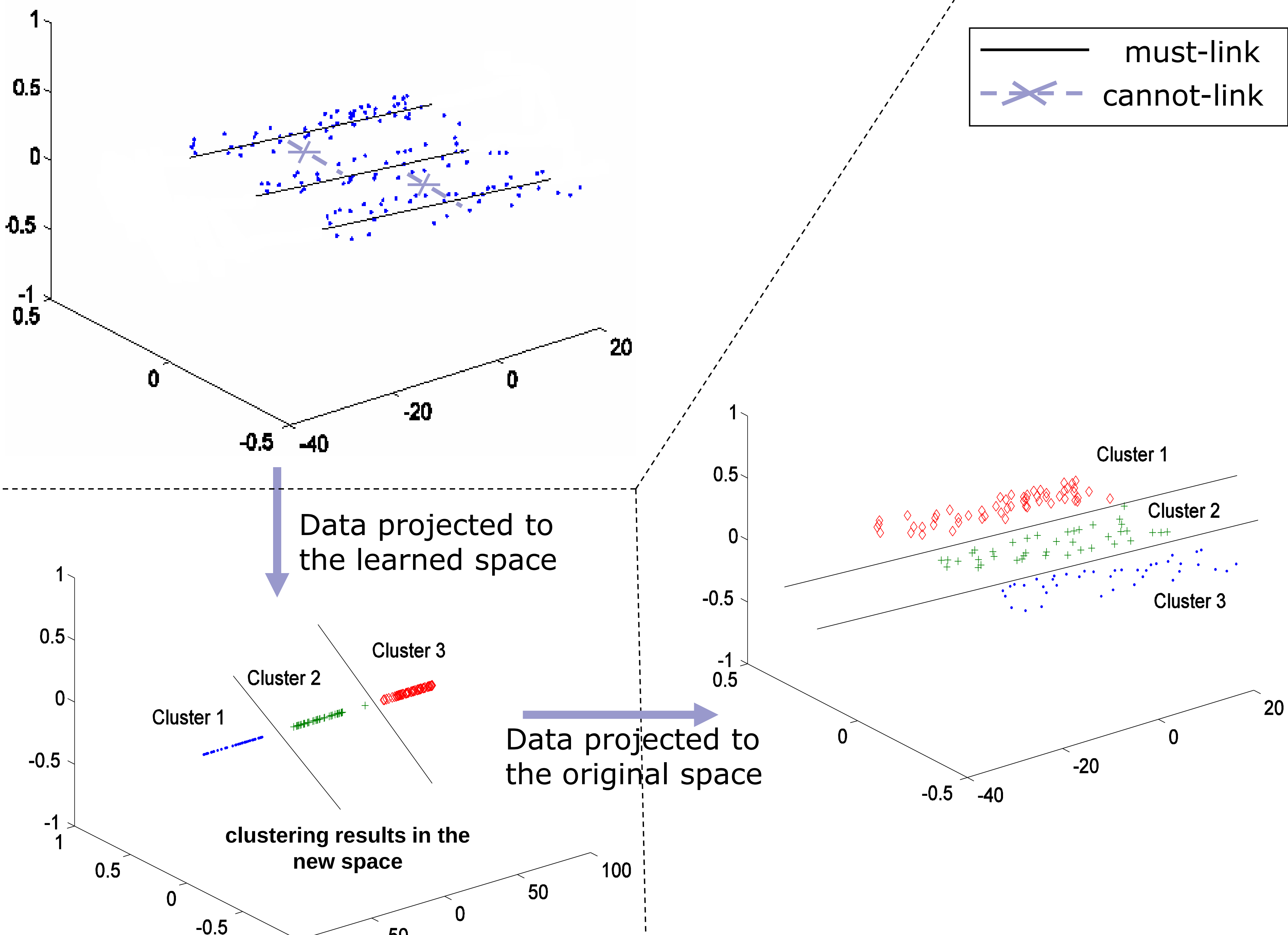




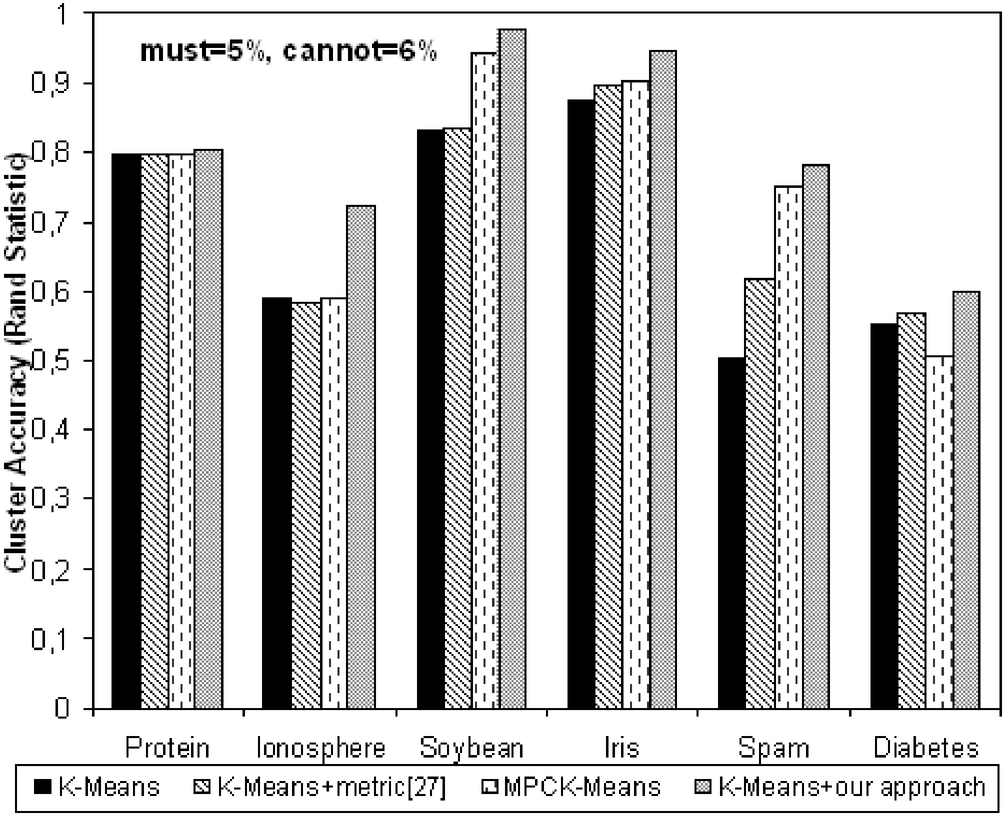
User constraints

Final clustering

M. Halkidi, D. Gunopulos, N. Kumar, M. Vazirgiannis, C. Domeniconi. "A Framework for Semi-supervised Learning based on Subjective and Objective Clustering Criteria". *in the Proceedings of ICDM Conference*, Houston, USA, November 2005


must-link
cannot-link
Data projected to the learned space
Cluster 1
Cluster 2
Cluster 3
clustering results in the new space
Data projected to the original space
Cluster 1
Cluster 2
Cluster 3

Clustering Accuracy on UCI datasets.



The clustering accuracy was averaged over 10 runs using randomly selected constraints (must-link=5% and cannot-link=6% of points).

Our approach achieves on average 12%, 9% and 6% higher clustering accuracy than the Naive K-Means, the Xing et al.'s approach and MPCK-Means, respectively.

UCI repository

Protein(d=20), Ionosphere(d=34),Soybean(d=35), Iris(d=4), Spam(d=57), Diabetes(d=8)

Conclusions & Further research directions

Promising areas in clustering research

- Semi-supervised learning
- Learning similarity measures
- Dimensionality reduction
- Nonlinearly separable clusters

Conclusions & Further research directions

- **Promising techniques**
  - **Model selection techniques**
    - learn the best model for your data (regression, MLE,..)
  - **Advanced similarity measure** learning
    - local weight learning
    - kernel learning

Ευχαριστώ!

References- Semi-supervised learning (1)

- B. Anderson, A. Moore, and D. Cohn. A nonparametric approach to noisy and costly optimization. In ICML, 2000.
- A. Bar-Hillel, T. Hertz, N. Shental, and D.Weinshall. Learning distance function using equivalence relations. In ICML, 2003.
- S. Basu, M. Bilenko, and R. Mooney. “A probabilistic framework for semi-supervised clustering”. In KDD, August 2004.
- M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In ICML, 2004.
- S. Basu, A. Banerjee and R. J. Mooney “Semi-supervised Framework by Seeding” in ICML, 2002.
- P. Bradley, K. Bennet, and A. Demiriz, “Constrainted K-Means Clustering”, Microsoft research Technical report, May 2000.
- A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In Conf. on Computational Learning Theory, pages 92 100, 1998.
- A. Blum J. Laffety, M.R. Rwedebangria, R. Reddy, “Semi-Supervised Learning Using Randomized Mincuts”. In ICML, 2004.
- M. Charikar, V. Guruswami and A. Wirth, “Clustering with Qualitative Information” in Proc. Of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003.

References-Semi-supervised learning (2)

- H. Chang, D.Y. Yeug. “Locally linear metric adaptation for semi-supervised clustering”In ICML 2004.
- D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback. In Technical Report TR2003- 1892, 2003.
- Davidson I. and Ravi, S. S. “Hierarchical Clustering with Constraints: Theory and Practice”, *In, PKDD 2005*
- Davidson I. and Ravi, S. S. “Clustering under Constraints: Feasibility Results and the k-Means Algorithm”, In SDM 2005.
- D. Gondek, S. Vaithyanathan, and A. Garg. “Clustering with Model-level Constraints”In SDM 2005.
- M. Halkidi, D. Gunopulos, N. Kumar, M. Vazirgiannis, C. Domeniconi. “A Framework for Semi-supervised Learning based on Subjective and Objective Clustering Criteria”. in ICDM 2005 .
- D. Klein, S. Kamvar and C. Manning. “From Instance-Level Constraintsto Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering” in ICML 2002.
- B. Kulis, S. Basu, I. Dhillon, R. Mooney. “Semi-sueprvised Graph Clustering: A Kernel Approach”, In ICML, 2005
- M. Law, A. Topchy, A. Jain. “Model-based clustering with Probabilistic Constraints”. In SDM 2005.
- I. Dhillon, Y. Guan & Kulis. “Kernel k-means spectral clustering and normalized cuts”. In KDD, 2004

References- Semi-supervised learning (3)

- Z. Lu, T. Leen. “Semi-supervised Learning with Penalized Probabilistic Clustering”, NIPS 2005.
- W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes in C, The art of Scientific Computing. Cambridge University Press, 1997.
- E. Segal, H. Wang, and D. Koller. Discovering molecular pathways from protein interaction and gene expression data. Bioinformatics, 19:264–272, July 2003.
- B. Stein, S. M. zu Eissen, and F.Wibrock. On cluster validity and the information need of users. In AIA, September 2003.
- Kiri Wagstaff and Claire Cardie. “Clustering with Instance-level Constraints”. In the Proceedings to the ICML Conference, Stanford, June 2000.
- K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl. “Constrained K-Means Clustering with Background Knowledge”. In the Proceeding of the 18th ICML Conference, Massachusetts, June 2001.
- E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In NIPS, December 2002.
- Z. Zhang, J. Kwok, D. Yeung. “Parametric distance metric learning with label information”. In IJCAI, 2003
- Y. Qu, S. Xu. “Supervised cluster analysis for microarray data based on multivariate Gaussian mixture” *Bioinformatics, Vol 20, No 12, 2004.*
- *M. Bilenko, S. Basu, R. Mooney. “Integrating Constraints and Metric Learning in Semi-Supervised clustering”, In ICML 2004, Banff, Canada, July 2004*