APRENDIZAJE SEMI-SUPERVISADO

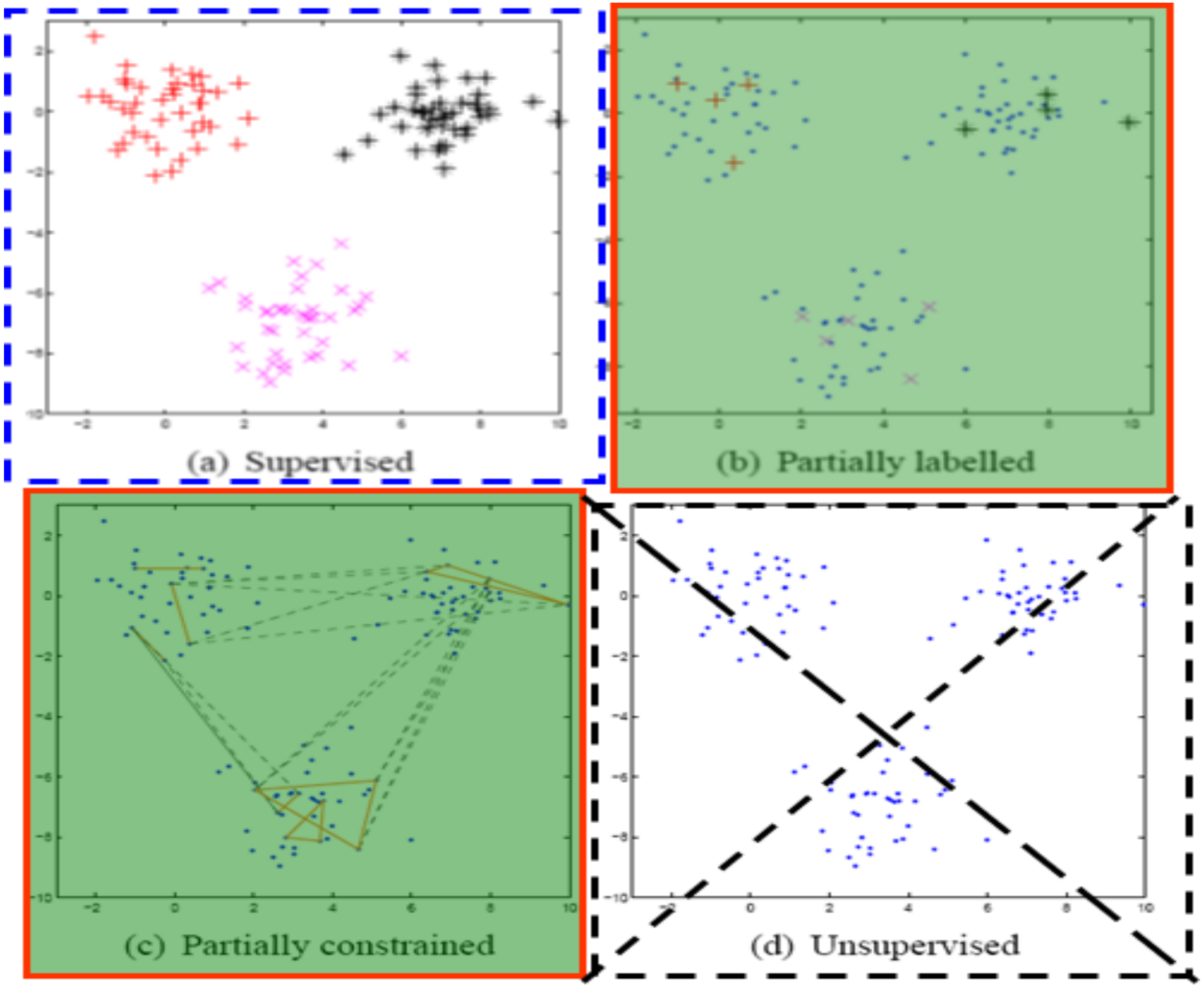
Minería de Datos: Aspectos Avanzados

Salvador García
salvagl@decsai.ugr.es

Objetivos

- Entender el problema del aprendizaje semi-supervisado situarlo en el contexto de la predicción.
- Conocer las diferentes estrategias que se emplean en el campo del aprendizaje semi-supervisado.
- Estudiar diferentes medidas y técnicas de evaluación en aprendizaje semi-supervisado.
- Presentar algunas propuestas clásicas en el enfoque de self-labelling, grafos y label propagation.

Spectrum of Learning Problems



Semi-Supervised Learning: SSL

In many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

Unlabeled data is much cheaper.

Speech

Customer modeling

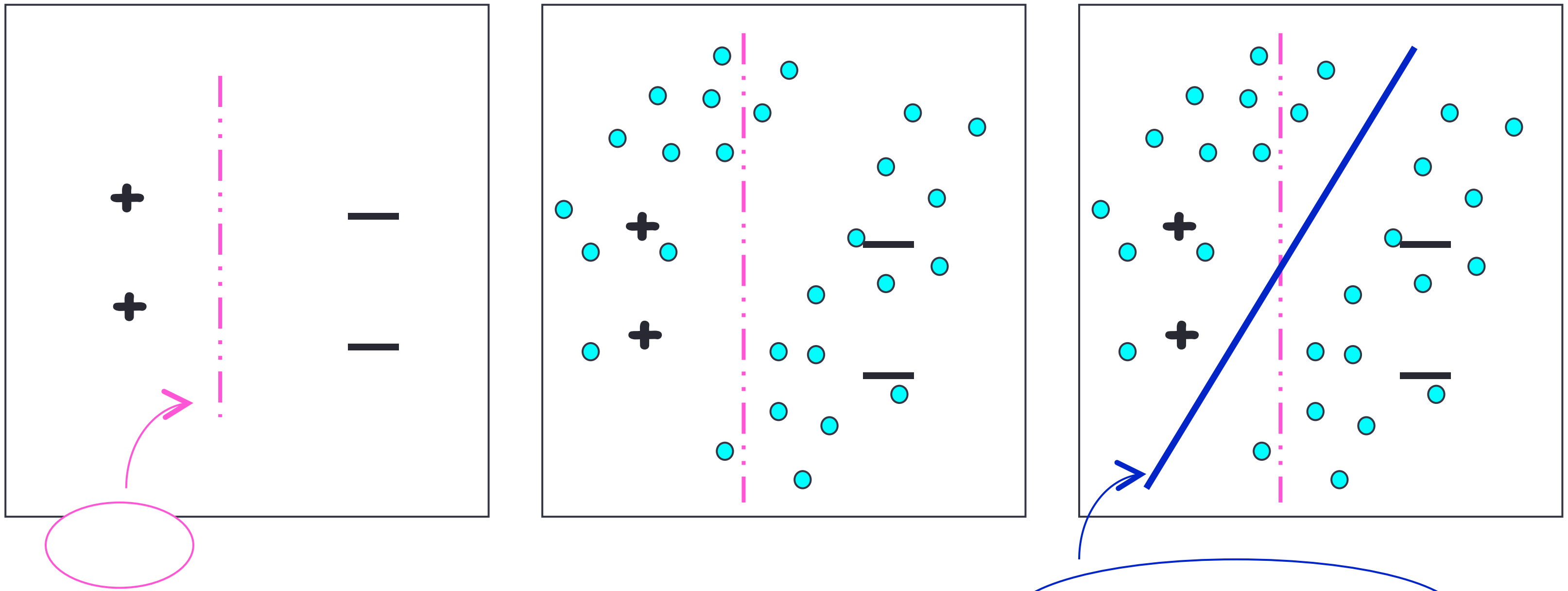
Images

Protein sequences

Medical outcomes

Web pages

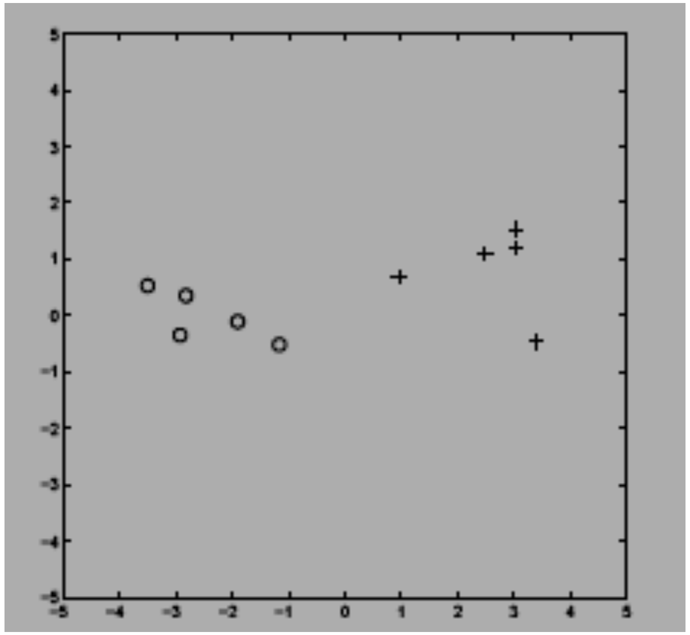
Intuition

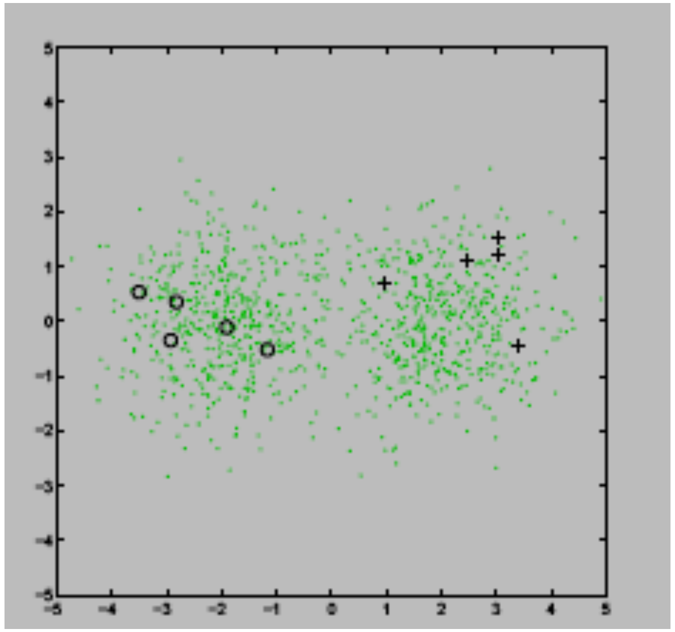


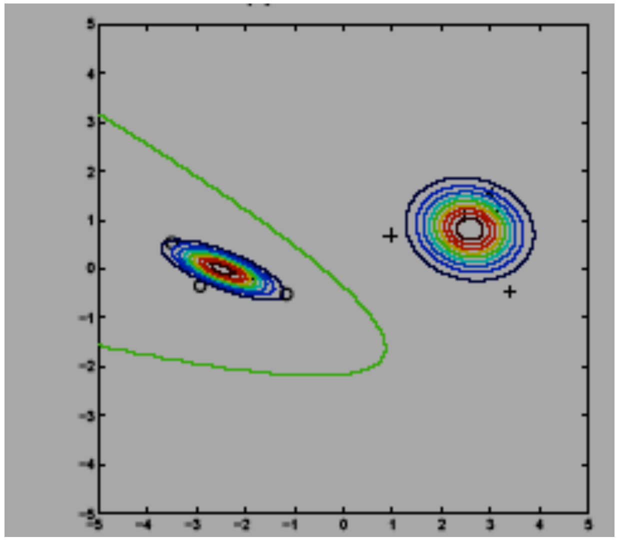
Labeled data only

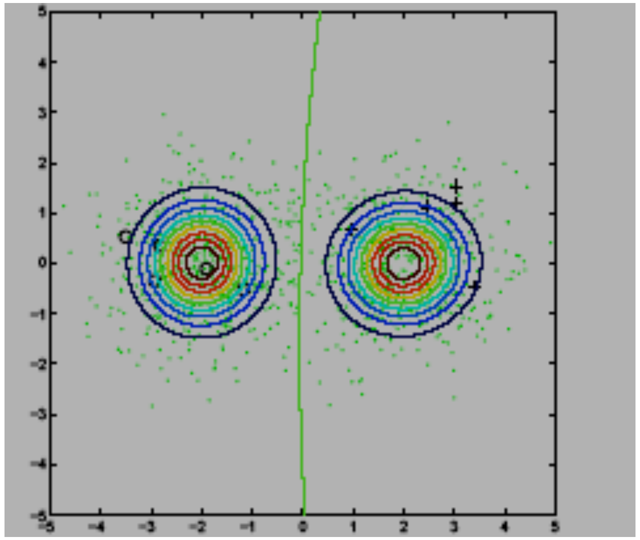
Transductive Learning

Intuition



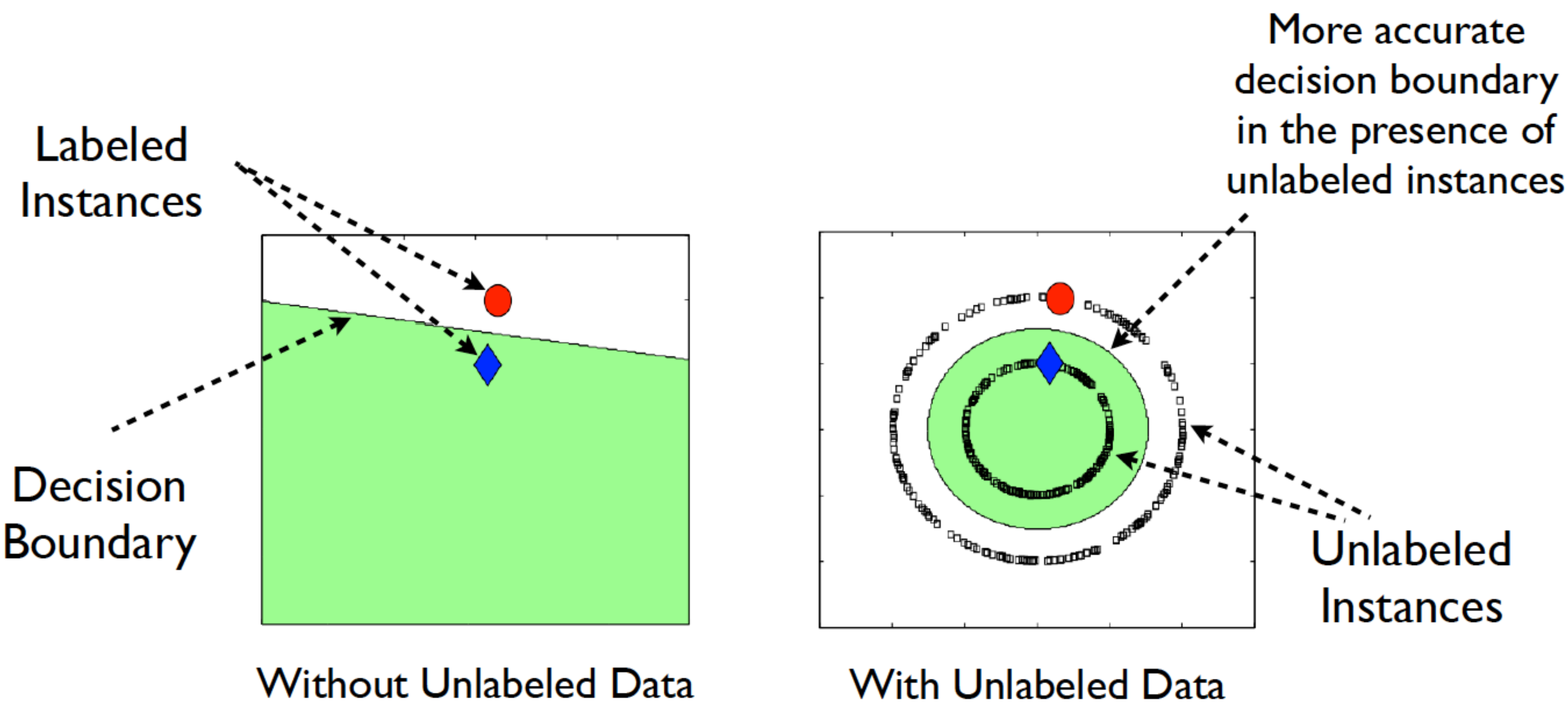






Intuition

How can unlabeled data be helpful?



Example from [Belkin et al., JMLR 2006]

Semi-Supervised Learning: SSL

Can we use unlabeled data to augment a small labeled sample to improve learning?

But unlabeled data is missing the most important info!!

But maybe still has useful regularities that we can use.

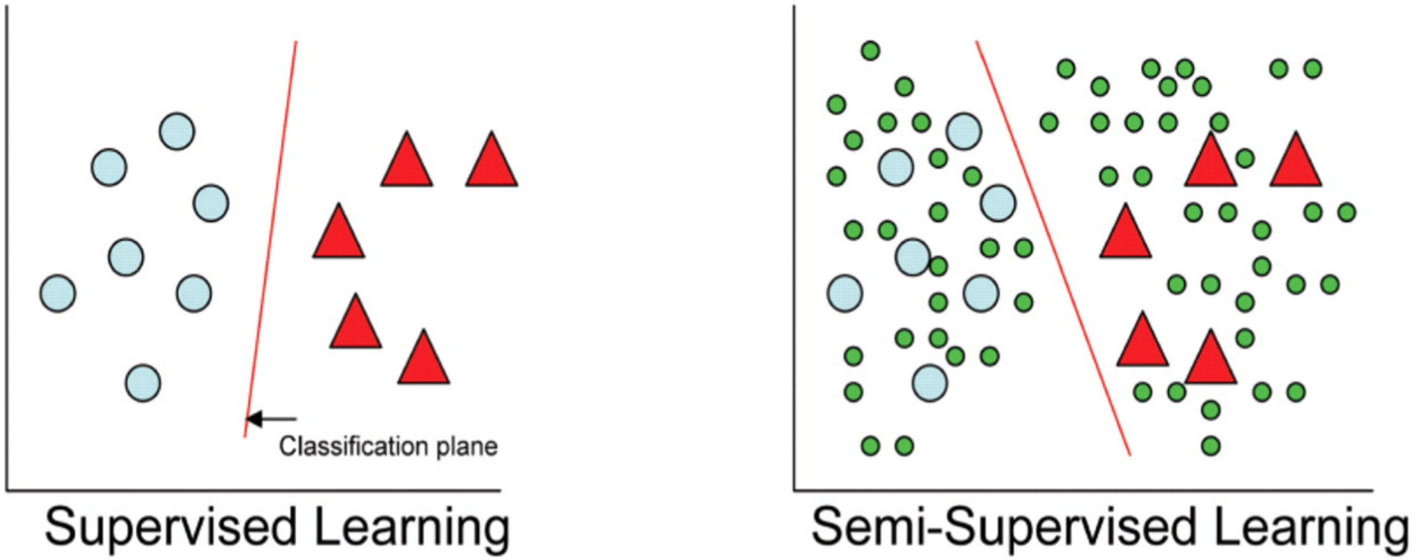
But

But...

But...

Semi-Supervised Learning: SSL

- **What is semi-supervised learning?**
  - Supervised learning + Additional unlabeled data
  - Unsupervised learning + Additional labeled data

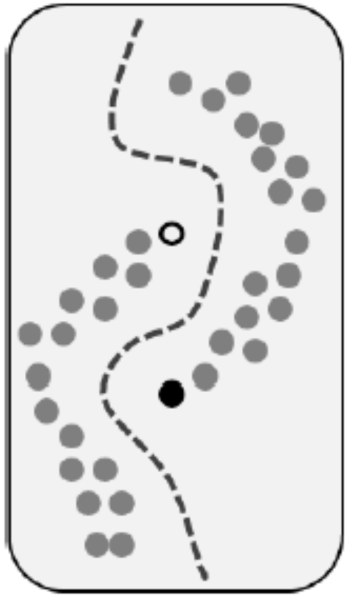


Semi-Supervised Learning: SSL

- Classification
  - **Transductive** – predict labels of unlabeled data
  - **Inductive** – learn a classification function
- Clustering (constrained clustering)
- Ranking (semi-supervised ranking)
- Almost every learning problem has a semi-supervised counterpart.

What is semi-supervised learning?

- Supervised learning + Additional unlabeled data
- Unsupervised learning + Additional labeled data



Why semi-supervised learning?

In many domains:

- Labeling could be expensive and difficult
- Unlabeled examples are easier to obtain
- Examples:
  - Web page classification
  - Speech recognition
  - Bioinformatics (e.g. protein sequences)
  - ...

Semi-Supervised Learning: SSL

- Forms:
  - **Classification,** clustering, regression, etc
- Two main settings for classification:
  - **Transductive**: Produce label only for the available unlabeled data.
    - The output of the method is not a classifier.
  - **Inductive**: Not only produce label for unlabeled data, but also produce a classifier.

Chapelle, Olivier, Schölkopf, Bernhard, and Zien, Alexander. Semi-Supervised Learning. The MIT Press, first edition, 2006.

Zhu, Xiaojin and Goldberg, Andrew B. Introduction to Semi-Supervised Learning. Morgan and Claypool, first edition, 2009.

Semi-Supervised Learning: SSL

- Familias de métodos en clasificación SS:
  - S3VMs
  - Métodos basados en Grafos y Label Propagation
  - Self Labeling
- Algoritmos de Self Labeling:
  - Self-Training
  - Co-Training
  - Tri-Training

Blum, Avrim and Mitchell, Tom. Combining labeled and unlabeled data with Co-Training. In Proceedings of the Annual ACM Conference on Computational Learning Theory, pp. 92–100, 1998.

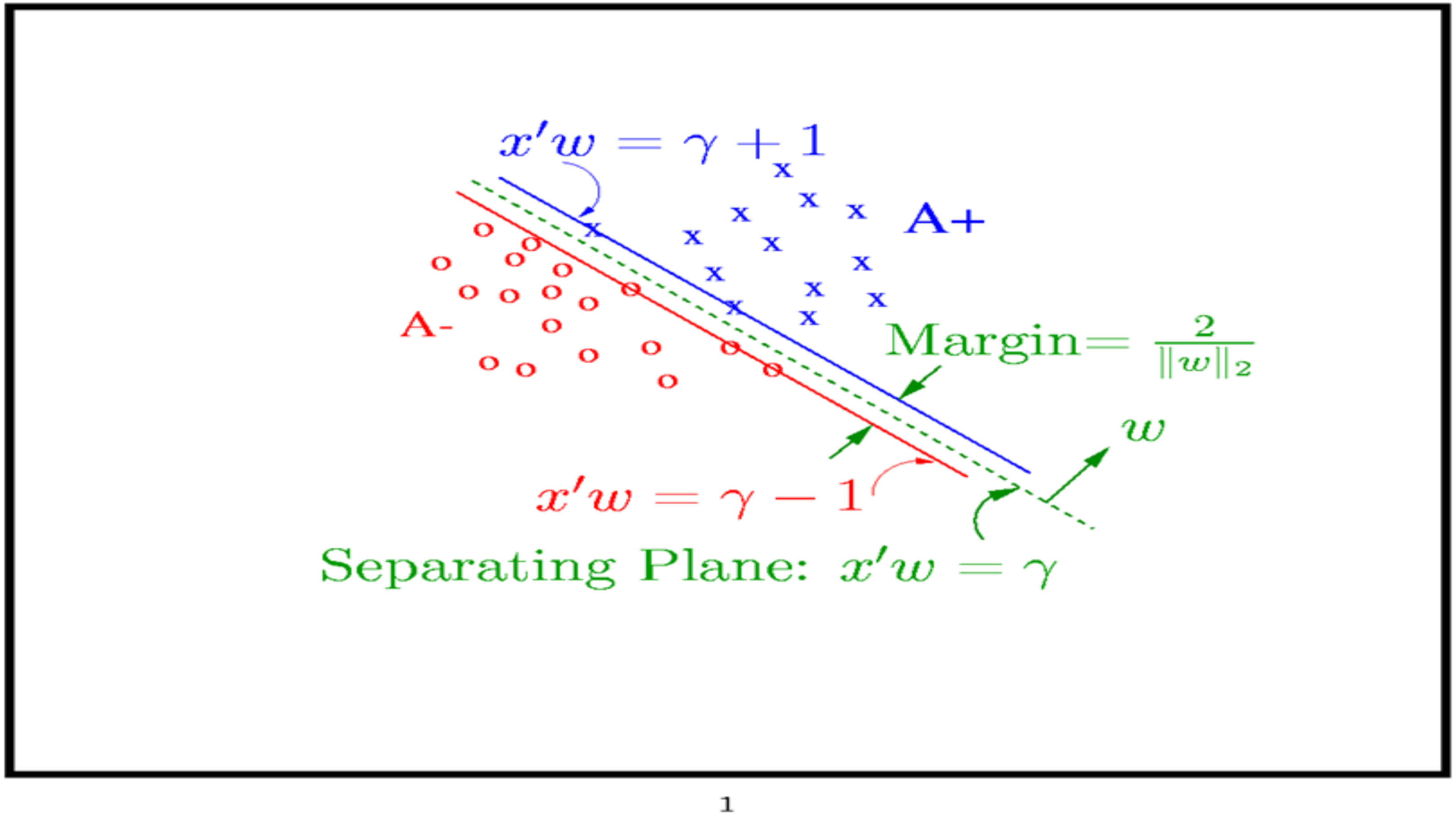
Zhou, Zhi-Hua and Li, Ming. Tri-training: Exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering, 17:1529–1541, 2005. ISSN 1041-4347.

Semi-Supervised Learning: SSL
Notación

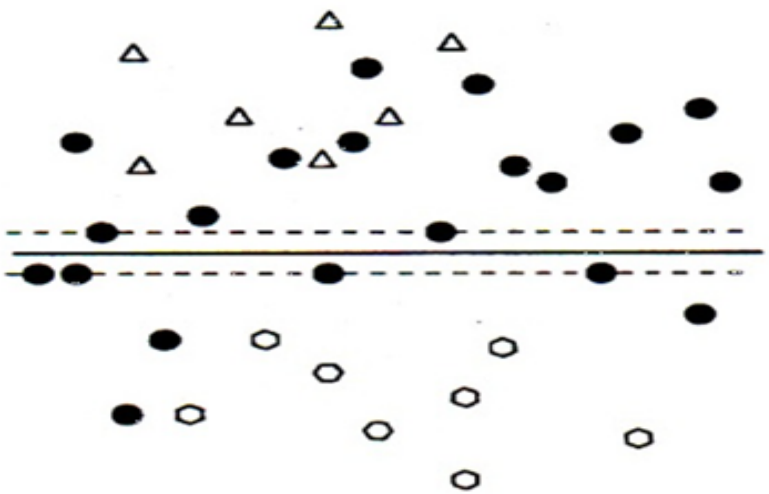
- $X_p = \{X_{p1}, X_{p2}, \ldots, X_{pD,}\ \omega\}$
- Conjunto ejemplos etiquetados *L*: *n* ejemplos X_p con ω conocida.
- Conjunto ejemplos no etiquetados *U*: *m* ejemplos X_q con ω desconocida, *m* >> *n*.
- $TR = L \cup U$
- Aprendizaje Transductivo:
  - Etiquetar *m* ejemplos X_q de *U*
- Aprendizaje Inductivo:
  - Clasificar los ejemplos de *TS* a partir de lo aprendido desde *TR*

Semi-Supervised Support Vector Machines: S3VMs

Linear SVM:



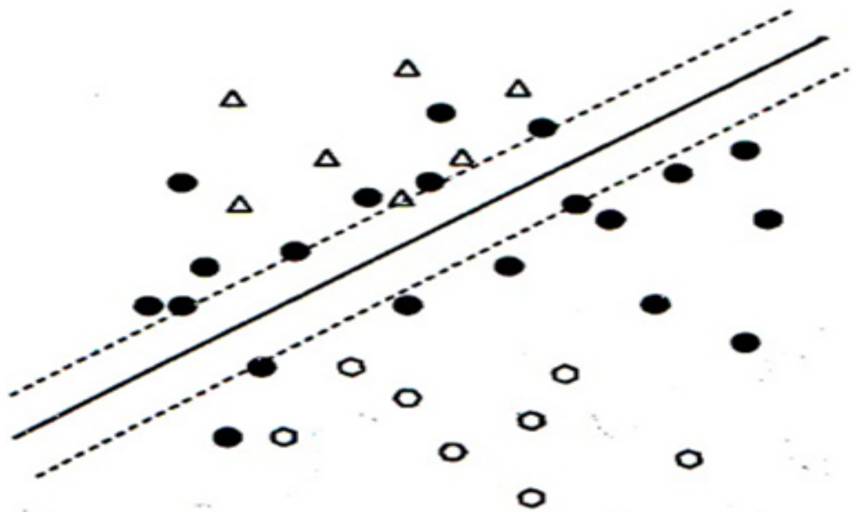
Semi-Supervised Support Vector Machines: S3VMs



SVM

Hollow shapes represent labeled data

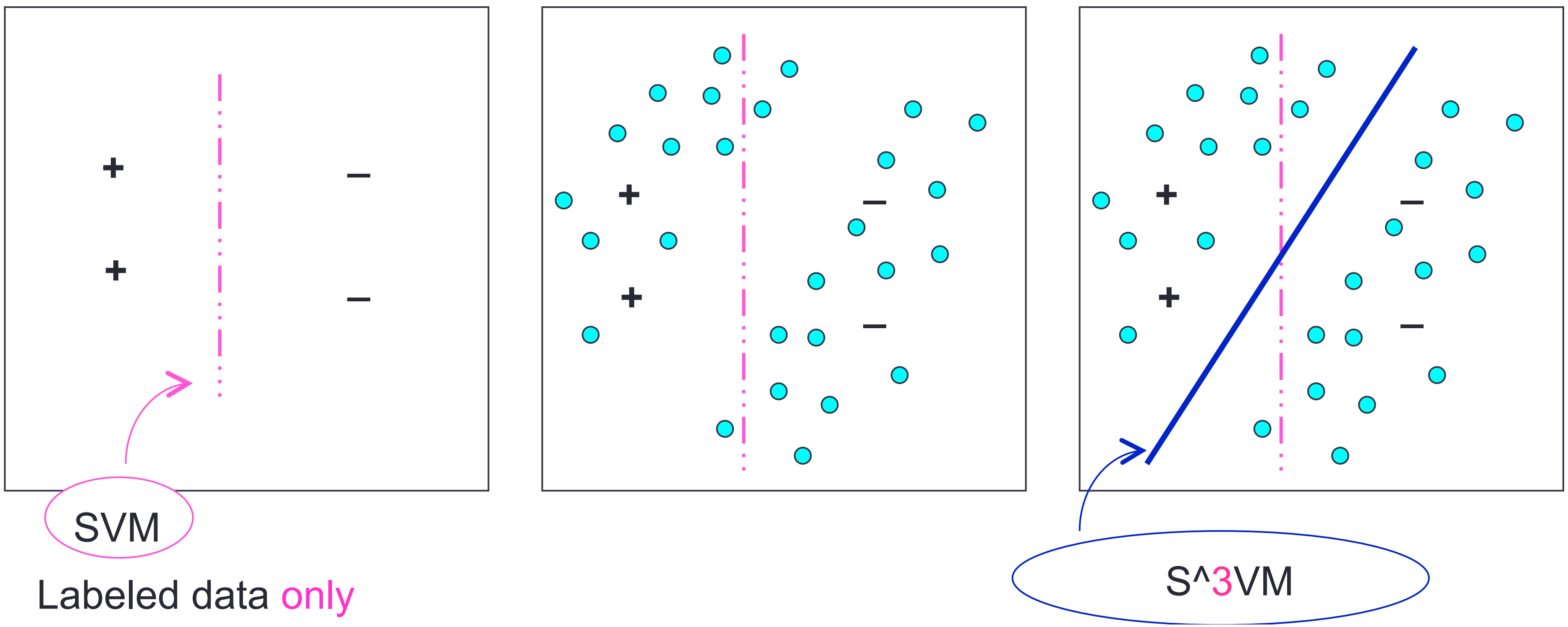
Solid shapes represent unlabeled data



S3VM

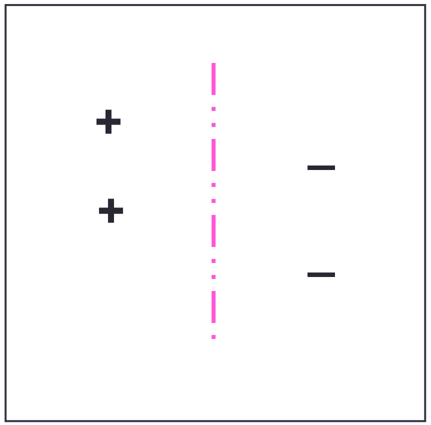
Semi-Supervised Support Vector Machines: S3VMs

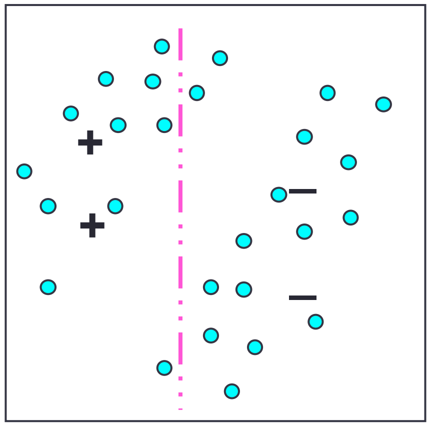
- Suppose we believe target separator goes through low density regions of the space/large margin.
- Aim for separator with large margin wrt labeled and unlabeled data. (L+U)

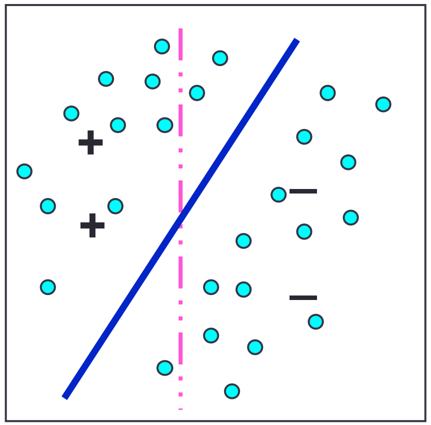


Semi-Supervised Support Vector Machines: S3VMs

- Unfortunately, optimization problem is now NP-hard. Algorithm instead does local optimization.
  - Start with large margin over labeled data. Induces labels on U.
  - Then try flipping labels in greedy fashion.
  - Or, branch-and-bound, other methods (Chapelle etal06)
- Quite successful on text data.

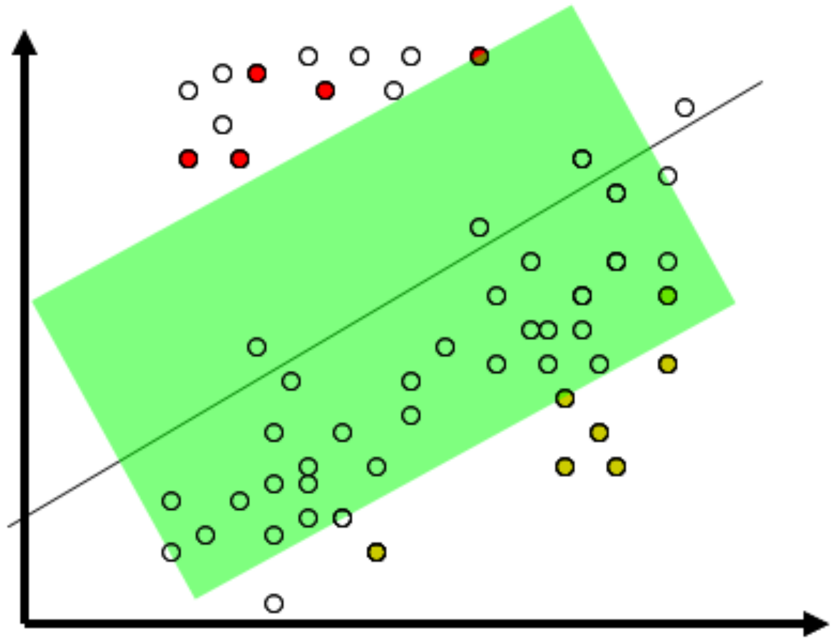






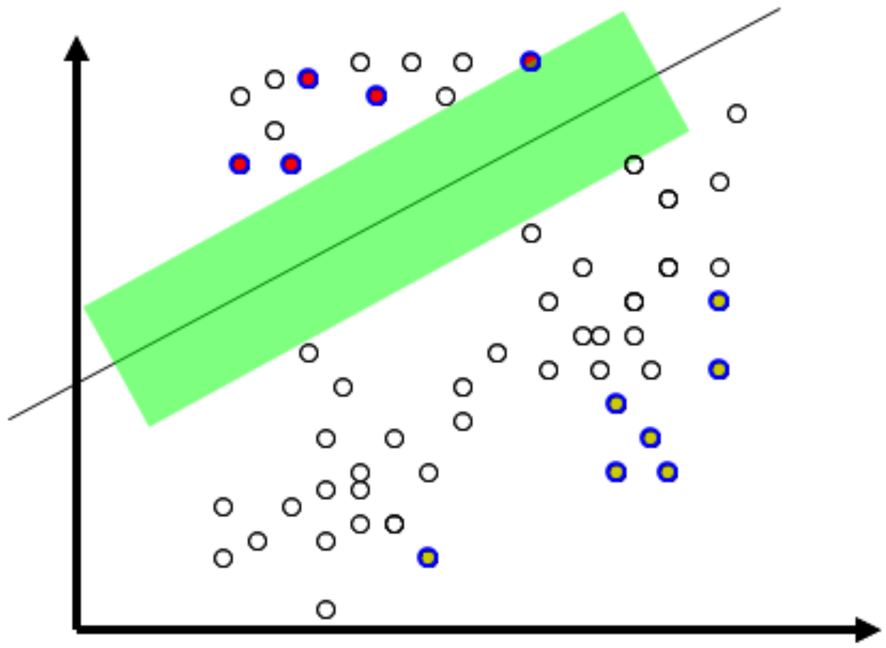
Transductive SVM

- Decision boundary given a small number of labeled examples
- How to change decision boundary given both labeled and unlabeled examples ?



Transductive SVM

- Decision boundary given a small number of labeled examples
- Move the decision boundary to low local density



Transductive SVM

Original SVM

$$\{\vec{w}^*, b^*\} = \underset{\vec{w}, b}{\operatorname{argmin}}\ \vec{w} \cdot \vec{w}$$

$$\left.\begin{array}{l} y_1(\vec{w} \cdot \vec{x}_1 + b) \geq 1 \\ y_2(\vec{w} \cdot \vec{x}_2 + b) \geq 1 \\ \ldots. \\ y_n(\vec{w} \cdot \vec{x}_n + b) \geq 1 \end{array}\right\} \text{labeled examples}$$

A binary variables for label of each example

Transductive SVM

$$\{\vec{w}^*, b^*\} = \boxed{\underset{y_{n+1}, \ldots, y_{n+m}}{\operatorname{argmin}}}\ \underset{\vec{w}, b}{\operatorname{argmin}}\ \vec{w} \cdot \vec{w}$$

$$\left.\begin{array}{l} y_1(\vec{w} \cdot \vec{x}_1 + b) \geq 1 \\ y_2(\vec{w} \cdot \vec{x}_2 + b) \geq 1 \\ \ldots. \\ y_n(\vec{w} \cdot \vec{x}_n + b) \geq 1 \end{array}\right\} \text{labeled examples}$$

$$\boxed{\left.\begin{array}{l} y_{n+1}(\vec{w} \cdot \vec{x}_{n+1} + b) \geq 1 \\ \ldots. \\ y_{n+m}(\vec{w} \cdot \vec{x}_{n+m} + b) \geq 1 \end{array}\right\} \text{unlabeled examples}}$$

Constraints for unlabeled data

Transductive SVM

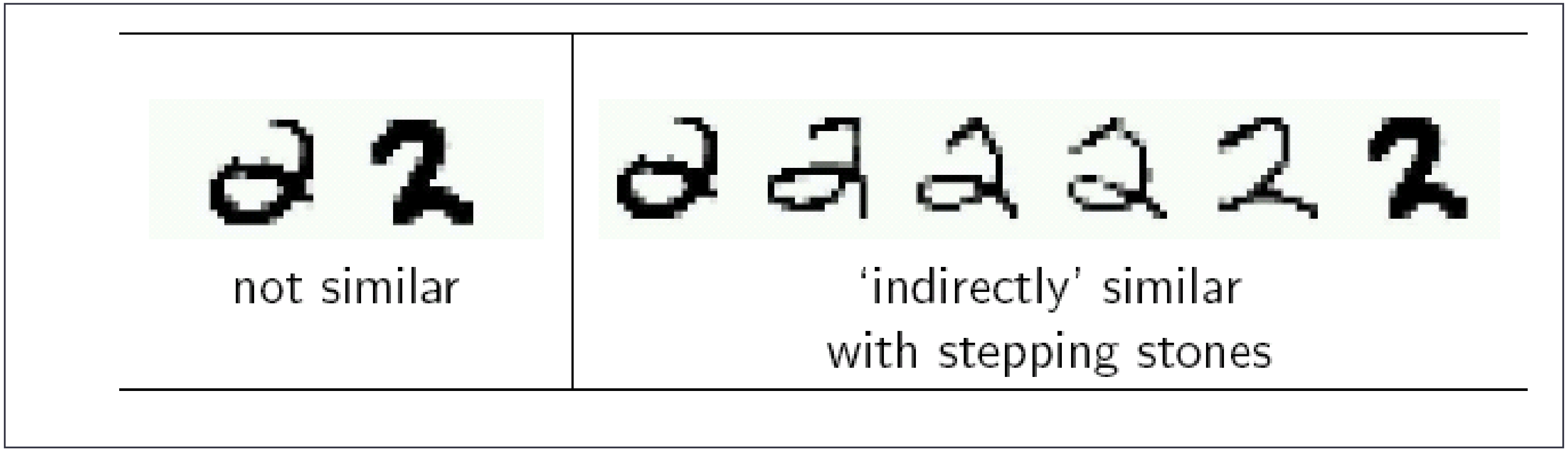
$$\{\vec{w}^*, b^*\} = \underset{y_{n+1},\dots,y_{n+m}}{\operatorname{argmin}} \; \underset{\vec{w},b}{\operatorname{argmin}} \; \vec{w}\cdot\vec{w} + \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \eta_i$$

$$\left.\begin{array}{l} y_1(\vec{w}\cdot\vec{x}_1 + b) \geq 1 - \xi_1 \\ y_2(\vec{w}\cdot\vec{x}_2 + b) \geq 1 - \xi_2 \\ \dots. \\ y_n(\vec{w}\cdot\vec{x}_n + b) \geq 1 - \xi_n \end{array}\right\} \text{labeled examples} \qquad \left.\begin{array}{l} y_{n+1}(\vec{w}\cdot\vec{x}_{n+1} + b) \geq 1 + \eta_1 \\ \dots. \\ y_{n+m}(\vec{w}\cdot\vec{x}_{n+m} + b) \geq 1 + \eta_m \end{array}\right\} \text{unlabeled examples}$$

- No longer convex optimization problem.
- Alternating optimization

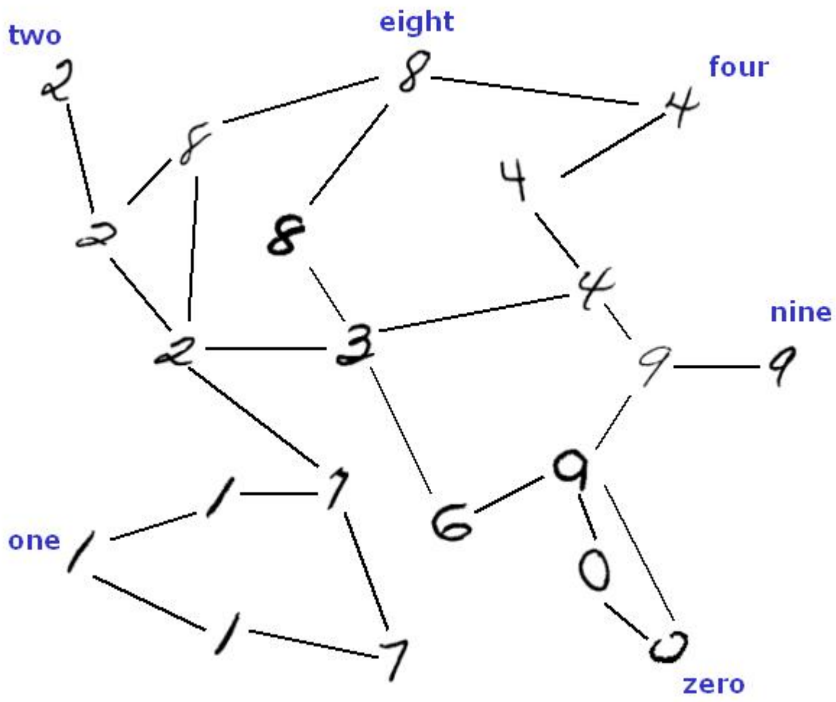
SSL: Graph-based Methods

- Suppose we believe that very similar examples probably have the same label.
- If you have a lot of labeled data, this suggests a Nearest-Neighbor type of alg.
- If you have a lot of unlabeled data, perhaps can use them as "stepping stones"



SSL: Graph-based Methods

- Idea: construct a graph with edges between very similar examples.
- Unlabeled data can help "glue" the objects of the same class together.



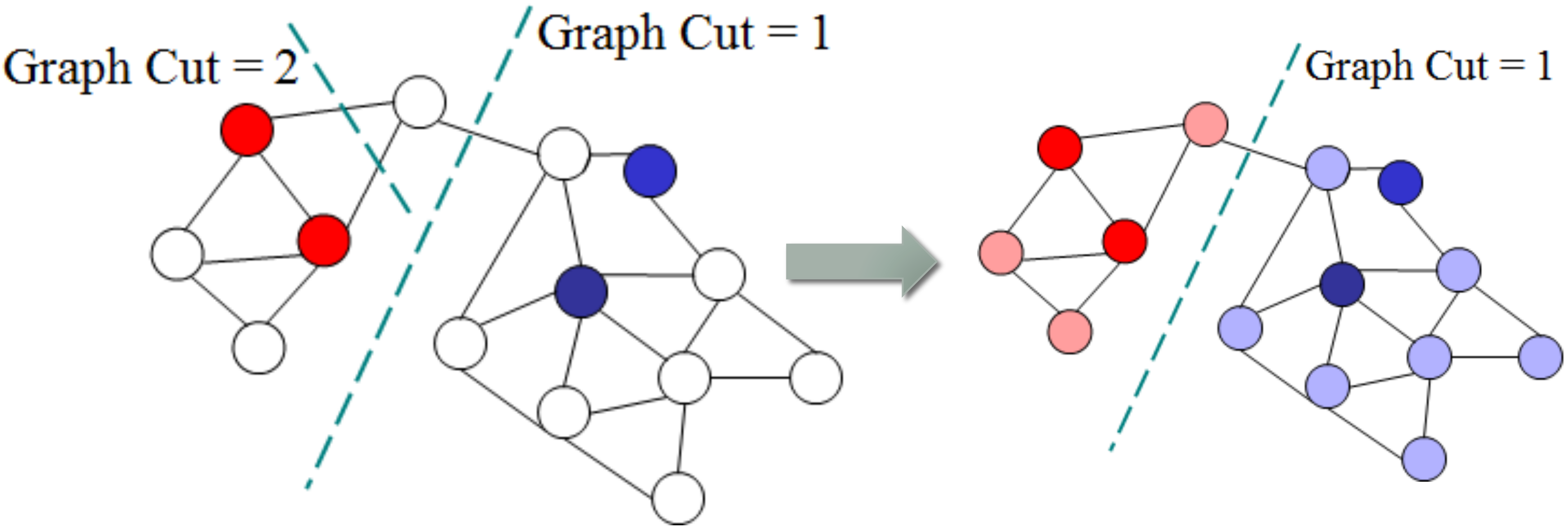
Solve for:

- Minimum cut
- Minimum "soft-cut"

  $$\sum_{e=(u,v)}(f(u)-f(v))^2$$
- Spectral partitioning
- …

SSL: Graph-based Methods

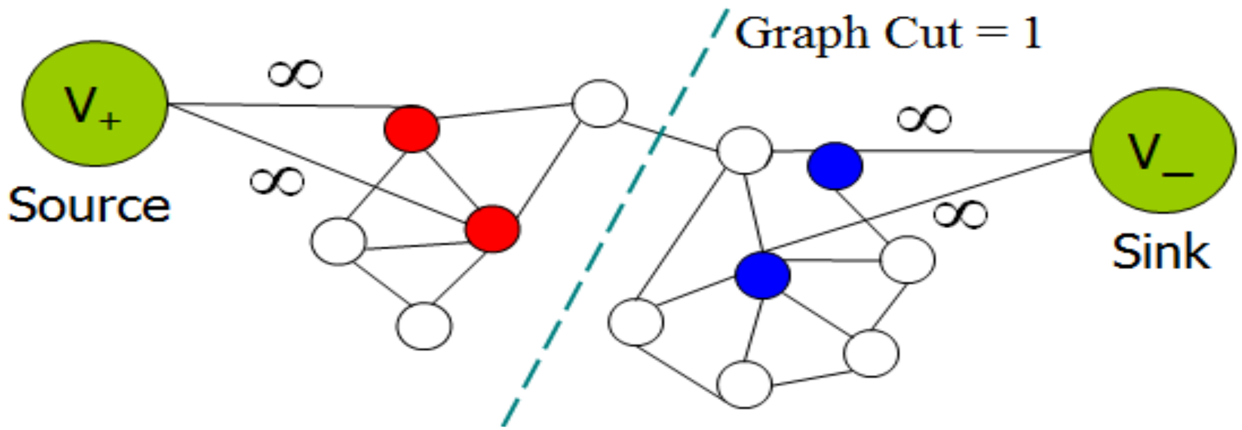
- Classification as graph partitioning
- Search for a classification boundary
  - Consistent with labeled examples
  - Partition with small graph cut



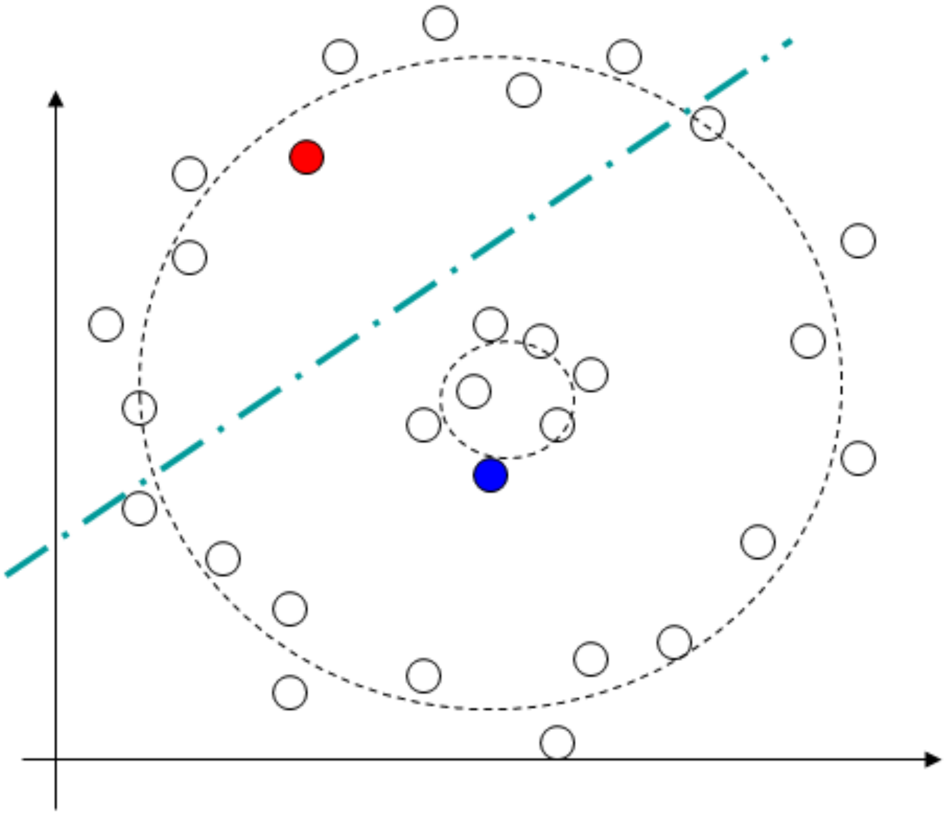
SSL: Graph-based Methods

Min-cuts [Blum and Chawla, ICML 2001]

- Additional nodes
  - V_+ : source, V-: sink
  - Infinite weights connecting sinks and sources
- High computational cost

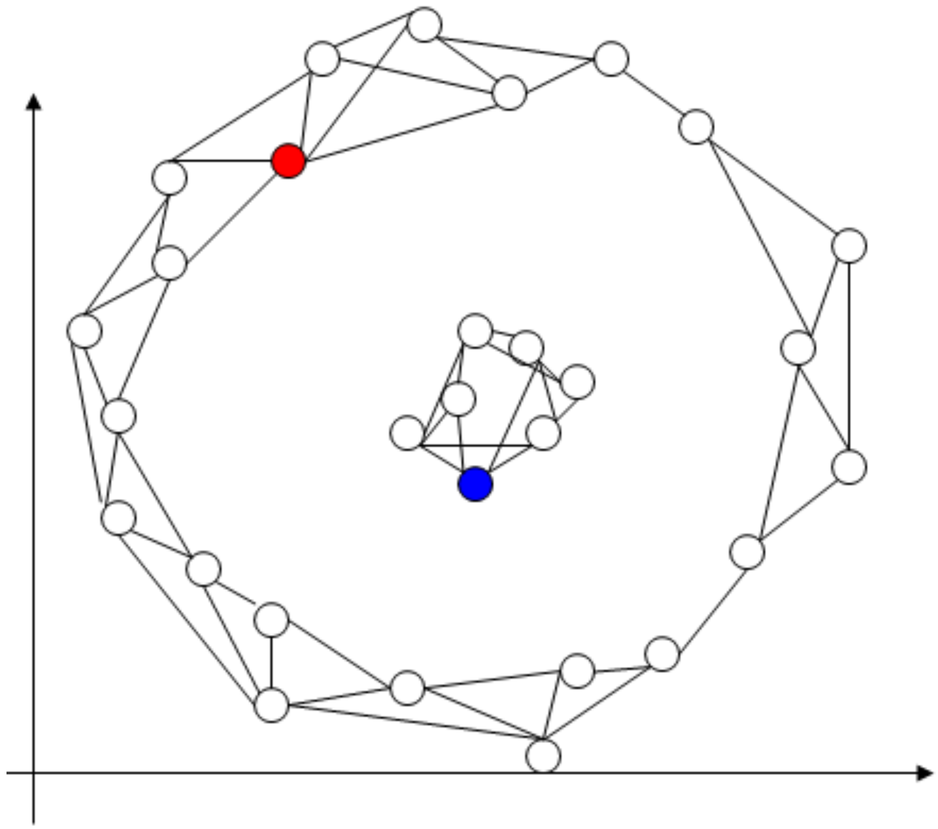


SSL: Label Propagation



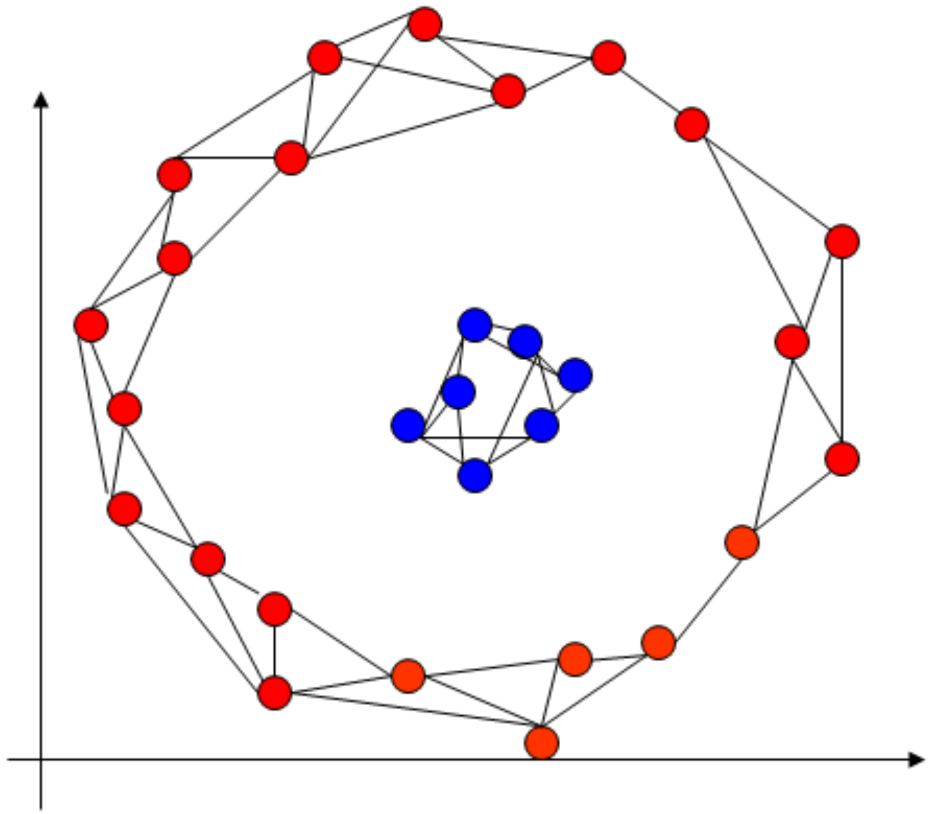
- A decision boundary based on the labeled examples is unable to take into account the layout of the data points
- How to incorporate the data distribution into the prediction of class labels?

SSL: Label Propagation



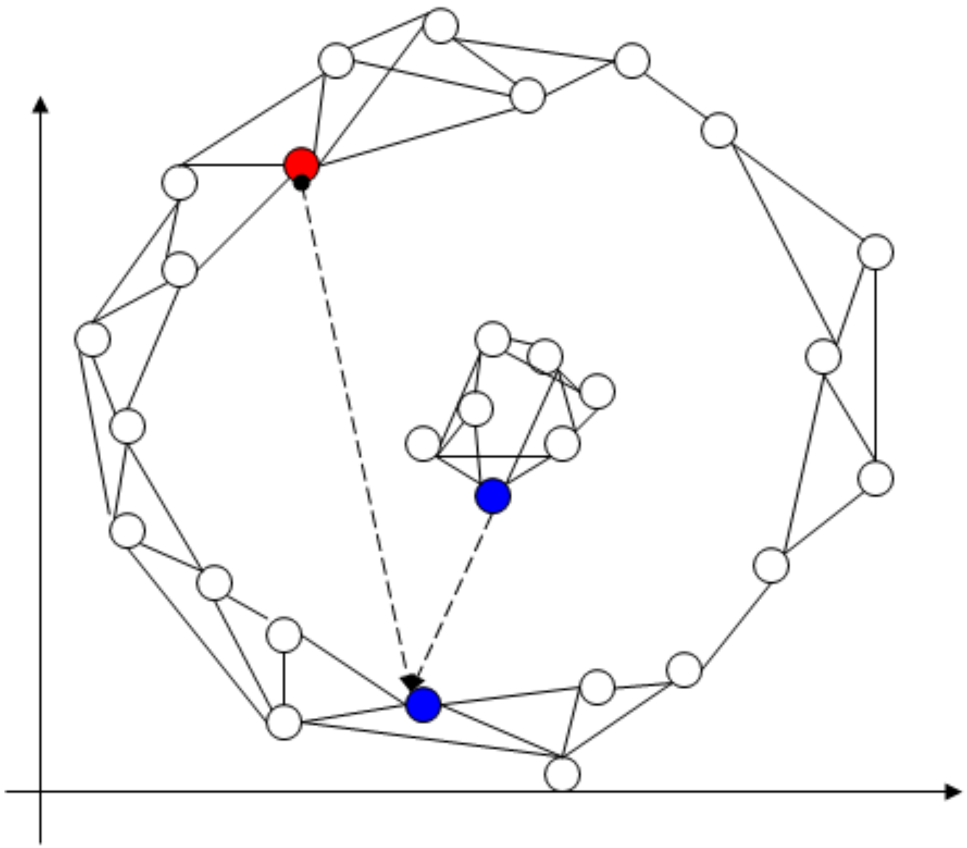
- Connect the data points that are close to each other

SSL: Label Propagation



- Connect the data points that are close to each other
- Propagate the class labels over the connected graph

SSL: Label Propagation



- Connect the data points that are close to each other
- Propagate the class labels over the connected graph
- Different from the K Nearest Neighbor

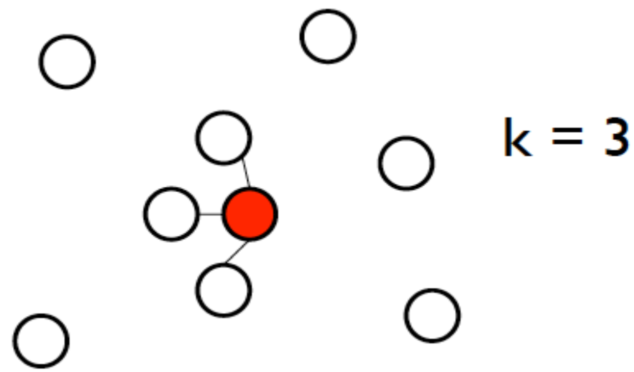
SSL: Label Propagation

Graph Construction

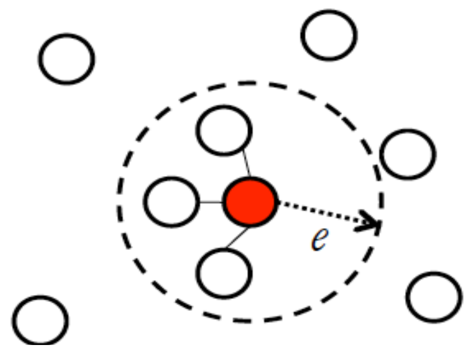
- Neighborhood Methods
  - k-NN Graph Construction (k-NNG)
  - e-Neighborhood Method
- Metric Learning
- Other approaches

SSL: Label Propagation

- k-Nearest Neighbor Graph (k-NNG)
  - add edges between an instance and its k-nearest neighbors



- e-Neighborhood
  - add edges to all instances inside a ball of radius e

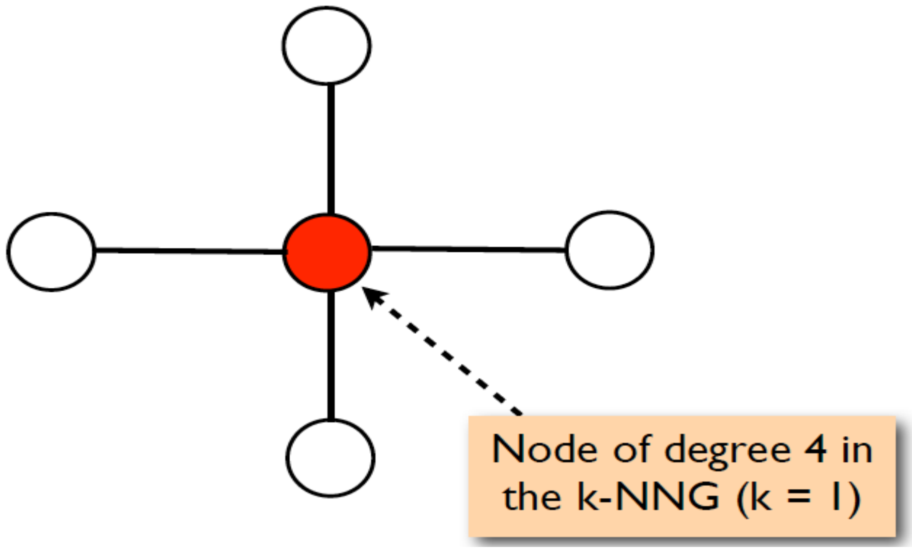


SSL: Label Propagation

Issues with k-NNG

- Not scalable (quadratic)
- Results in an asymmetric graph
  - b is the closest neighbor of a, but not the other way

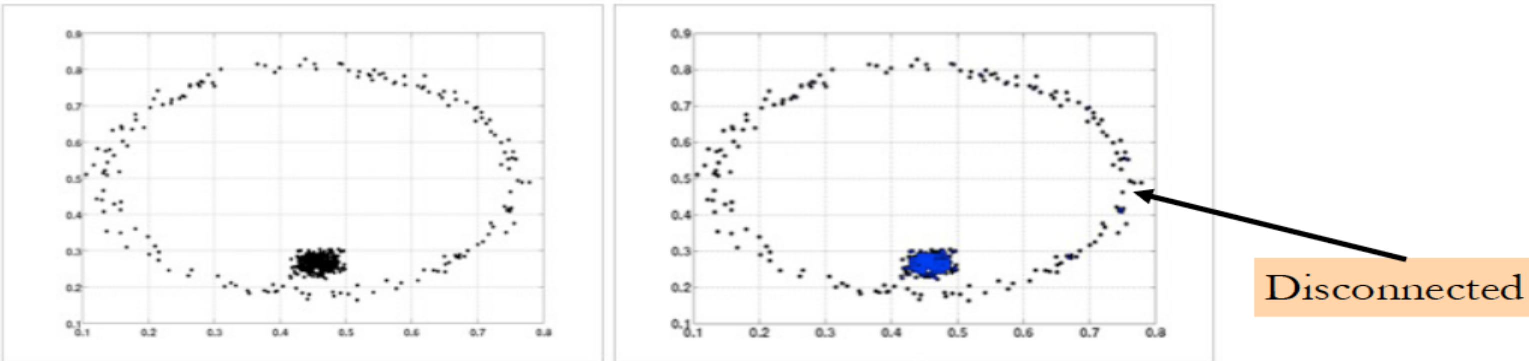
- Results in irregular graphs
  - some nodes may end up with higher degree than other nodes



SSL: Label Propagation

Issues with e-Neighborhood

- Not scalable
- Sensitive to value of e : not invariant to scaling
- Fragmented Graph: disconnected components



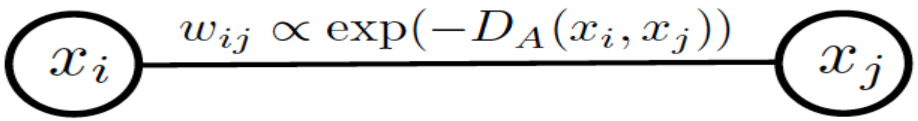
Data

e-Neighborhood

Figure from [Jebara et al., ICML 2009]

SSL: Label Propagation

Graph Construction using Metric Learning



$$D_A(x_i, x_j) = (x_i - x_j)^T A (x_i - x_j)$$

Estimated using Mahalanobis metric learning algorithms

- Supervised Metric Learning
  - ITML [Kulis et al., ICML 2007]
  - LMNN [Weinberger and Saul, JMLR 2009]
- Semi-supervised Metric Learning
  - IDML [Dhillon et al., UPenn TR 2010]

The Self-Training Algorithm

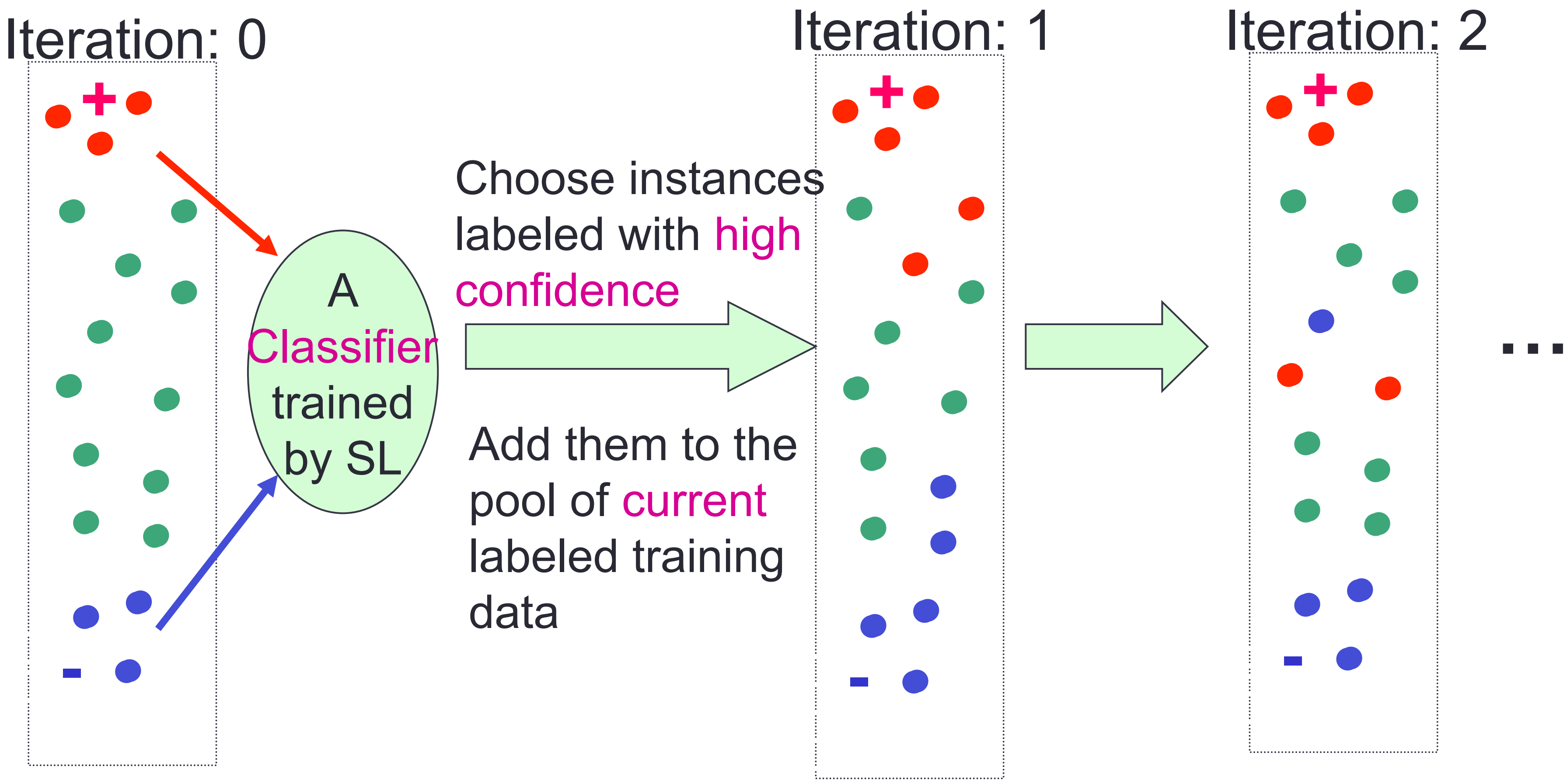
(Yarowsky 1995)

- Suposición:
  - Las propias prediciones con alta confianza son correctas.
- Algoritmo Self-training:
  1. Entrena f desde $\{(x_{1:n}, y_{1:n})\}$
  2. Predice sobre U
  3. Añade (x, f(x)) a los datos etiquetados
     - A. Añade todo
     - B. **Añade los más confiables**
     - C. Añade pesos
  4. Repite

- Ventajas:
  - Simple
  - Wrapper
- Inconvenientes:
  1. Errores tempranos
  2. No se puede conocer su convergencia

The Self-Training Algorithm

(Yarowsky 1995)



The Co-Training Algorithm

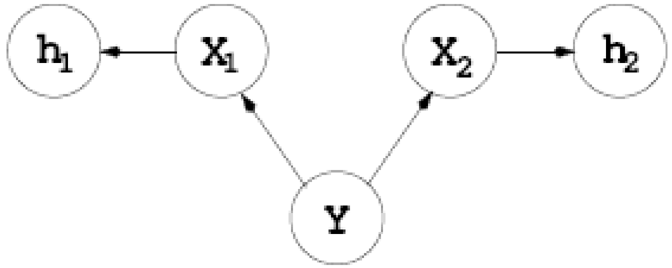


Figure 1: The co-training scenario with rules h_1 and h_2.

Two views – X1, X2

Two distinct hypothesis classes H1, H2 consisting of functions predicting Y from X1 and X2 respectively

Bootstrap using h1єH1, h2єH2

"If X1 is conditionally independent of X2 given Y then given a weak predictor in H1 and given an algorithm which can learn H2 under random misclassification noise, then it is possible to learn a good predictor in H2"

The Co-Training Algorithm
(Blum and Mitchell 1998)

- Instances contain two **sufficient sets of features**
  - i.e. an instance is $x=(x_1,x_2)$
  - Each set of features is called a **View**

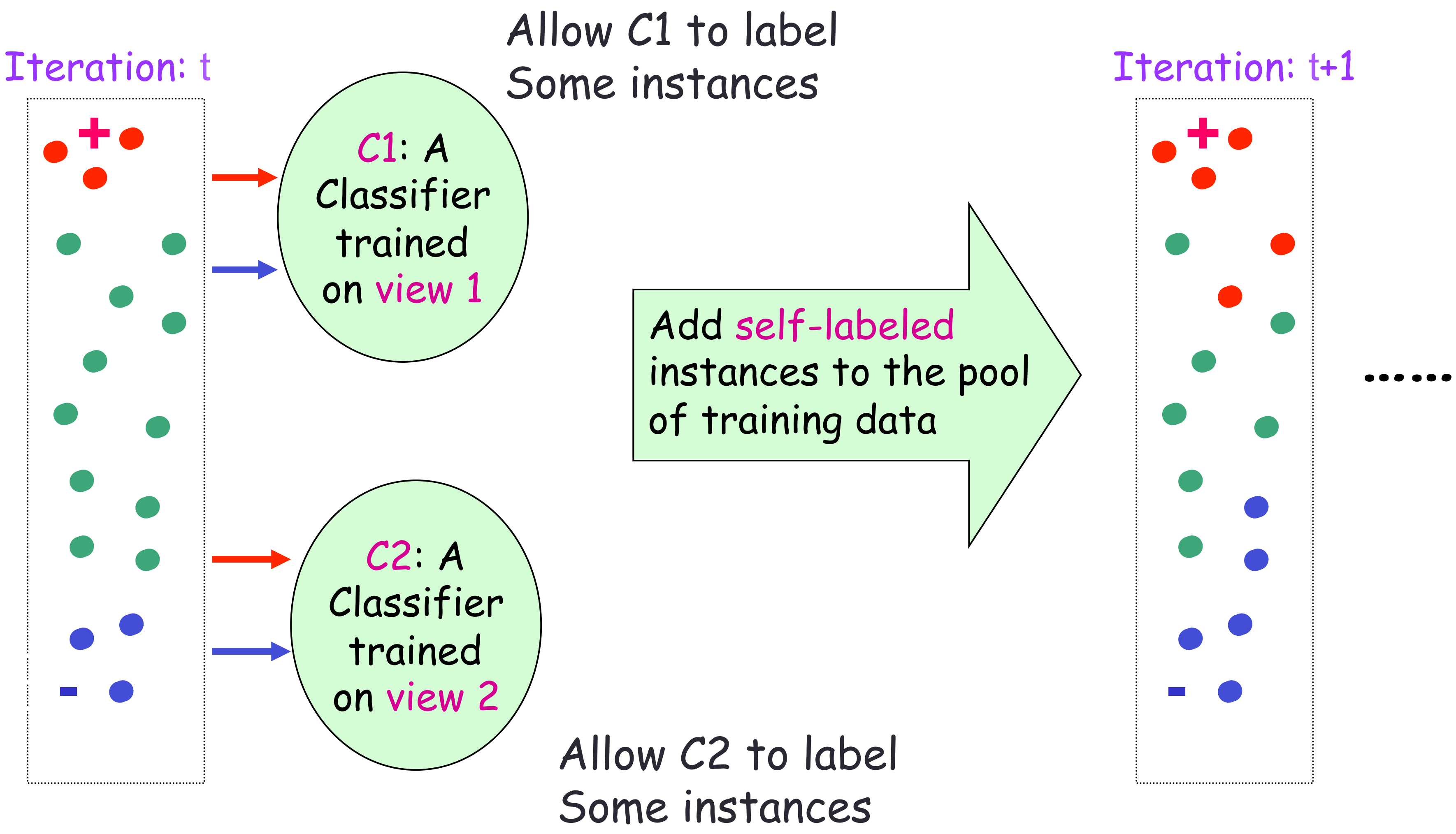
- Two views are **independent given the label**:

$$P(x_1|x_2, y) = P(x_1|y)$$
$$P(x_2|x_1, y) = P(x_2|y)$$

- Two views are **consistent:**

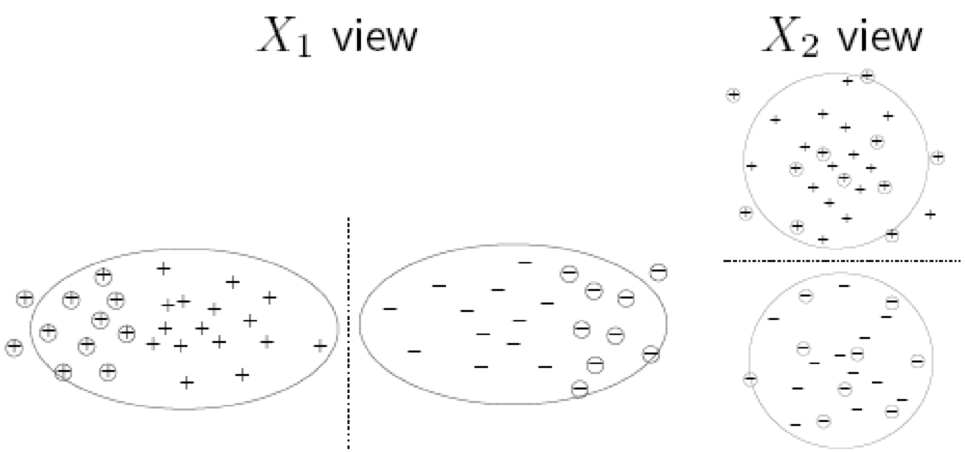
$$\exists c_1, c_2 : c^{opt}(x) = c_1(x_1) = c_2(x_2)$$

The Co-Training Algorithm



The Co-Training Algorithm

- Suposición:
  - División atributos $x=[x^{(1)}; x^{(2)}]$ existe
  - $x^{(1)}$ ó $x^{(2)}$ solo es sufficiente para entrenar un clasificador
  - $x^{(1)}$ ó $x^{(2)}$ son condicionalmente independientes dada la clase



- Algoritmo Co-Training:

1. Entrena 2 clasificadores:
   $f^{(1)}$ desde $(X_l^{(1)}, Y_l)$
   $f^{(2)}$ desde $(X_l^{(1)}, Y_l)$
2. Clasifica X_u con $f^{(1)}$ y $f^{(2)}$ separadamente.
3. Añade $f^{(1)}$'s k-más-confidentes $(x, f^{(1)}(x))$ a $f^{(2)}$'s datos etiquetados.
4. Añade $f^{(2)}$'s k-más-confidentes $(x, f^{(2)}(x))$ a $f^{(1)}$'s datos etiquetados.
5. Repite.

The Tri-Training Algorithm

- Se propuso para mejorar Co-Training, eliminando la posibilidad de dividir los atributos en dos subconjuntos.
- Utiliza tres clasificadores, h_1, h_2, h_3 ; del mismo algoritmo de aprendizaje supervisado.
- La diversidad entre clasificadores solo se puede conseguir manipulando *L (conjunto etiquetado)*.

The Tri-Training Algorithm

- Algoritmo
  - Muestreo con reemplazo de *L*.
  - Repite
    - Determinar los ejemplos no etiquetados para los que se obtiene mayor confianza.
    - Cada ejemplo de *U* tiene la oportunidad de ser etiquetado por h_1.
    - Un ejemplo no etiquetado X_q puede ser etiquetado para h_i, siempre que haya consenso por parte de los otros dos clasificadores. X_q se usará para modificar el modelo aprendido por h_i.
    - Se puede dar el caso que dos clasificadores predigan una clase incorrecta para el ejemplo no etiquetado, añadiendo ruido en la hipótesis aprendida por el otro clasificador. Para resolverlo, el modelo añade un mecanismo para compensar la influencia negativa de los ejemplos mal etiquetados.
  - Hasta que el modelo aprendido por los tres clasificadores no varía.
- La clasificación final se realiza mediate voto por mayoría de los clasificadores entrenados.

The Tri-Training Algorithm: TriSM

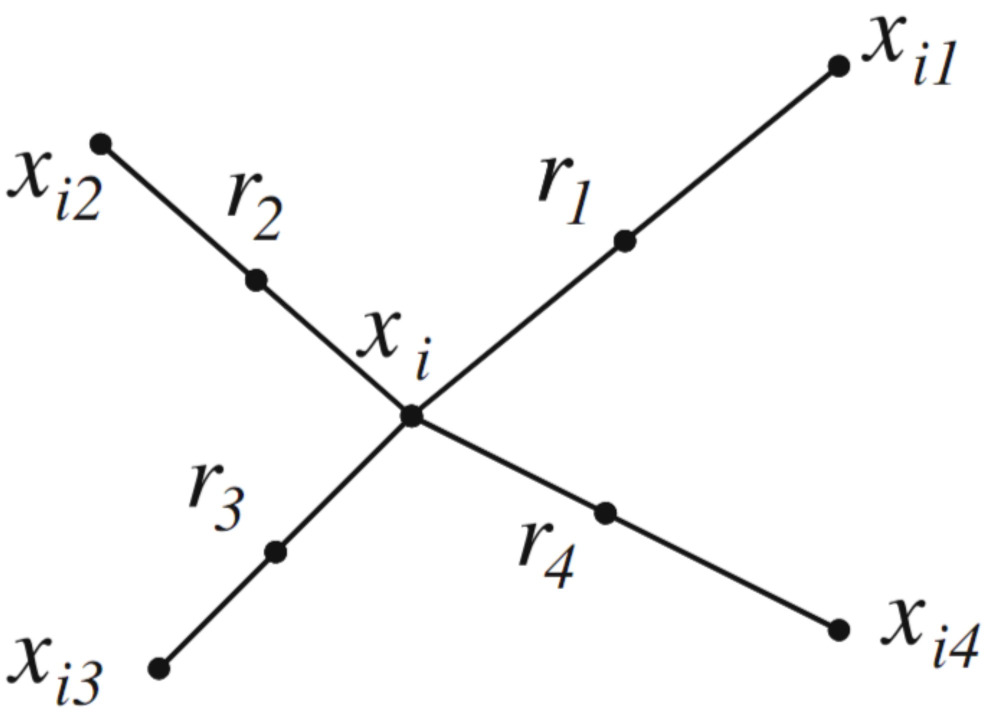
- Tri-Training presenta problemas en la búsqueda de diversidad, limitadas a los datos de entrada. El modelo de muestreo con reemplazo es insuficiente.

- Debido a que L << U, las diferencias de los muestreos no son muy significativas y se producen outliers de forma frecuente.

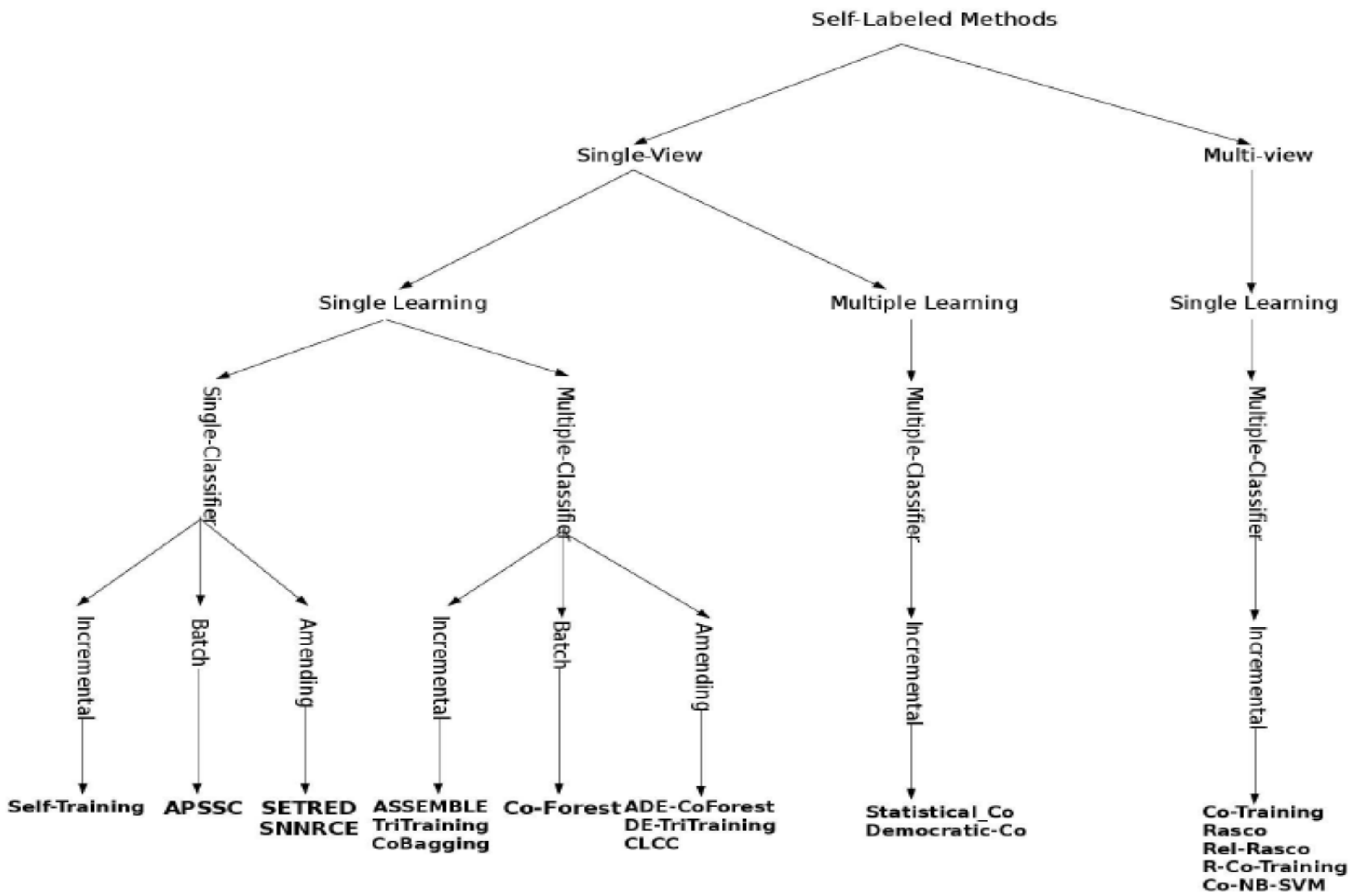
- Para resolver este problema, se introduce **TriSM**: Tri-training con sobremuestreo, introduciendo ejemplos sintéticos etiquetados.

The Tri-Training Algorithm: TriSM

- Con *L* completo, sobremuestreamos con una técnica basada en SMOTE, pero aplicada a todas las clases.
- Cada clasificador usará los datos etiquetados originales junto a algunos datos sintéticos.
- El ratio de sobremuestreo es constante para todas las clases.



Taxonomy of Self-Labeling Techniques



Experiments on Self-Labeling

- 55 data sets from the UCI repository.
- Base classifiers: *k*-NN, C4.5, Naive Bayes, SMO.
- Performance measures: Accuracy, Kappa, and runtime.
- 10 fold-cross validation
- We conduct different analyses:
  - Transductive and inductive capabilities.
  - Different labeled ratios: 10%, 20%, 30% and 40%
  - Number of classes: binary and multi-class problems.
  - 9 high dimensional data sets with small labeled ratio.
  - How far is SSL from supervised learning?

Associated web-page:

`http://sci2s.ugr.es/SelfLabeled`

Experiments on Self-Labeling

10%			20%			30%			40%		
	Acc	△K		Acc	△K		Acc	△K		Acc	△K
Democratic-Co	0.7304	-2	***TriTraining (C4.5)***	0.7609	-1	***Democratic-Co***	0.7744	-1	***Democratic-Co***	0.7834	-1
TriTraining (C4.5)	0.7297	+1	***Co-Bagging (C45)***	0.7597	+1	***TriTraining (C4.5)***	0.7708	-2	**TriTraining (C4.5)**	0.7812	-3
Co-Bagging (C4.5)	0.7278	+1	**Democratic-Co**	0.7545	-3	***Co-Bagging (C4.5)***	0.7707	0	*Co-Training (SMO)*	0.7781	+2
Self-Training (C4.5)	0.7214	0	Self-Training (C4.5)	0.7526	-1	Self-Training (C4.5)	0.7652	-2	Self-Training (C4.5)	0.7754	-3
SETRED	0.7178	0	Co-Training (C4.5)	0.7524	-2	*Co-Training (SMO)*	0.7642	+4	Co-Training (C4.5)	0.7754	-3
TriTraining (KNN)	0.7176	-11	*Co-Training (SMO)*	0.7447	+3	Co-Training (C4.5)	0.7639	-3	*Co-Bagging (SMO)*	0.7738	+3
SNNRCE	0.7151	-2	Co-Bagging (KNN)	0.7431	-8	Co-Bagging (SMO)	0.7568	-1	*TriTraining (SMO)*	0.7707	+3
CoForest	0.7124	+1	*TriTraining (SMO)*	0.7414	+4	Self-Training (SMO)	0.7553	+1	Self-Training (SMO)	0.7700	+2
Self-Training (KNN)	0.7101	-3	SETRED	0.7406	-2	SETRED	0.7550	-2	Co-Bagging (C4.5)	0.7638	0
Co-Bagging (KNN)	0.7097	-3	DE-TriTraining (SMO)	0.7394	0	TriTraining (SMO)	0.7545	+5	SETRED	0.7628	-1
Co-Training (C4.5)	0.7088	-3	DE-TriTraining (C4.5)	0.7387	-6	Co-Bagging (KNN)	0.7536	-3	Co-Bagging (KNN)	0.7628	-7
Co-Training (SMO)	0.7069	+6	Co-Bagging (SMO)	0.7386	+4	Self-Training (KNN)	0.7530	0	DE-TriTraining (C4.5)	0.7605	-5
DE-TriTraining (C4.5)	0.7058	-5	CoForest	0.7372	-1	DE-TriTraining (SMO)	0.7527	+3	DE-TriTraining (SMO)	0.7593	+3
DE-TriTraining (SMO)	0.7019	-1	Self-Training (KNN)	0.7367	+1	DE-TriTraining (C4.5)	0.7500	-1	Rasco (C4.5)	0.7591	+2
DE-TriTraining (KNN)	0.7016	-7	SNNRCE	0.7350	-4	CoForest	0.7497	+2	Self-Training (KNN)	0.7586	0
TriTraining (SMO)	0.6988	+6	TriTraining (KNN)	0.7303	-6	SNNRCE	0.7474	-4	Rel-Rasco (C4.5)	0.7578	+2
Self-Training (SMO)	0.6969	+9	Co-Training (KNN)	0.7302	-3	Co-Training (KNN)	0.7454	-2	CoForest	0.7565	-2
Co-Training (KNN)	0.6958	-5	Self-Training (SMO)	0.7298	+9	DE-TriTraining (KNN)	0.7426	-8	SNNRCE	0.7545	-3
Co-Training (NB)	0.6949	0	DE-TriTraining (KNN)	0.7294	-4	TriTraining (KNN)	0.7412	-8	Co-Training (KNN)	0.7539	-1
...	...	...	...	...	...	...	...	...	...	...	...

Experiments on Self-Labeling

Statistical comparison: Friedman test + Bergmann-Hommel to find out distinctive algorithms in $n * n$ comparisons.

Figure: Transductive

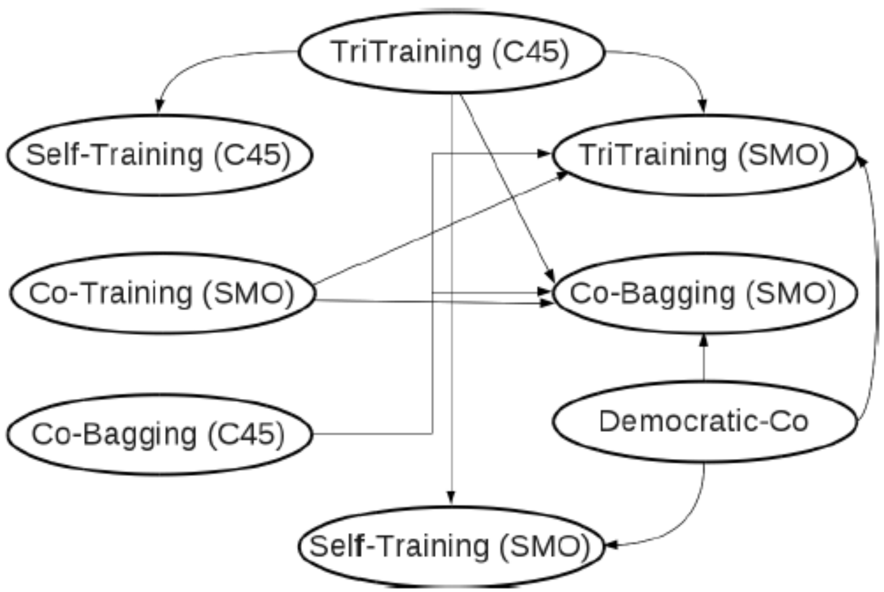
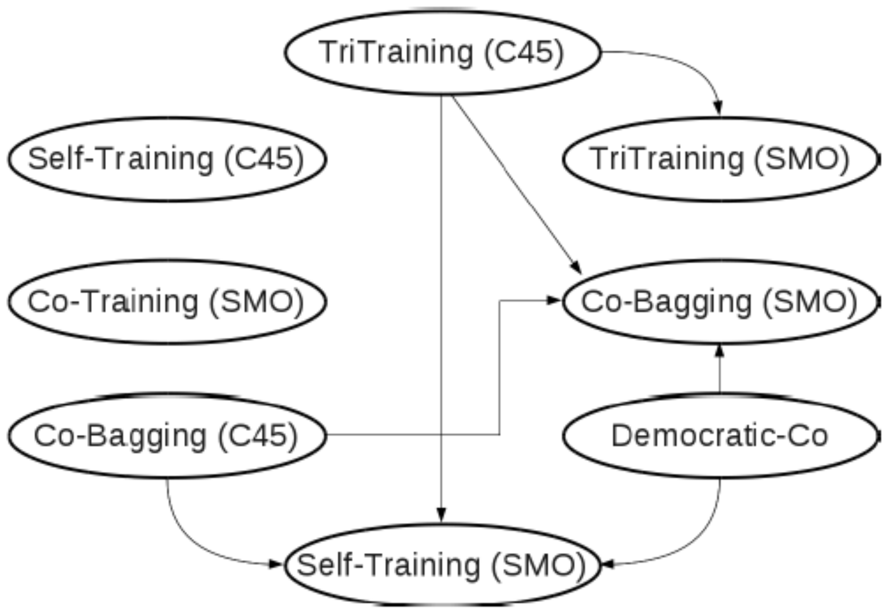


Figure: Inductive



Experiments on Self-Labeling
Lessons Learned

- Multiple-classifier+single view approaches have shown the best behavior.
- Experiments on high dimensional data with very reduced labeled data denote that much more work is required.
- SSL techniques are quite far from supervised learning, especially with a reduced labeled ratio (10%).
- A SSL module has been developed for the KEEL platform.

Self-Labeling: Our ideas

Two different works:

- **On the Characterization of Noise Filters for Self-Training Semi-Supervised Learning**.
- A Framework based on Synthetic Examples Generation for Self-Labeled Semi-Supervised Classification.

Associated web-page:

`http://sci2s.ugr.es/SelfTraining+Filters/`

Self-Labeling: Our ideas

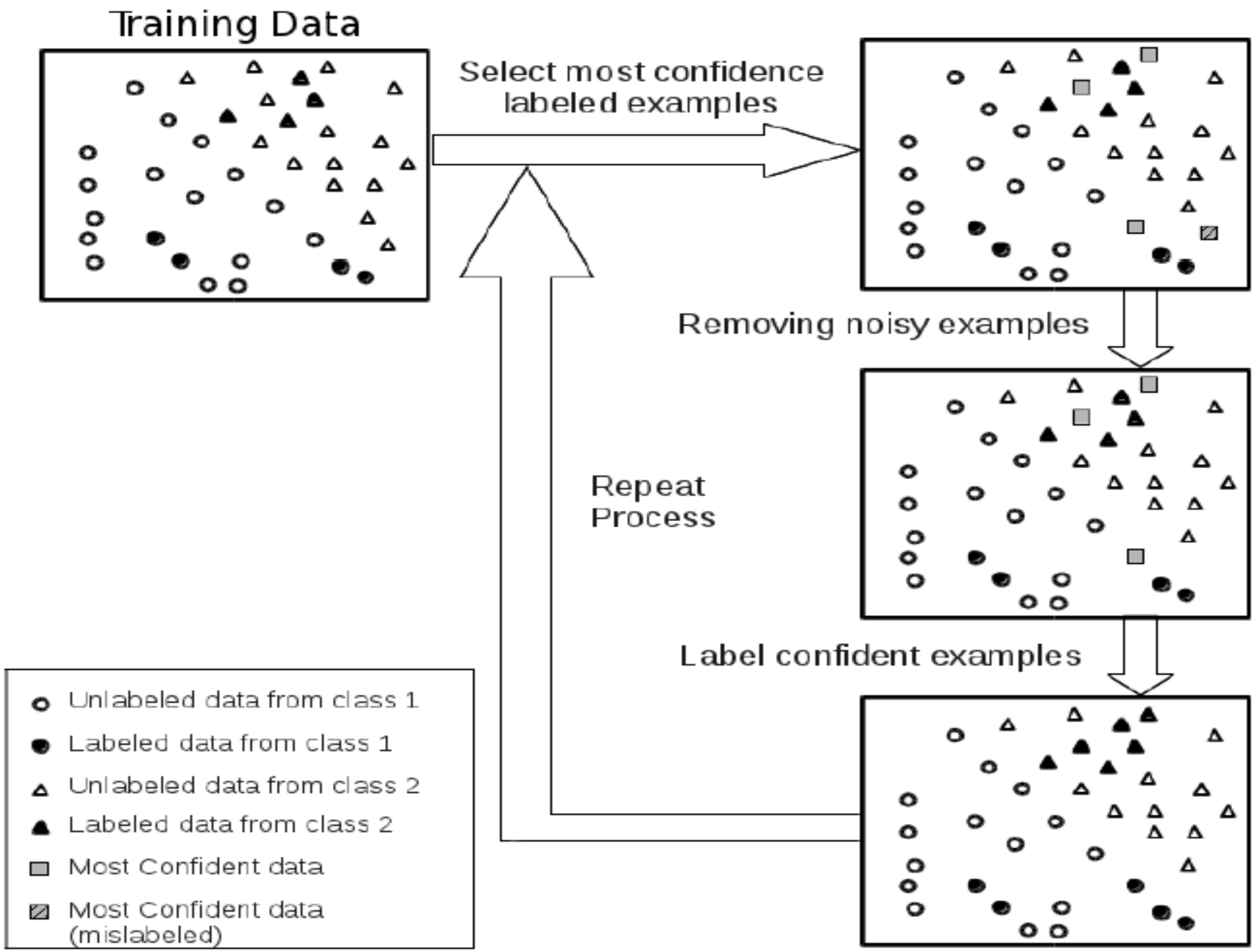
Motivation:

- Self-Training approach exemplifies the behavior of self-labeling techniques.
- Two types of noisy instances may appear:
  - Labeled data distribution may lead to wrong classifications.
  - There may be outliers within the original unlabeled data.
- Prototype selection models may detect noisy data.

Objectives:

- To remove noisy instance during the self-training process.
- To characterize which noise filters are more appropriate.

Self-Labeling: Our ideas



Self-Labeling: Our ideas

Two different works:

- On the Characterization of Noise Filters for Self-Training Semi-Supervised Learning.
- **A Framework based on Synthetic Examples Generation for Self-Labeled Semi-Supervised Classification.**

Self-Labeling: Our ideas

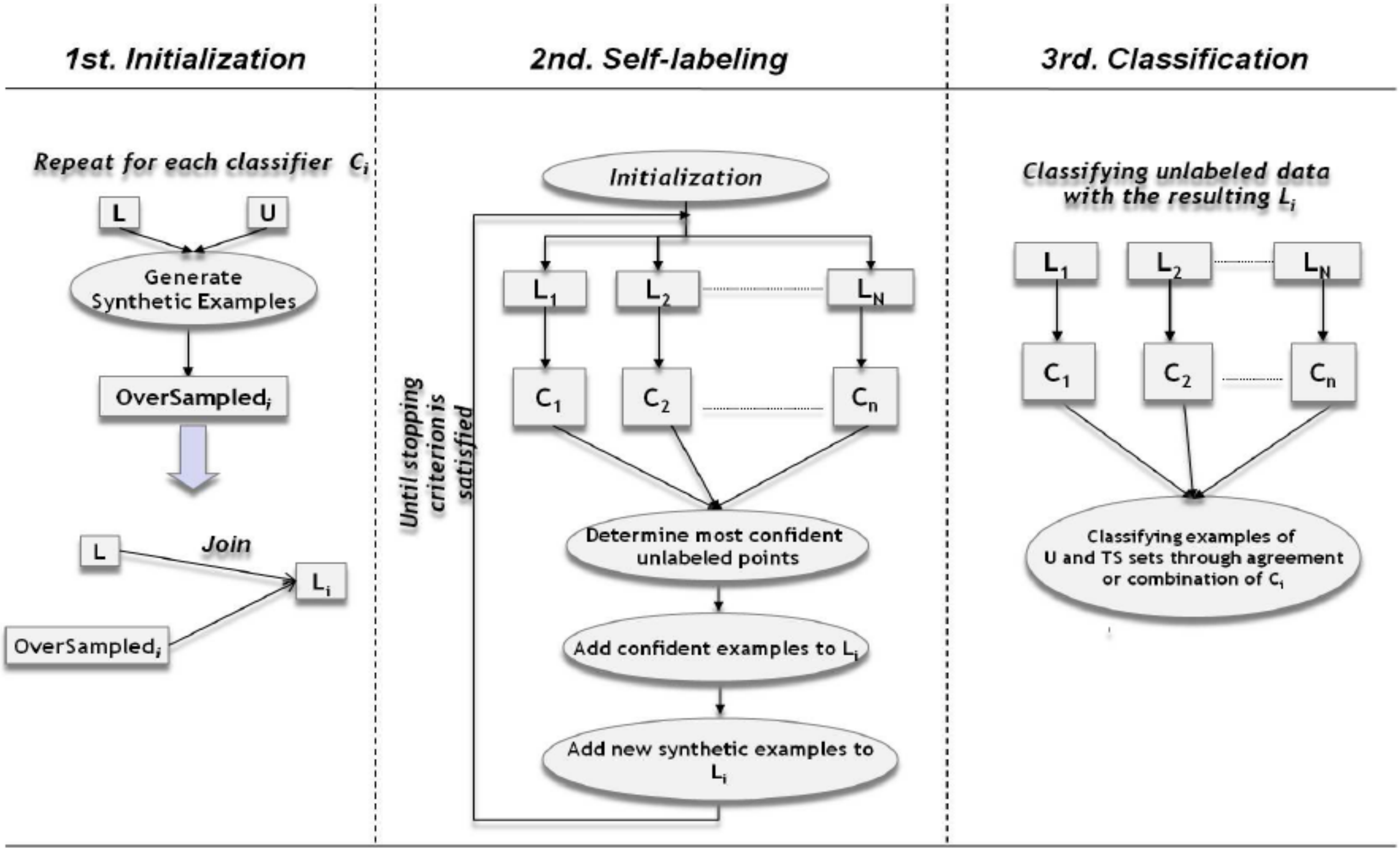
Motivation:

- Self-labeled techniques are limited by the number of labeled points and their distribution.
- Multiple classifier models use diversity mechanisms (bootstrapping). They behave as classical approaches when the number of labeled data is insufficient.

Objectives:

- To generate synthetic labeled data, aiming to:
    - Introduce diversity to multiple classifier approaches.
    - Fulfill the labeled data distribution.
- This synthetic data will be generated by PG models.

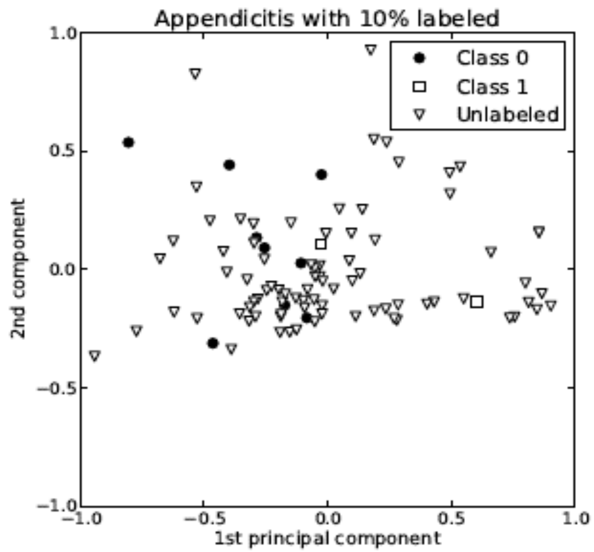
Self-Labeling: Our ideas

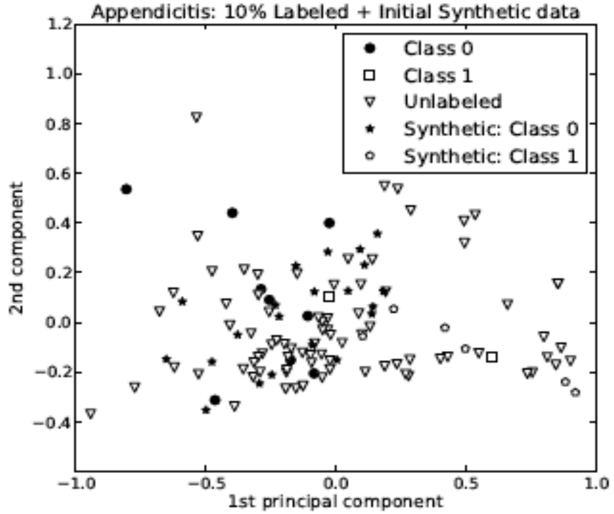


Self-Labeling: Our ideas

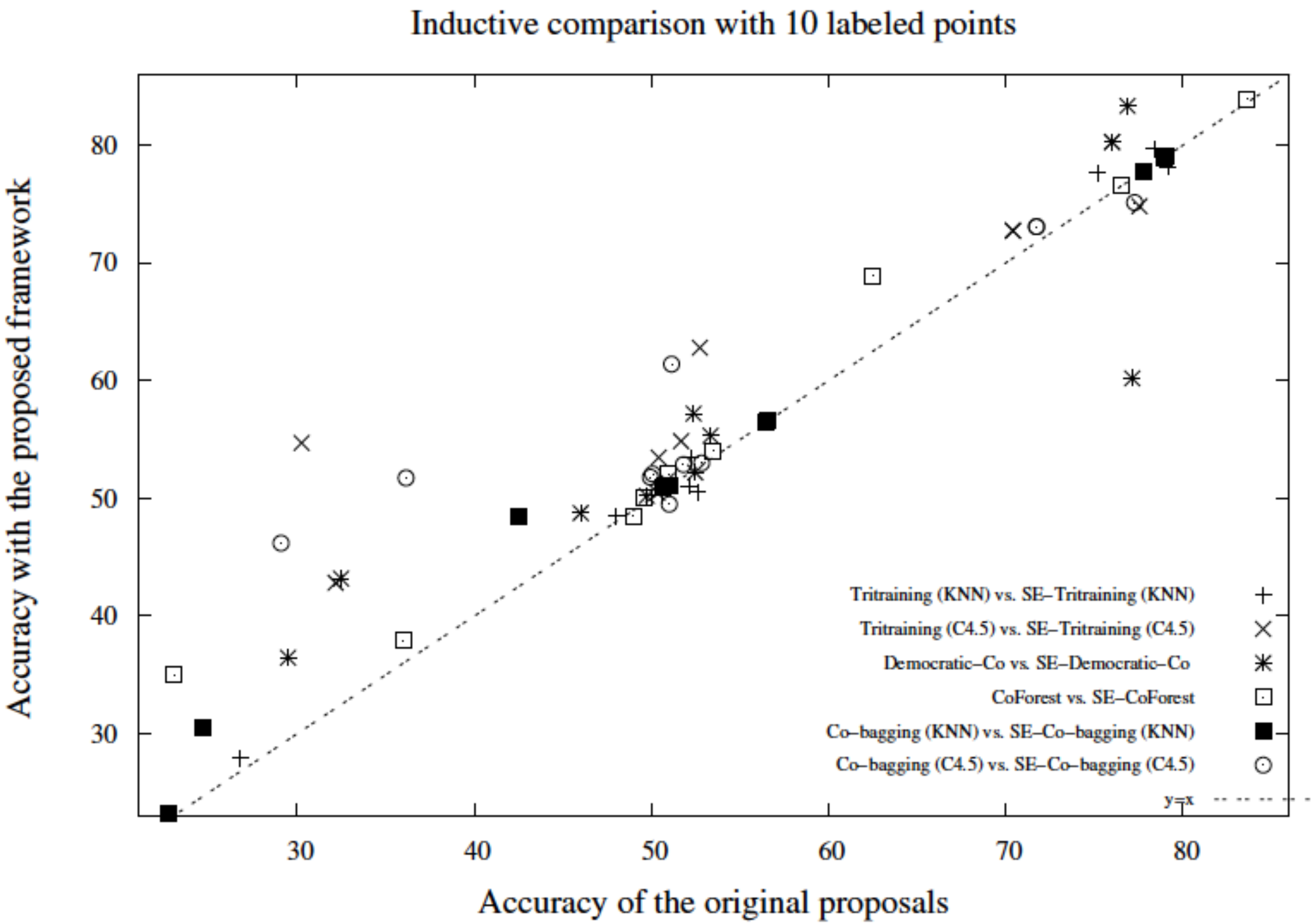
How can we generate new labeled data?

- Multi-class oversampling by using labeled and unlabeled examples.
- **Caveat**: It may generate noisy examples.
- **Solution**: Positioning adjustment based on DE.

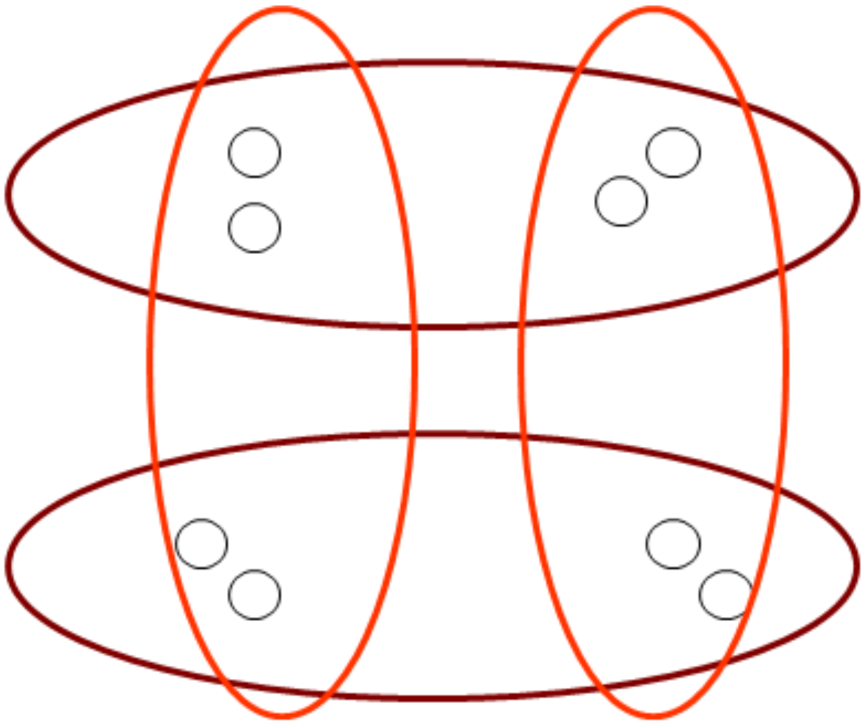




Experiments on high dimensional data sets:

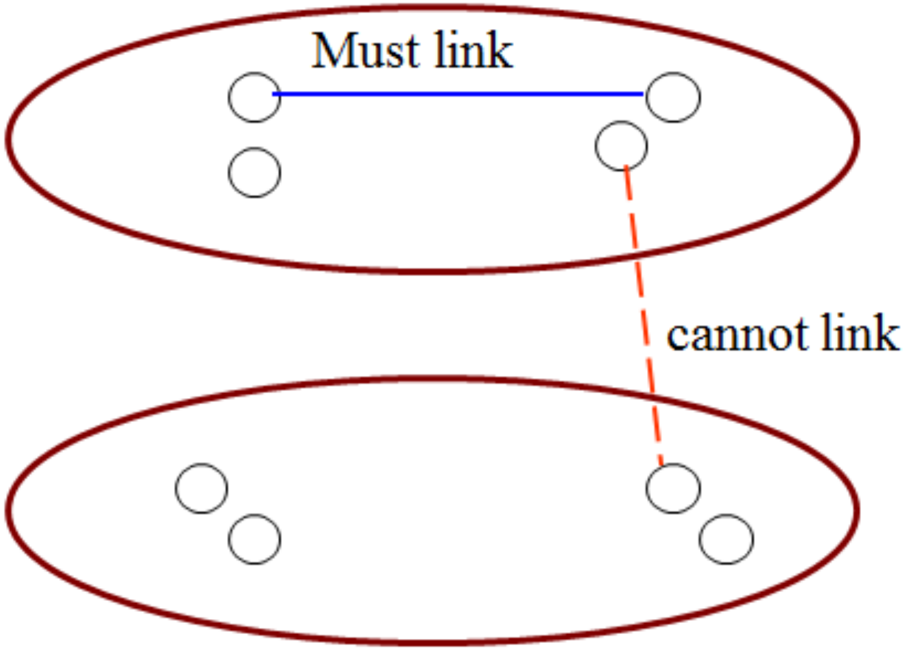


Semi-Supervised Clustering



- Clustering data into two clusters

Semi-Supervised Clustering



- Clustering data into two clusters
- Side information:
  - Must links vs. cannot links

Semi-Supervised Clustering

- Also called constrained clustering
- Two types of approaches
  - Restricted data partitions
  - Distance metric learning approaches

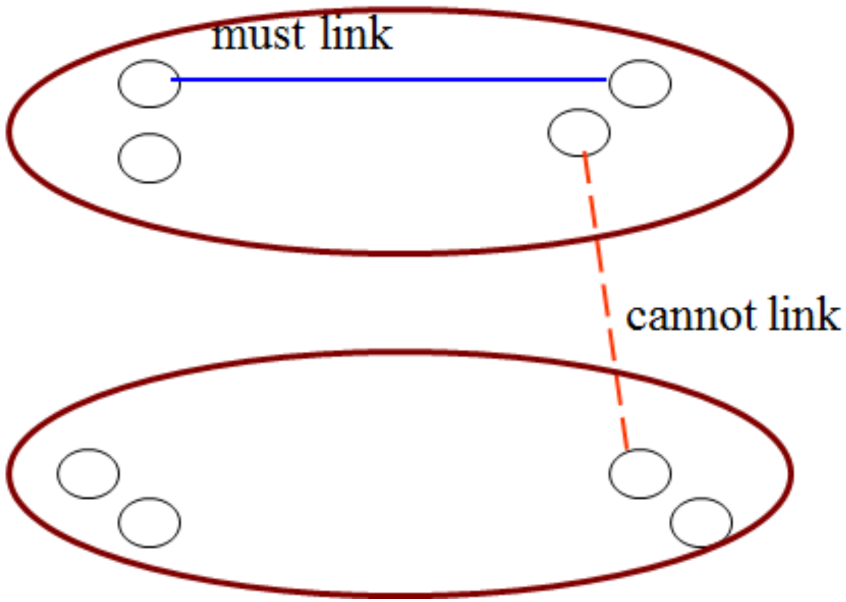
Restricted Data Partitions

- Require data partitions to be consistent with the given links
- Links → hard constraints
  - E.g. constrained K-Means (Wagstaff et al., 2001)
- Links → soft constraints
  - E.g., Metric Pairwise Constraints K-means (Basu et al., 2004)

Restricted Data Partitions

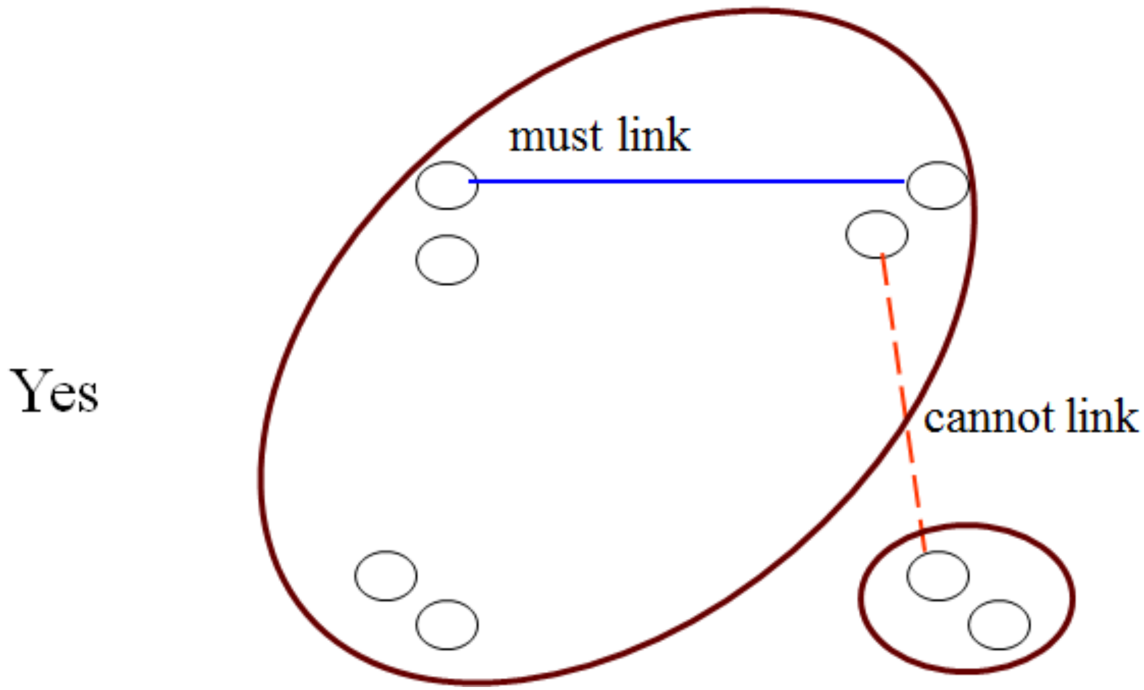
- Hard constraints
  - Cluster memberships must obey the link constraints

Yes



Restricted Data Partitions

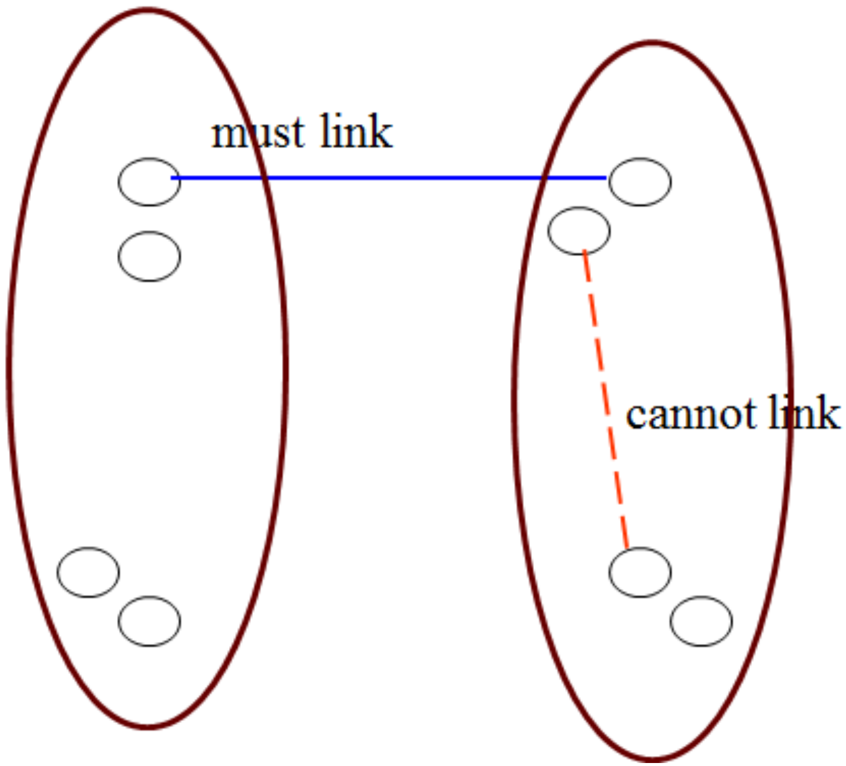
- Hard constraints
  - Cluster memberships must obey the link constraints



Restricted Data Partitions

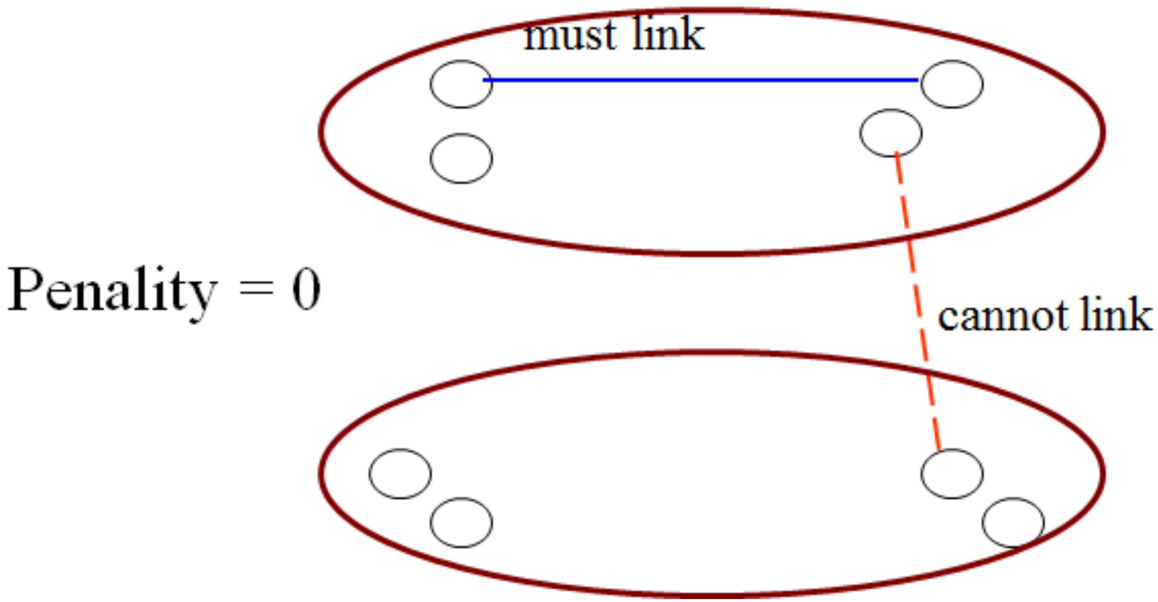
- Hard constraints
  - Cluster memberships **must** obey the link constraints

No



Restricted Data Partitions

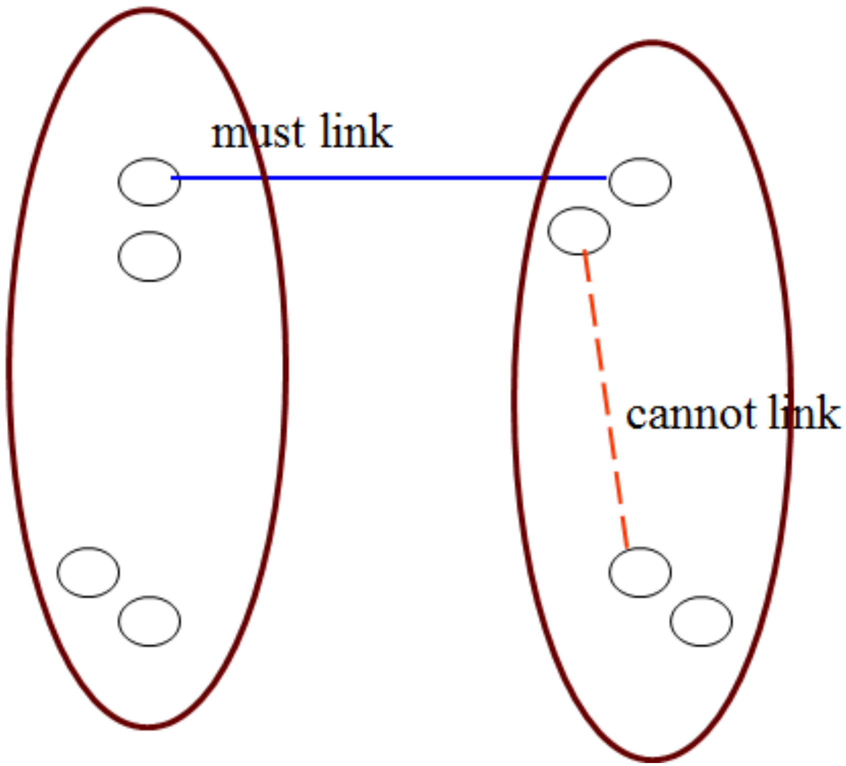
- Soft constraints
  - **Penalize** data clustering if it violates some links



Restricted Data Partitions

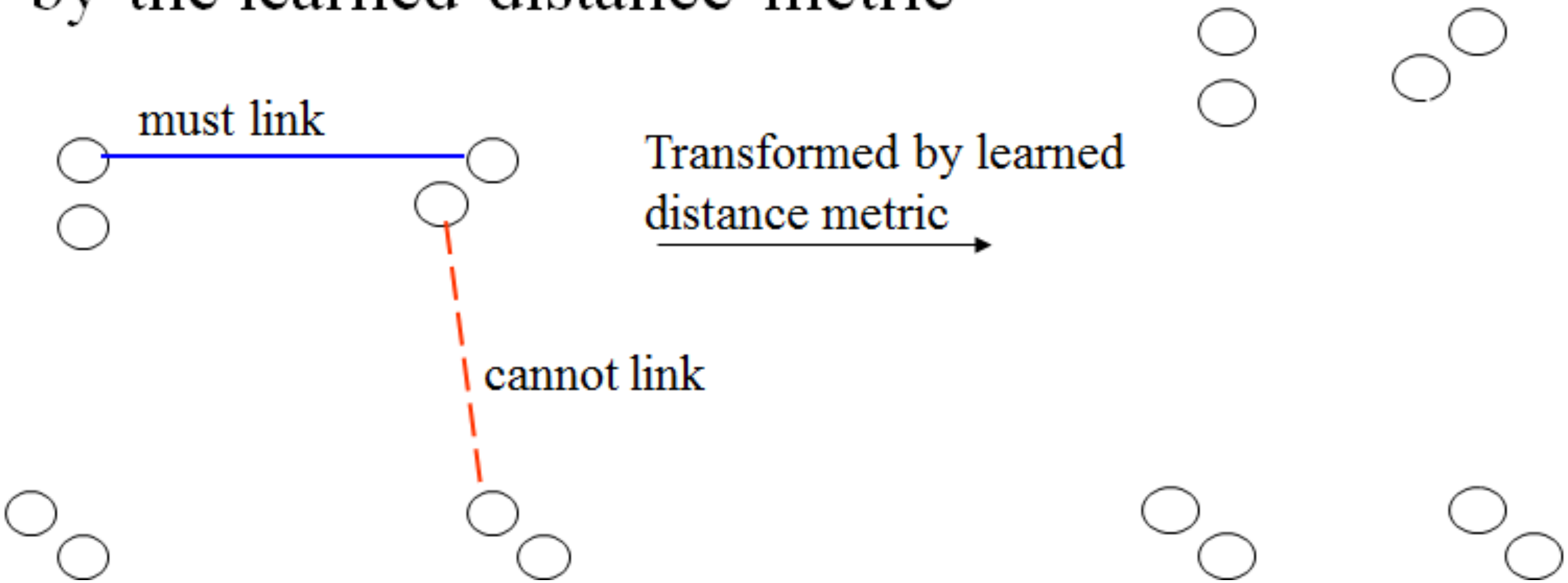
- Soft constraints
  - Cluster memberships must obey the link constraints

Penality = 1



Distance Metric Learning

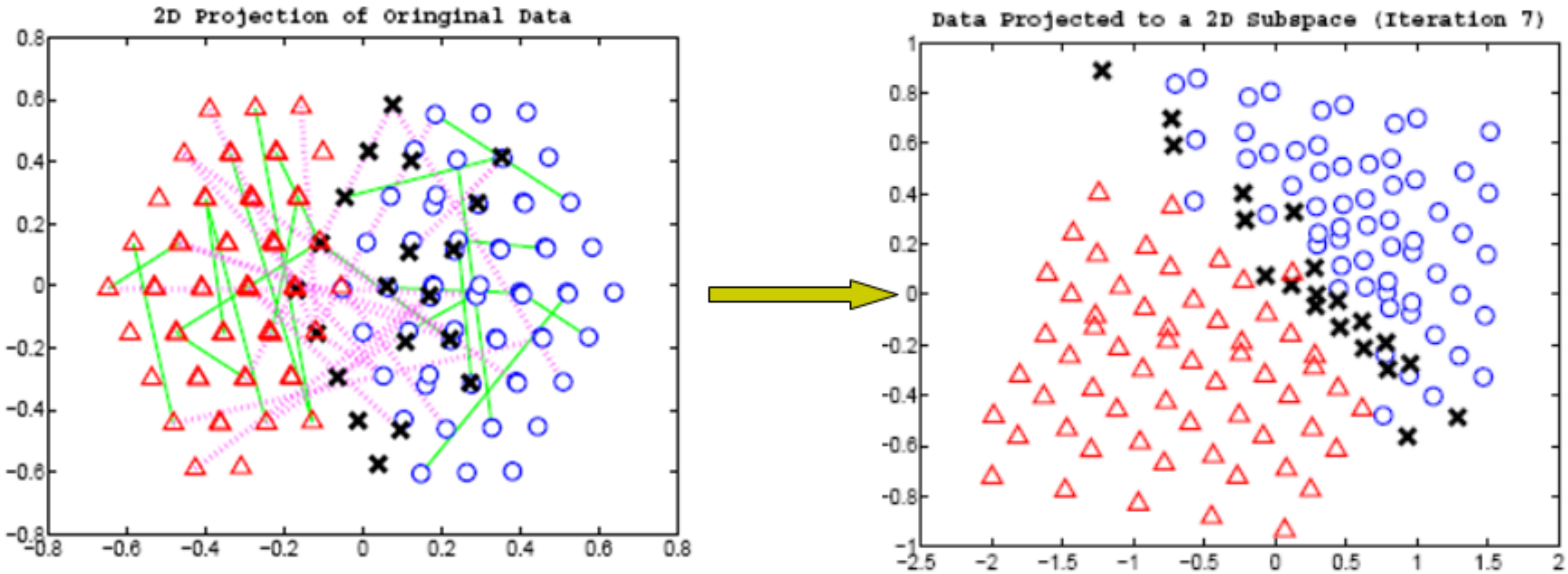
- Learning a distance metric from pairwise links
  - Enlarge the distance for a cannot-link
  - Shorten the distance for a must-link
- Applied K-means with pairwise distance measured by the learned distance metric



Distance Metric Learning

2D data projection using Euclidean distance metric

2D data projection using learned distance metric



Solid lines: must links

dotted lines: cannot links