

# Aprendizaje Semi-supervisado



Diplomatura en Ciencia de Datos,  
Aprendizaje Automático y sus Aplicaciones  
FaMAF-UNC  
agosto 2018

# Para saber más

Un buen tutorial de Jerry Zhu

<http://pages.cs.wisc.edu/~jerryzhu/pub/sslchicago09.pdf>

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

# Contexto

Los datos etiquetados son escasos y caros

Los datos no etiquetados son abundantes y gratis

Objetivo: aprender de datos etiquetados y no etiquetados, para obtener:

- Menos overfitting, mejor generalización
- Más capacidad para tratar ejemplos no vistos (del mismo universo, NO transfer learning)

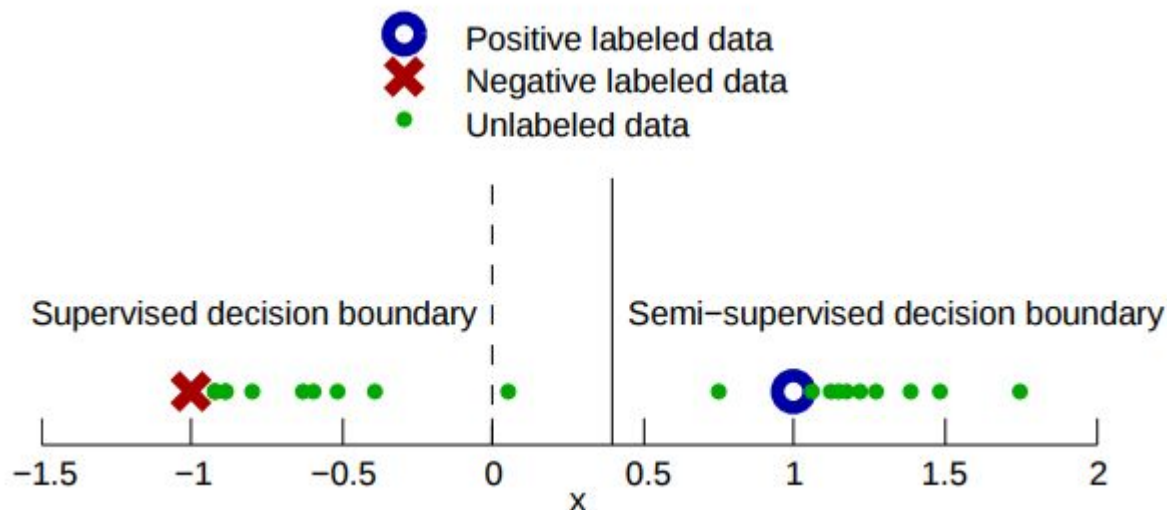
También: usar datos etiquetados para mejorar algoritmos no supervisados

- Clustering with rules
- Reglas de asociación con clase

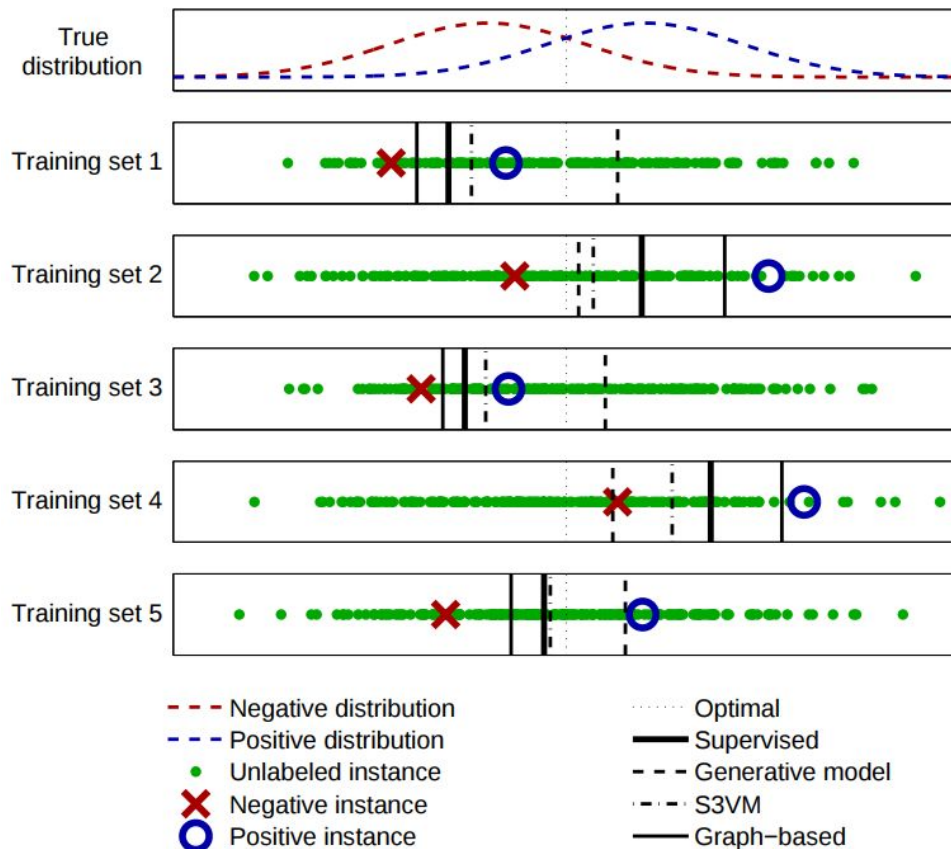
Fundamento cognitivo

Promesa: mejorar la  
performance gratis!

# Cómo ayudan los datos no etiquetados?



# Asunciones equivocadas... empeoran



# Notación

instance  $x$ , label  $y$  learner  $f : X \rightarrow Y$  labeled data  $(X_l, Y_l) = \{(x_1:l, y_1:l)\}$  unlabeled data  $X_u = \{x_{l+1:l+u}\}$ , available during training. Usually  $l \gg u$ . Let  $n = l + u$  test data  $\{(x_{n+1}, y_{n+1}), \dots, (x_n, y_n)\}$ , not available during training

Inductive semi-supervised learning: Given  $\{(x_i, y_i) \mid i=1, \dots, l\}$ ,  $\{x_j \mid l+1 \leq j \leq l+u\}$ , learn  $f : X \rightarrow Y$  so that  $f$  is expected to be a good predictor on future data, beyond  $\{x_j \mid l+1 \leq j \leq l+u\}$

Transductive learning: Given  $\{(x_i, y_i) \mid i=1, \dots, l\}$ ,  $\{x_j \mid l+1 \leq j \leq l+u\}$ , learn  $f : X \rightarrow Y$  so that  $f$  is expected to be a good predictor on the unlabeled data  $\{x_j \mid l+1 \leq j \leq l+u\}$ . Note  $f$  is defined only on the given training sample, and is not required to make predictions outside them.

# Modelos disjuntos vs. conjuntos

Aprender conjuntamente vs. concatenar módulos



# **Autoaprendizaje (self-learning) (bootstrapping)**

# Algoritmo de autoaprendizaje

1. Obtener un conjunto pequeño de datos etiquetados
  2. Aprender un clasificador de los datos etiquetados
  3. Aplicar el clasificador sobre datos no etiquetados
  4. Incorporar datos etiquetados automáticamente al conjunto de entrenamiento
  5. Volver a 2.
- 
- ¿Qué ejemplos etiquetados automáticamente incorporamos?
    - Mayor confianza
    - Los n mejores
    - Todos

# Un ejemplo: Yarowsky (1995)

Desambiguación de palabras

1. Ejemplos iniciales
2. Aprender una lista de decisión
3. Buscar más ejemplos con la lista
4. Iterar a 2.

# Un ejemplo: Yarowsky (1995)

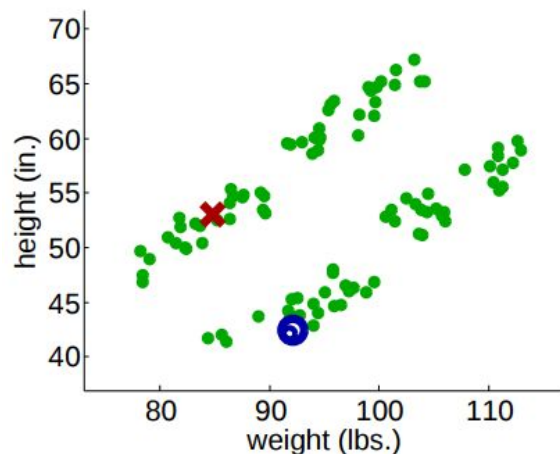
Desambiguación de palabras

One sense per collocation

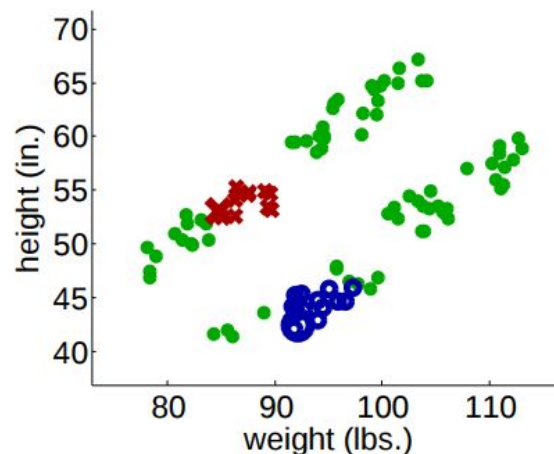
1. Ejemplos iniciales
2. Aprender una lista de decisión
3. Buscar más ejemplos con la lista
4. Iterar a 2.

One sense per discourse

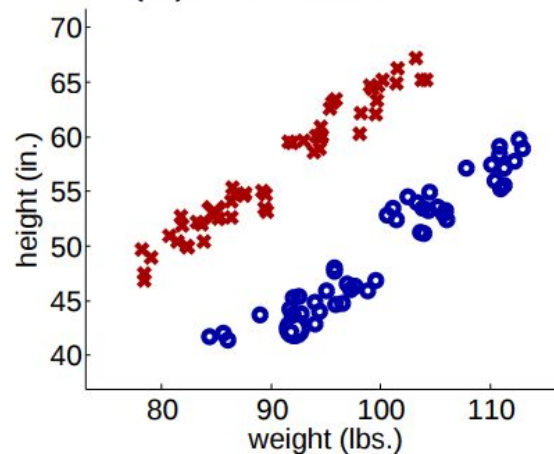
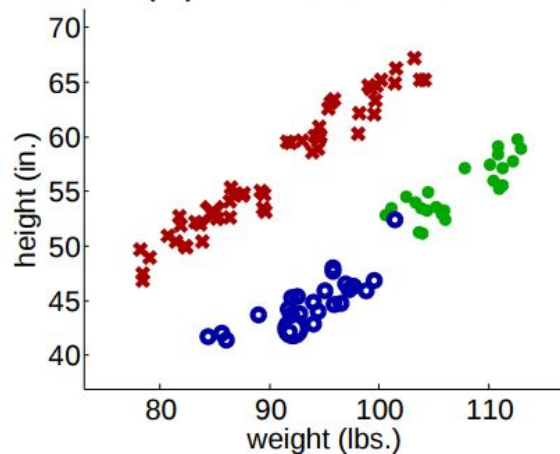
En cada documento, la misma palabra tiene siempre el mismo sentido

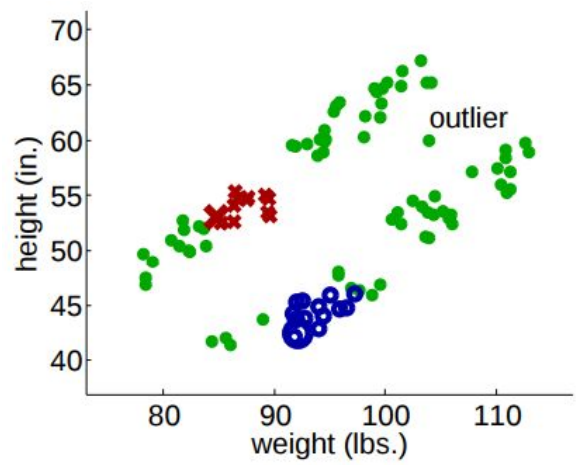


(a) Iteration 1

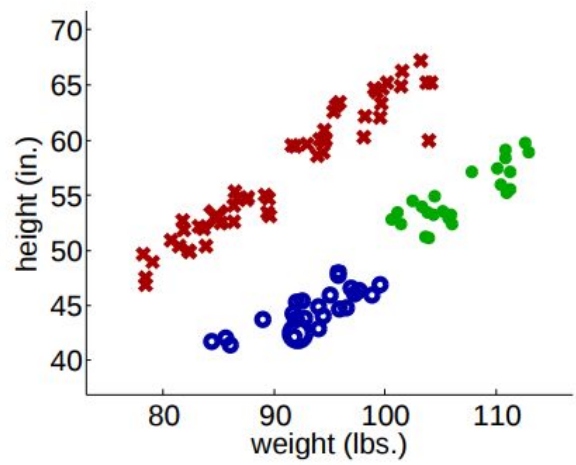


(b) Iteration 25

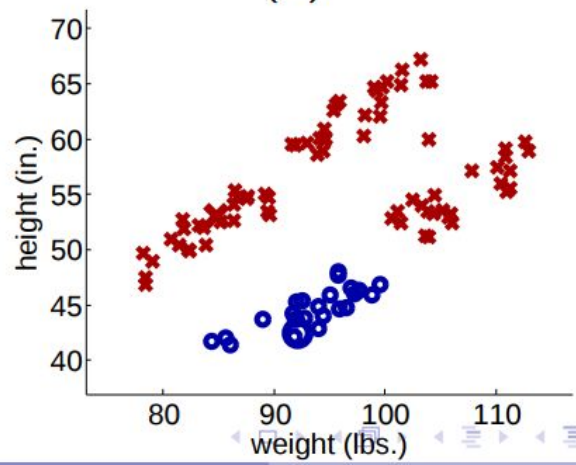
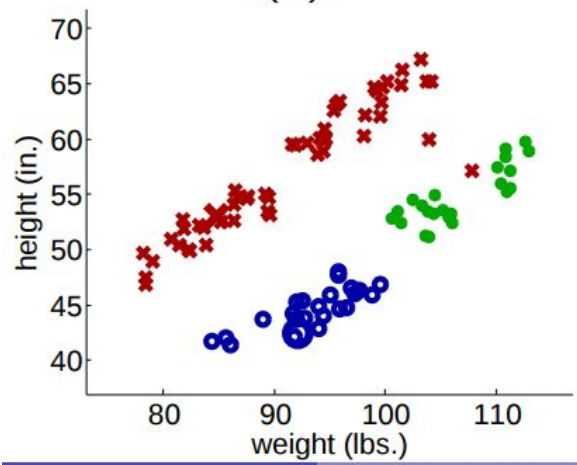




(a)



(b)



# Valoración de autoaprendizaje

## Ventajas:

- Muy fácil de implementar
- Se adapta a cualquier aprendedor (es un wrapper)
- Funciona muy bien para muchas tareas

## Desventajas:

- Amplificación del error ← estrategias correctivas
- Puede haber regiones del espacio a las que no llega ← estrategias complementarias

# **Co-aprendizaje (co-training)**



# Combinar estrategias complementarias

Aprendedores complementarios sobre diferentes facetas de un mismo objeto

- Página web / producto: imagen y texto
- Entidades nombradas: palabra y contexto

# Algoritmo de co-aprendizaje

1. Obtener un conjunto pequeño de datos etiquetados
  2. Aprender **dos** clasificadores **complementarios** de los datos etiquetados
  3. Aplicar los clasificadores sobre datos no etiquetados
  4. Incorporar datos etiquetados automáticamente al conjunto de entrenamiento
  5. ¿Eliminar datos etiquetados automáticamente del conjunto de entrenamiento?
  6. Volver a 2.
- ¿Qué ejemplos etiquetados automáticamente incorporamos?
    - Mayor confianza, uno solo, ambos?
    - Donde los dos clasificadores estén de acuerdo

# Valoración de co-aprendizaje

## Ventajas:

- Muy fácil de implementar
- Se adapta a cualquier aprendedor (es un wrapper)
- Funciona muy bien para muchas tareas

## Desventajas:

- Muchos problemas no se dividen bien en facetas disjuntas
- Es posible que un solo clasificador usando ambas facetas tenga mejor desempeño

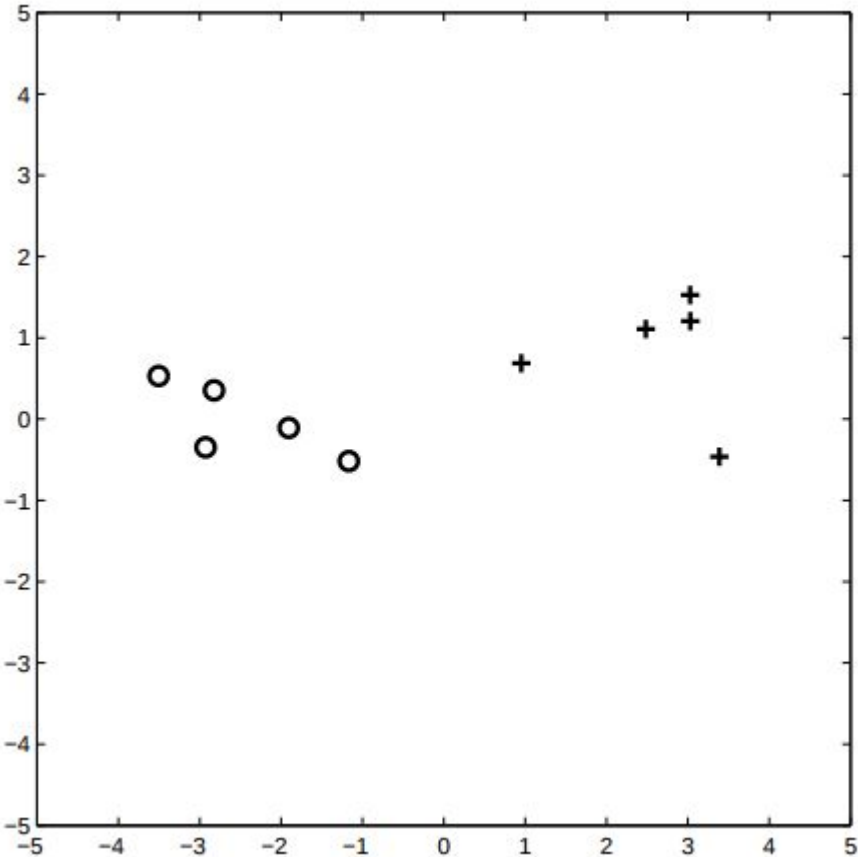
# **Modelos generativos**

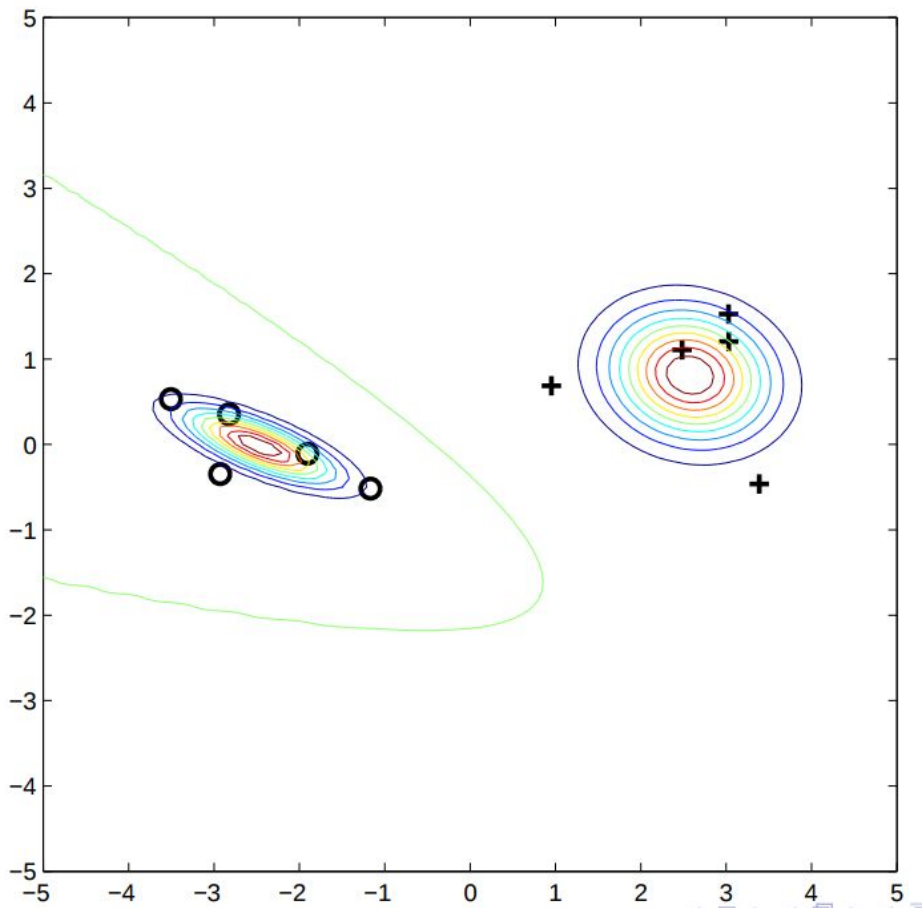
# Modelos generativos

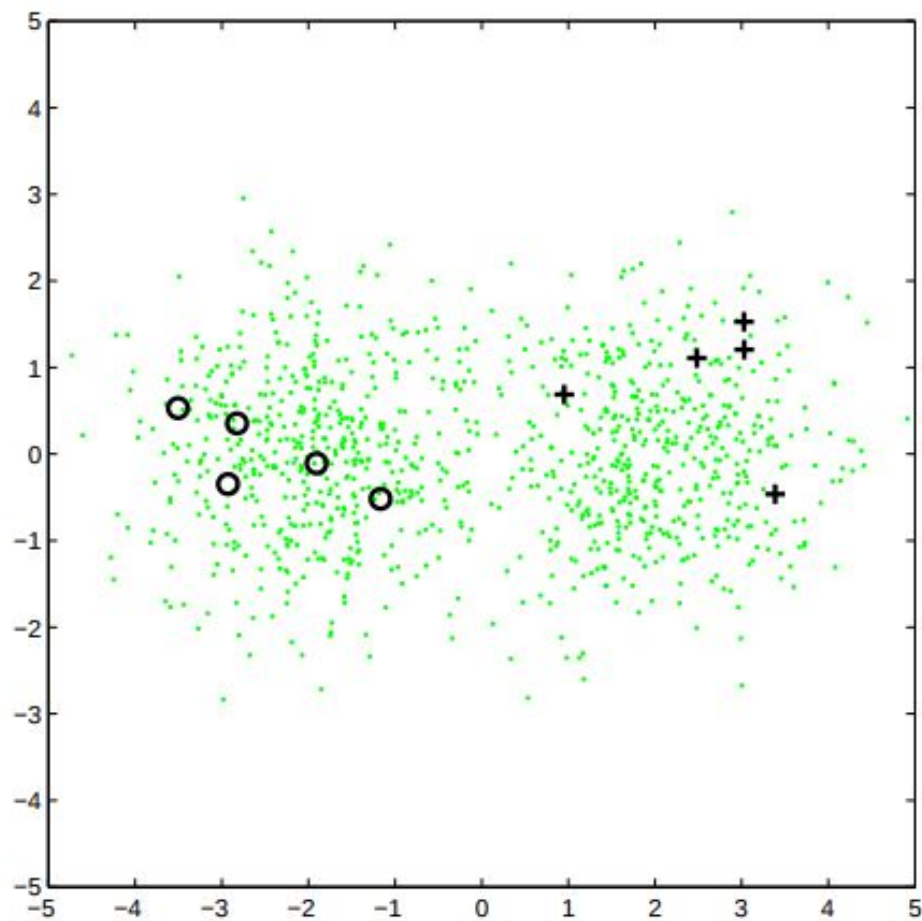
En el tutorial:

Modelos generativos con gaussianas

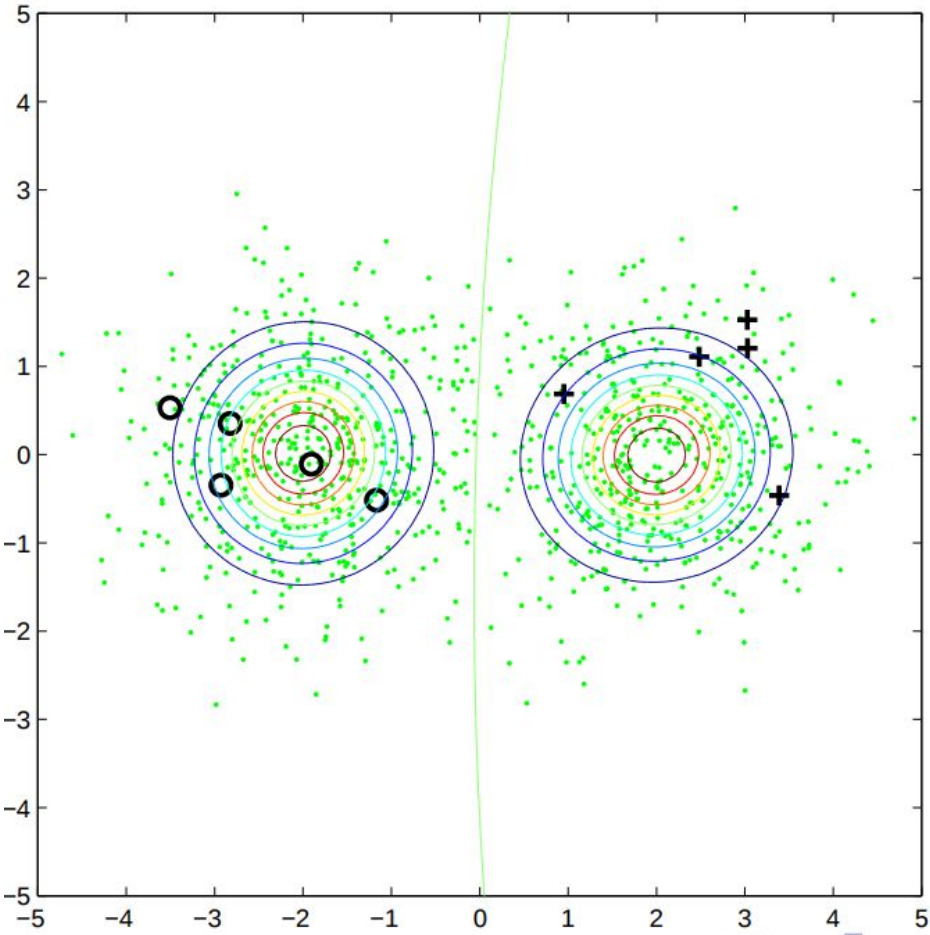
Usando Maximum Likelihood Estimation y Expectation Maximization



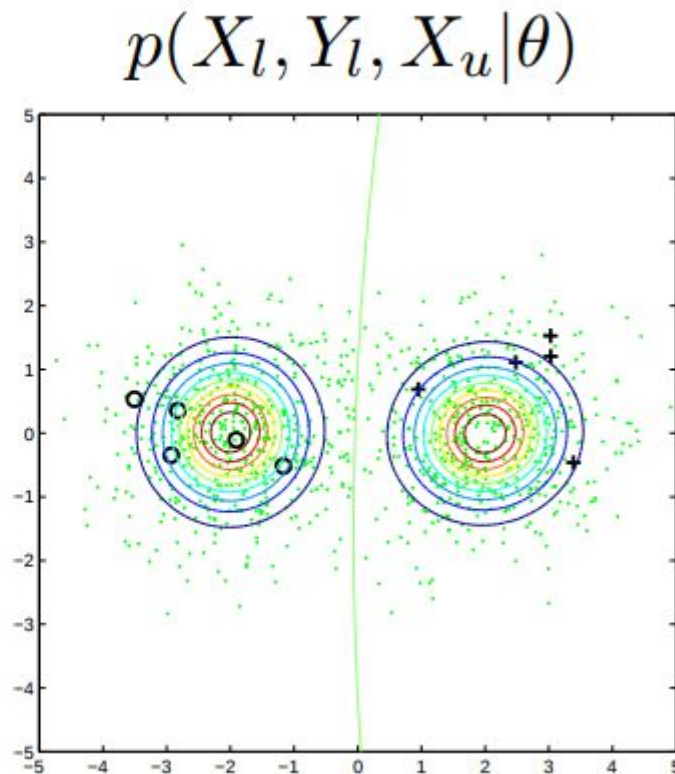
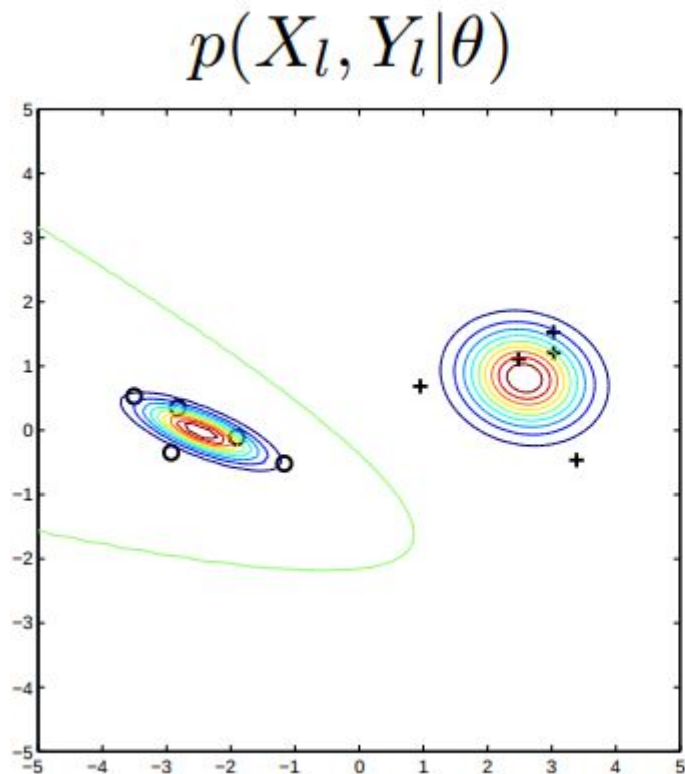








# Maximizar diferentes parámetros



# Cuánto podemos aprender?

No free lunch!

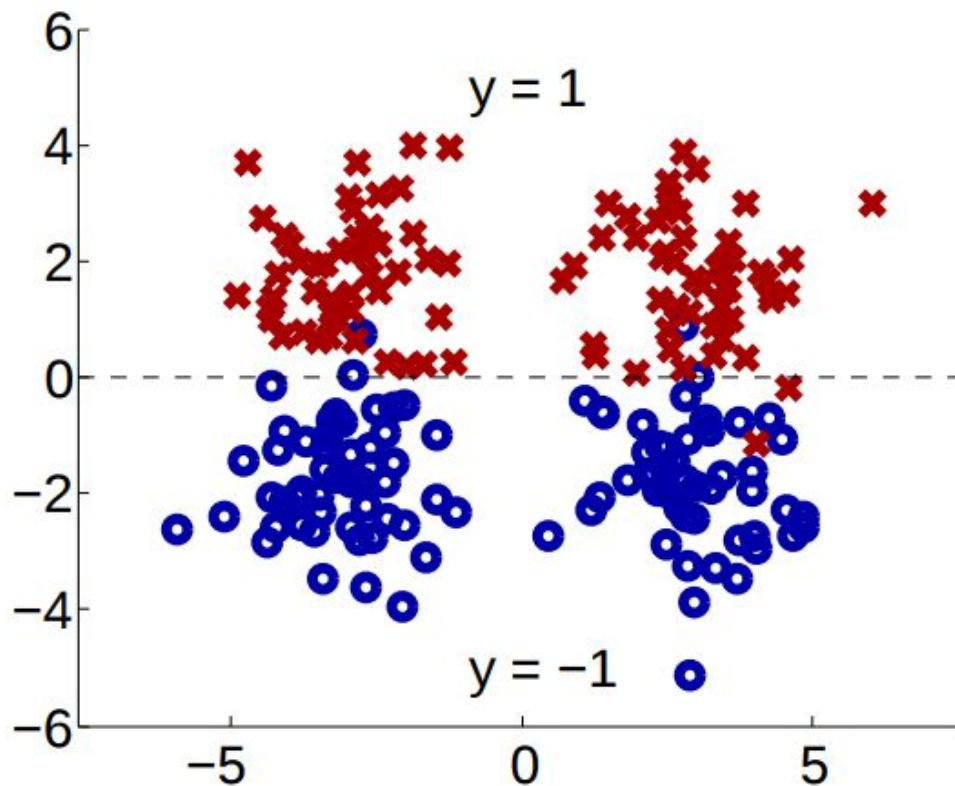
Si asumimos pocas cosas, ganamos poca información

Si asumimos muchas cosas, nos podemos equivocar

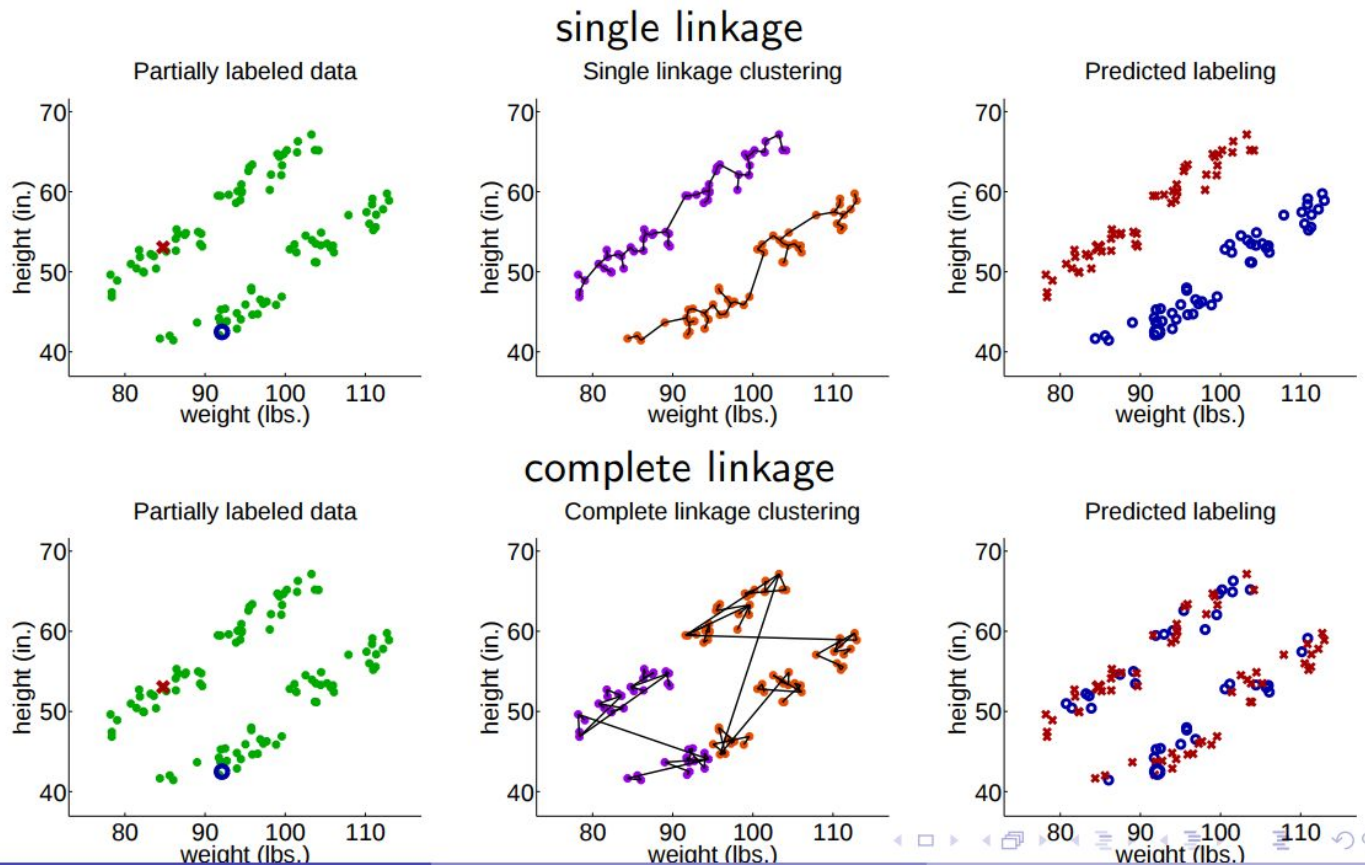
→ Mixtura de gaussianas

→ Modelos más complejos

# Un modelo simple no lo captura bien



# Relacionado: cluster-and-label



# Valoración de modelos generativos

Ventajas:

- Buen fundamento matemático
- Se obtiene un modelo generativo

Desventajas:

- Si la asunción está mal, el error es grande

# Modelos basados en grafos

	<i>d<sub>1</sub></i>	<i>d<sub>3</sub></i>	<i>d<sub>4</sub></i>	<i>d<sub>2</sub></i>
asteroid	●	●		
bright	●	●		
comet		●		
year				
zodiac				
.				
.				
airport				
bike				
camp			●	
yellowstone			●	●
zion				●



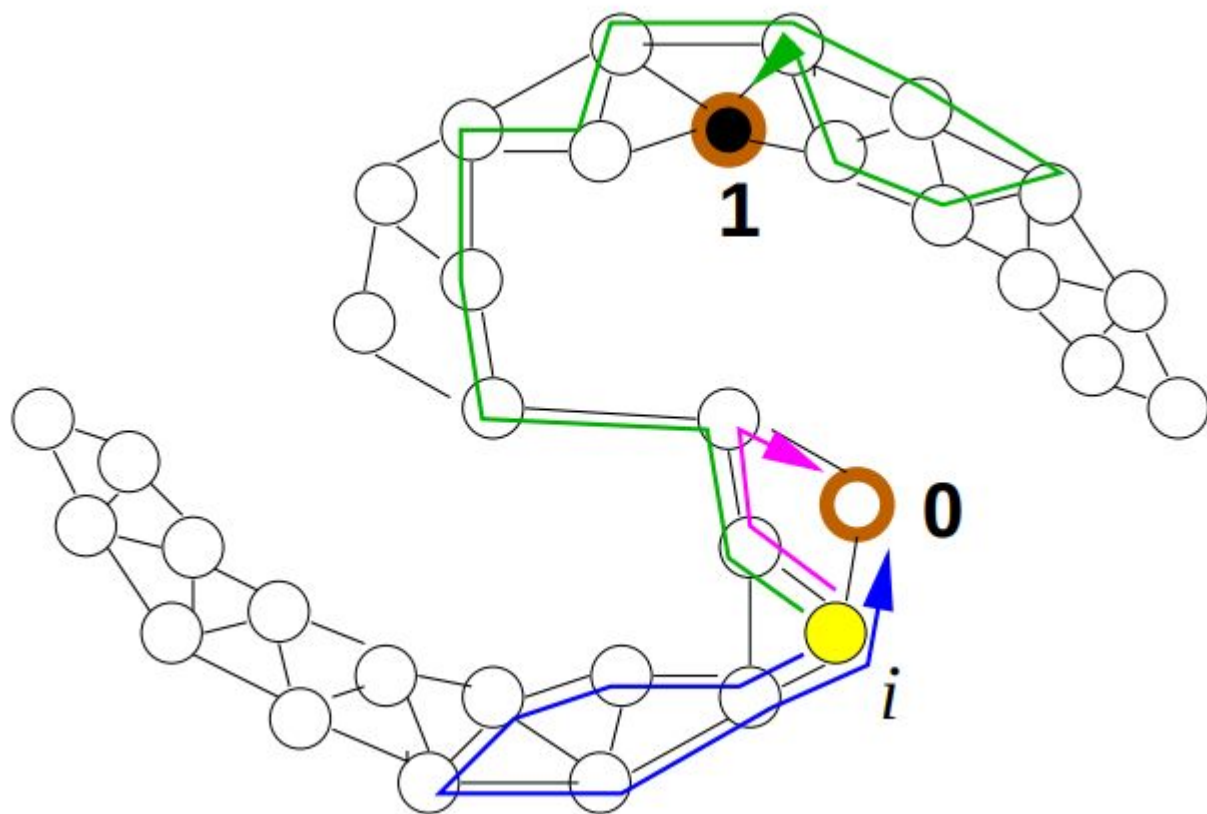
	<i>d<sub>1</sub></i>	<i>d<sub>3</sub></i>	<i>d<sub>4</sub></i>	<i>d<sub>2</sub></i>
asteroid	•			
bright	•			
comet				
year				
zodiac		•		
.				
.				
airport			•	
bike			•	
camp				
yellowstone				•
zion				•

[illegible]

---

d 2

d d d d 2 2



**Otros algoritmos**

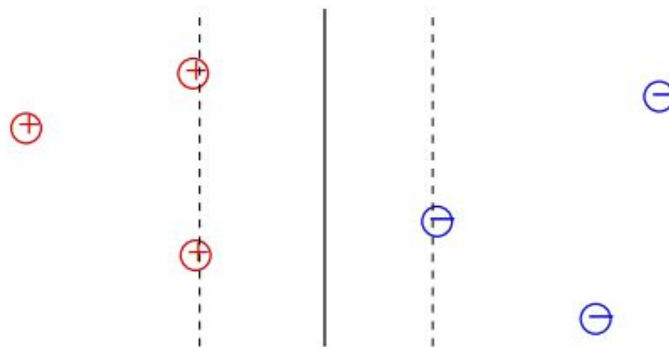
# Otros algoritmos

- Multiview learning
- Manifold learning
- Semi-supervised Support Vector Machines
- Ladder Networks
- Positive Unlabelled

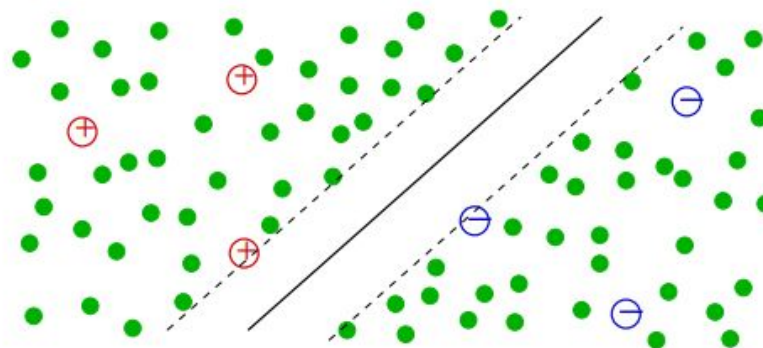
## Aproximaciones disjuntas

- Usar embeddings como pre-proceso
- Usar clusters para generalizar

# SVMs



Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)



# Active learning

1. Obtener un conjunto pequeño de datos etiquetados
  2. Aprender un clasificador de los datos etiquetados
  3. Aplicar el clasificador sobre datos no etiquetados
  4. Seleccionar los ejemplos que, de tener etiqueta manual, maximizarían el rendimiento del clasificador
  5. Un oráculo (humano) etiqueta los ejemplos, y se incorporan a los datos etiquetados
  6. Volver a 2
- Qué ejemplos maximizan aprendizaje? Con mayor incertidumbre? Más representativos?
  - Combinar con self-learning



# Supervisado → No supervisado

Usar datos etiquetados para mejorar algoritmos no supervisados

- Clustering with rules
- Constrained Clustering
- Reglas de asociación con clase
- K-nn con etiquetas de usuarios, etiquetas de items
- Etiquetas sobre los datos