Semi-Supervised Learning

Aarti Singh

Machine Learning 10-701/15-781
April 19, 2010

Slides Courtesy: Jerry Zhu

HW, Exam & Project

- HW 5: out today, due April 26 (Monday) @ beginning of class
- Project Poster Session: May 4 (Tuesday), 3-6 pm, NSH Atrium
- Project Report: May 5 (Wednesday) @ midnight by email
- Exam: May 7 (Friday), 5:30-8:30 pm, DH 2302

Supervised Learning

Feature Space $\mathcal{X}$ **Label** Space $\mathcal{Y}$

Goal: Construct a **predictor** $f : \mathcal{X} \to \mathcal{Y}$ to minimize

$$R(f) \equiv \mathbb{E}_{XY} [\text{loss}(Y, f(X))]$$

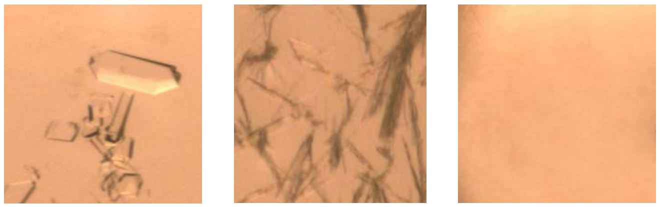
Optimal predictor (Bayes Rule) depends on unknown P_{XY}, so instead *learn* a good prediction rule from training data $\{(X_i, Y_i)\}_{i=1}^n \overset{\text{iid}}{\sim} P_{XY}$ (unknown)

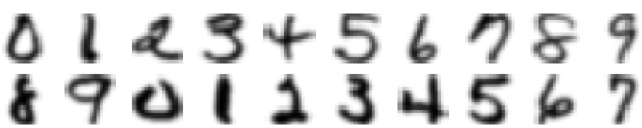
Training data ⇒ **Learning algorithm** ⇒ Prediction rule

$\{(X_i, Y_i)\}_{i=1}^n$ $\widehat{f}_n$

Labeled

Labeled and Unlabeled data





Unlabeled data, X_i

Cheap and abundant !

Human expert / Special equipment / Experiment

"Crystal" "Needle" "Empty"

"0" "1" "2" …

"Sports"
"News"
"Science"
…

Labeled data, Y_i

Expensive and scarce !

Example: Hard to obtain labels

Task: speech analysis

- Switchboard dataset
- telephone conversation transcription
- 400 hours annotation time for each hour of speech

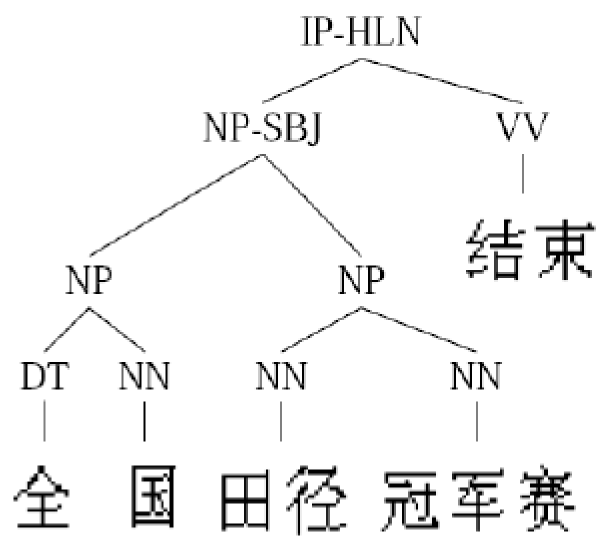
film $\Rightarrow$ `f ih_n uh_gl_n m`

be all $\Rightarrow$ `bcl b iy iy_tr ao_tr ao l_dl`

Example: Hard to obtain labels

Task: natural language parsing

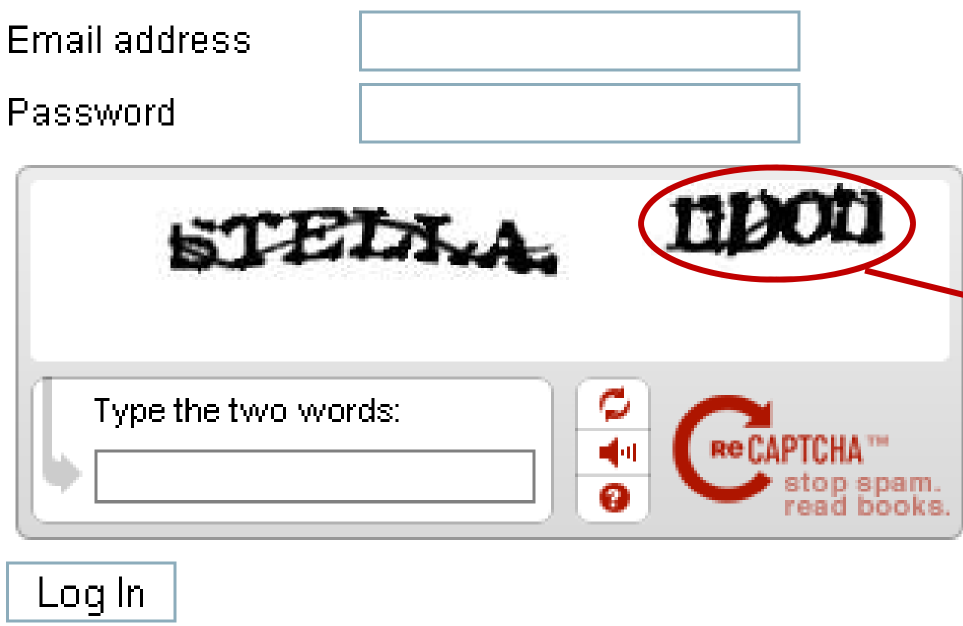
- Penn Chinese Treebank
- 2 years for 4000 sentences



"The National Track and Field Championship has finished."

Free-of-cost labels?

Luis von Ahn: Games with a purpose (ReCaptcha)



Word rejected by OCR
(Optical Character Recogintion)

You provide a free label!

Semi-Supervised learning

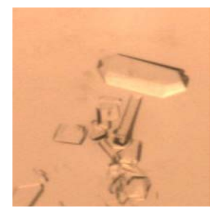
Training data ⇒ **Learning algorithm** ⇒ Prediction rule

$\{(X_i, Y_i)\}_{i=1}^{n}$ $\qquad \widehat{f}_{n,m}$

$\{X_i\}_{i=1}^{m}$

Supervised learning (SL)

Labeled data $\{X_i, Y_i\}_{i=1}^{n}$



"Crystal"

X_i $\qquad Y_i$

Semi-Supervised learning (SSL)

Labeled data $\{X_i, Y_i\}_{i=1}^{n}$ **and** Unlabeled data $\{X_i\}_{i=1}^{m}$ $\qquad m \gg n$

Goal: Learn a better prediction rule than based on labeled data alone.

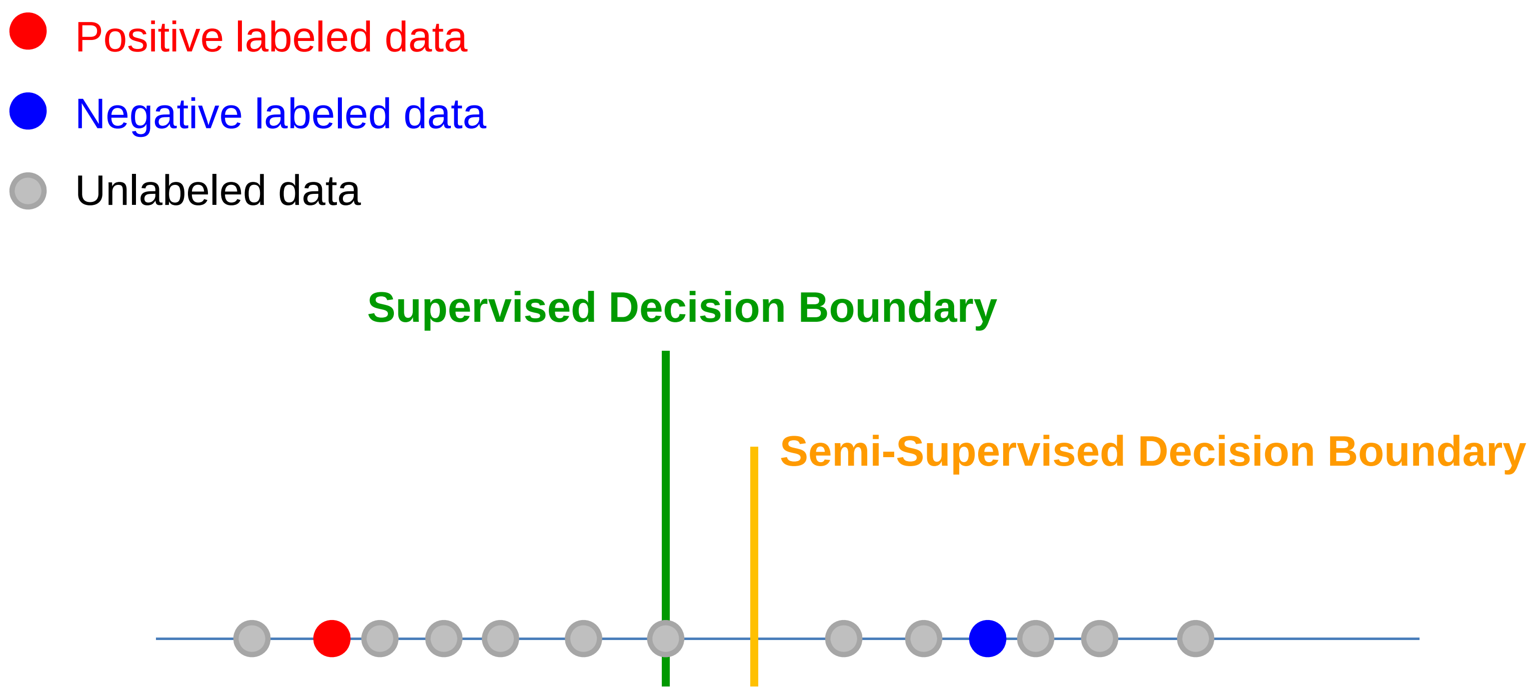
Semi-Supervised learning in Humans

Cognitive science

Computational model of how humans learn from labeled and unlabeled data.

- concept learning in children: $\mathbf{x}$=animal, y=concept (e.g., dog)
- Daddy points to a brown animal and says "dog!"
- Children also observe animals by themselves

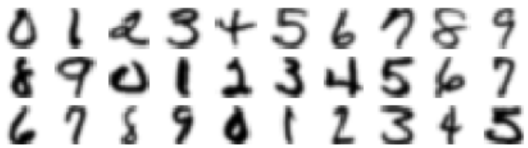
Can unlabeled data help?

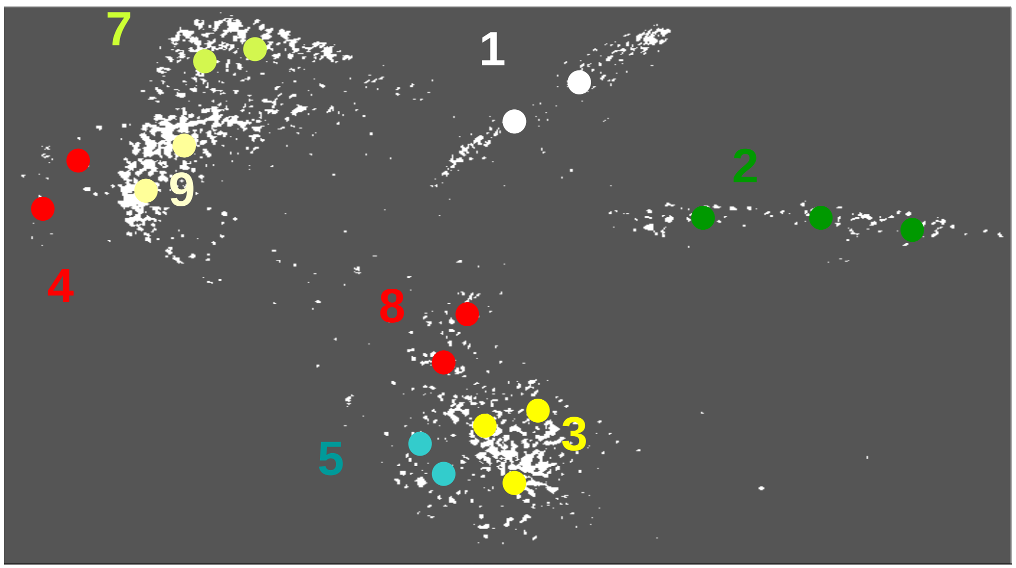


Assume each class is a coherent group (e.g. Gaussian)

Then unlabeled data can help identify the boundary more accurately.

Can unlabeled data help?

Unlabeled Images  Labels “0” “1” “2” …



“Similar” data points have “similar” labels

Self-training

Our first SSL algorithm:

Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
1. Initially, let $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
2. Repeat:
3. Train f from L using supervised learning.
4. Apply f to the unlabeled instances in U.
5. Remove a subset S from U; add $\{(\mathbf{x}, f(\mathbf{x})) | \mathbf{x} \in S\}$ to L.

Self-training is a *wrapper* method

- the choice of learner for f in step 3 is left completely open
- good for many real world tasks like natural language processing
- but mistake by f can reinforce itself

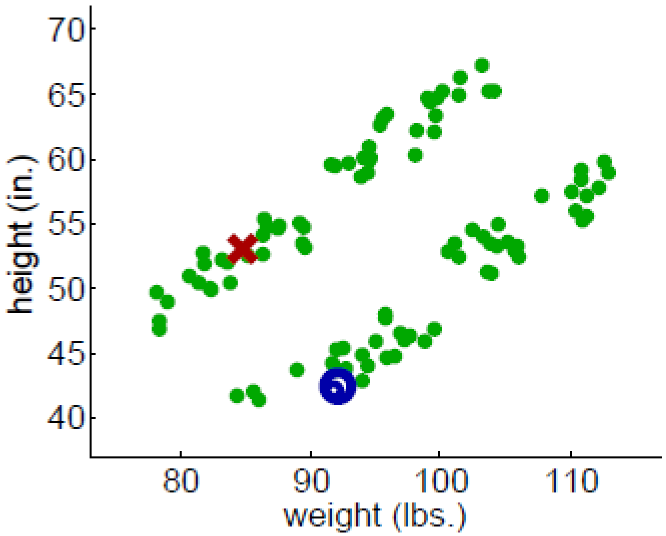
Self-training Example

Propagating 1-NN

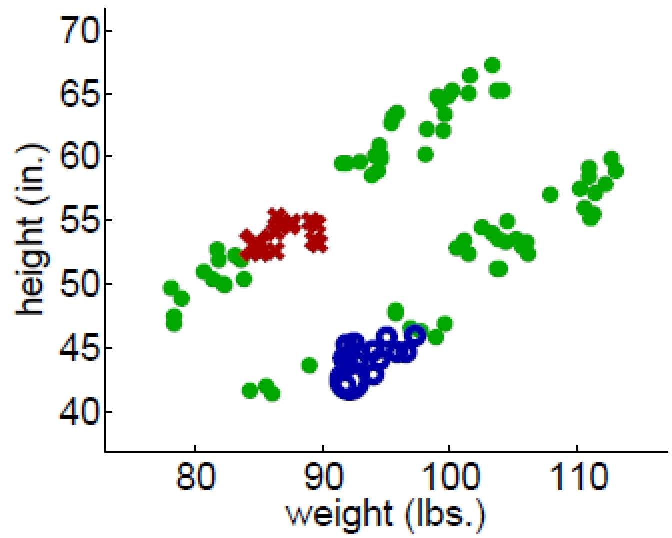
Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, distance function $d()$.

1. Initially, let $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
2. Repeat until U is empty:
3. Select $\mathbf{x} = \mathrm{argmin}_{\mathbf{x} \in U} \min_{\mathbf{x}' \in L} d(\mathbf{x}, \mathbf{x}')$.
4. Set $f(\mathbf{x})$ to the label of $\mathbf{x}$'s nearest instance in L. Break ties randomly.
5. Remove $\mathbf{x}$ from U; add $(\mathbf{x}, f(\mathbf{x}))$ to L.

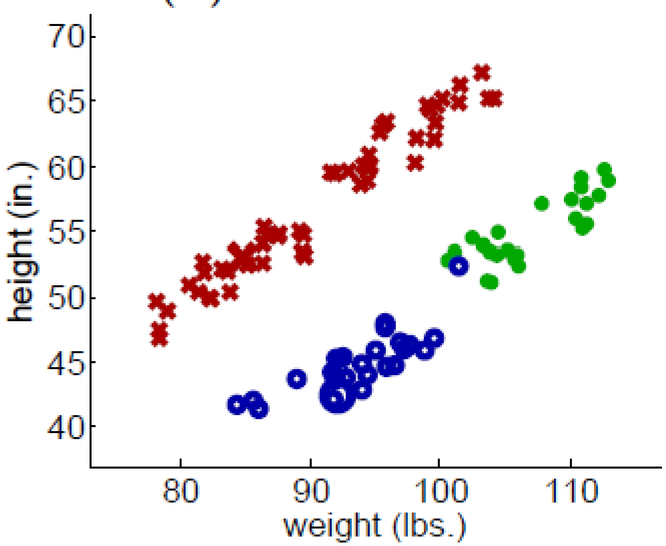
Propagating 1-Nearest-Neighbor: now it works



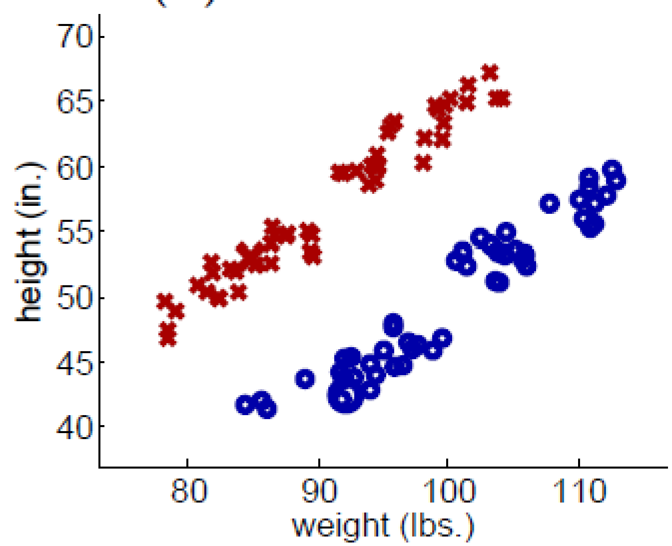
(a) Iteration 1



(b) Iteration 25



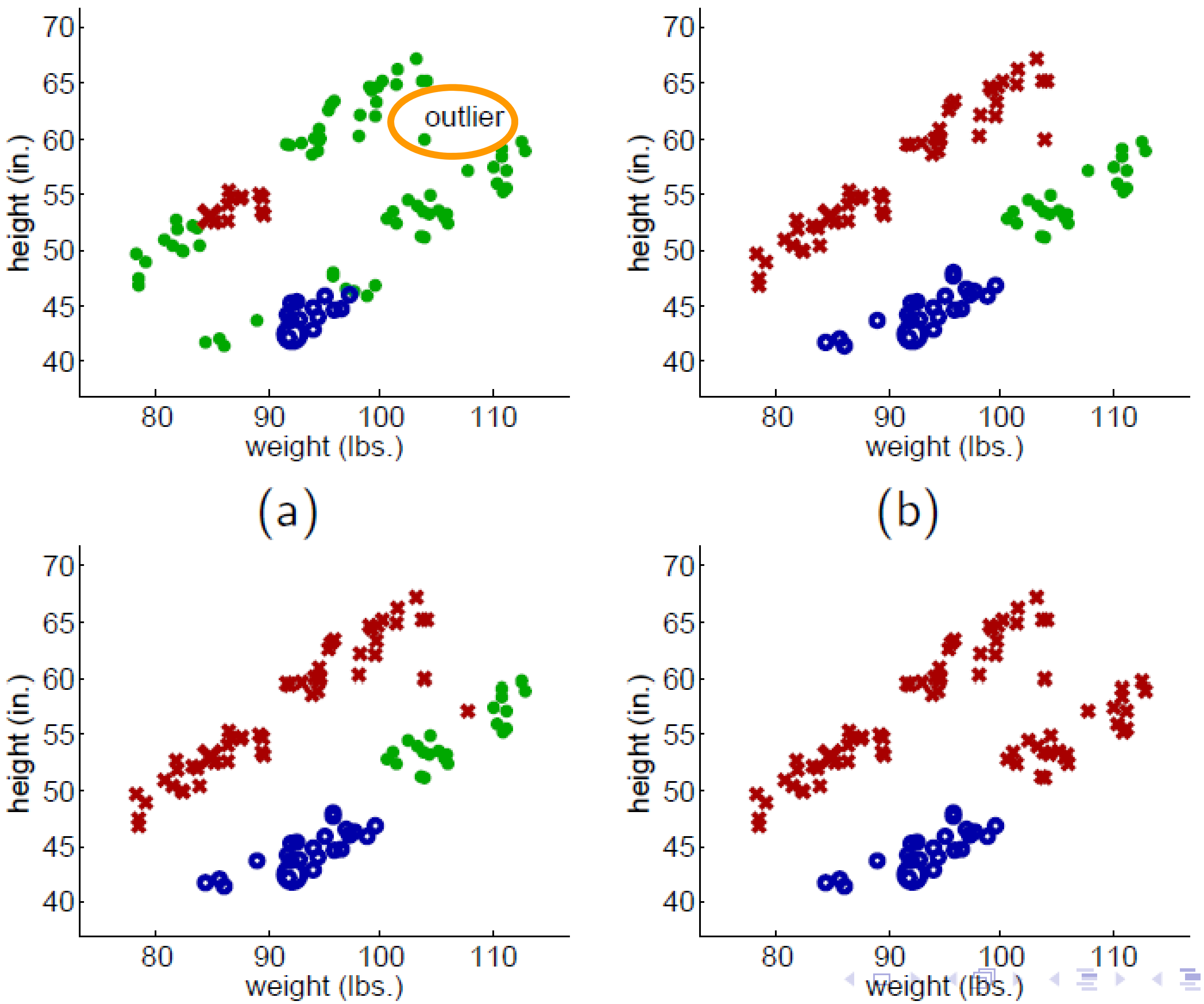
(c) Iteration 74



(d) Final labeling of all instances

Propagating 1-Nearest-Neighbor: now it doesn't

But with a single outlier...



Some SSL Algorithms

- Generative methods – assume a model for p(x,y) and maximize joint likelihood

  Mixture models

- Multi-view methods – multiple independent learners that agree on prediction for unlabeled data

  Co-training

- Graph-based methods – assume the target function p(y|x) is smooth wrt a graph or manifold

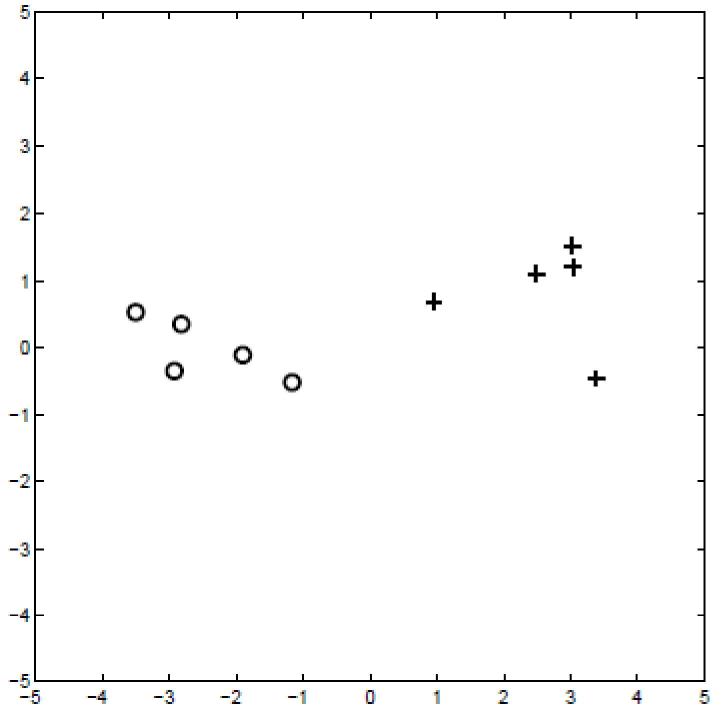
  Graph/Manifold Regularization

Some SSL Algorithms

- Generative methods – assume a model for p(x,y) and maximize joint likelihood
  Mixture models
- Multi-view methods – multiple independent learners that agree on prediction for unlabeled data
  Co-training
- Graph-based methods – assume the target function p(y|x) is smooth wrt a graph or manifold
  Graph/Manifold Regularization

Mixture Models

Labeled data (X_l, Y_l):



Assuming each class has a Gaussian distribution, what is the decision boundary?

Mixture Models

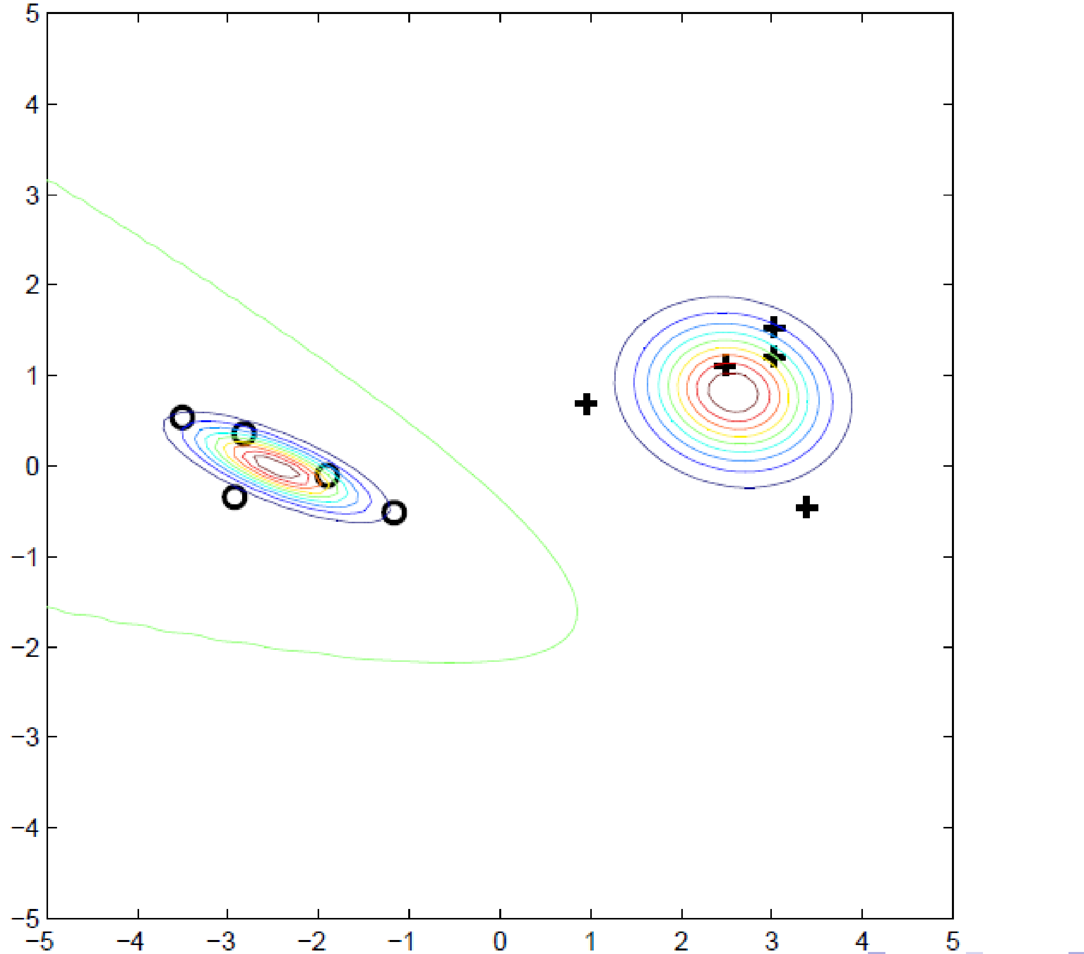
Model parameters: $\theta = \{w_1, w_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$
The GMM:

$$\begin{aligned} p(x, y|\theta) &= p(y|\theta)p(x|y,\theta) \\ &= w_y \mathcal{N}(x; \mu_y, \Sigma_y) \end{aligned}$$

Classification: $p(y|x,\theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)}$ $\gtrless$ 1/2

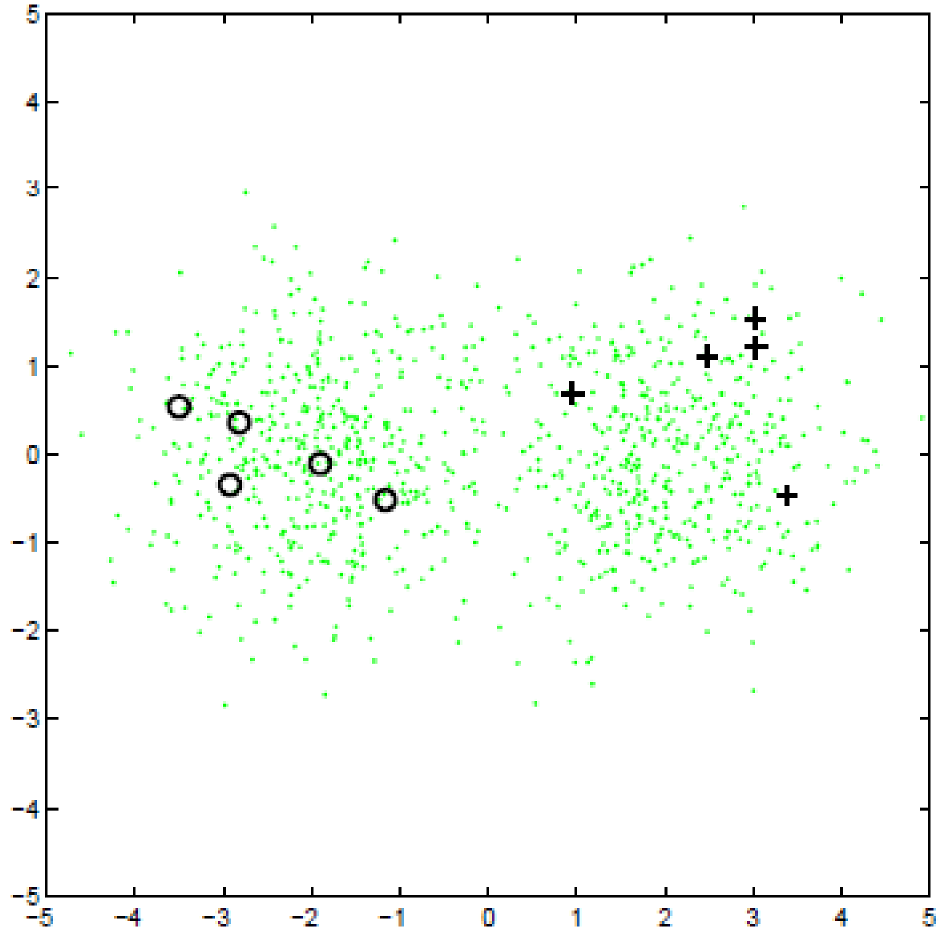
Mixture Models

The most likely model, and its decision boundary:



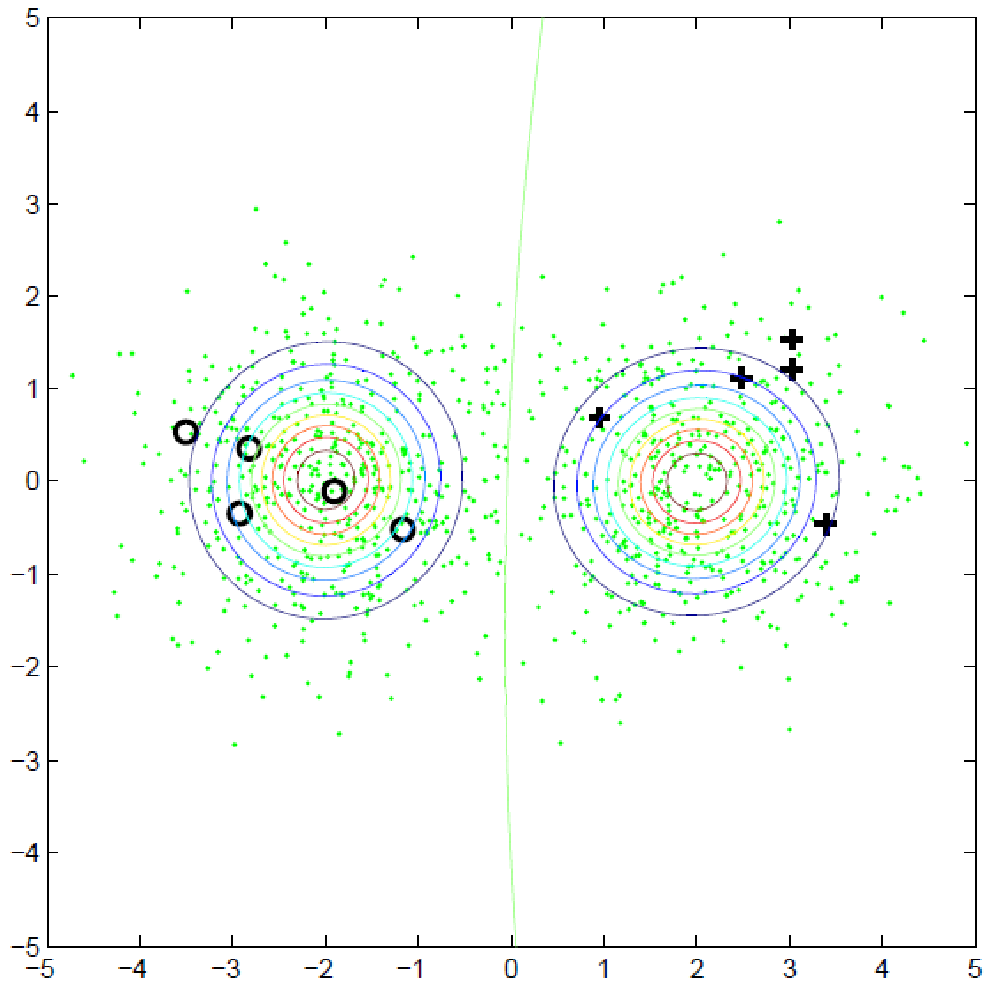
Mixture Models

Adding unlabeled data:



Mixture Models

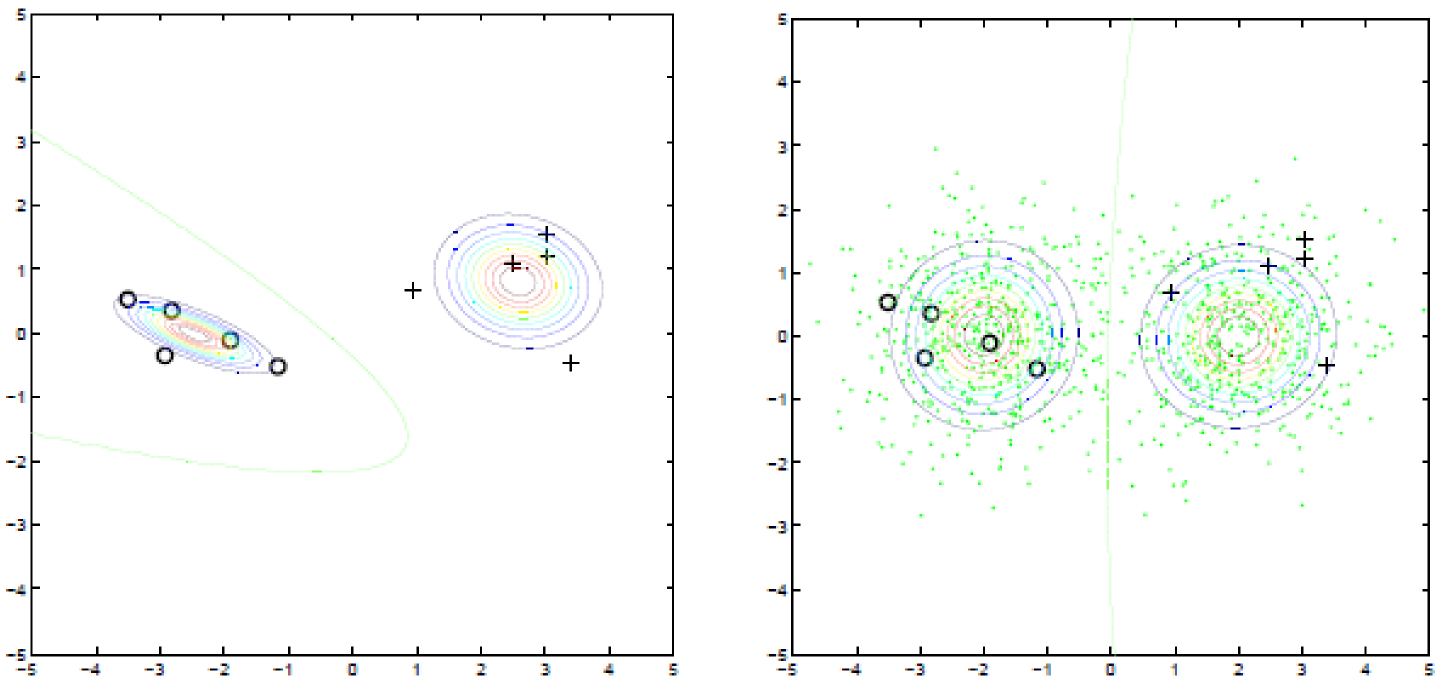
With unlabeled data, the most likely model and its decision boundary:



Mixture Models

They are different because they maximize different quantities.

$p(X_l, Y_l|\theta)$ $\qquad$ $p(X_l, Y_l, X_u|\theta)$



Mixture Models

Assumption

knowledge of the model form $p(X, Y|\theta)$.

- joint and marginal likelihood

$$p(X_l, Y_l, X_u|\theta) = \sum_{Y_u} p(X_l, Y_l, X_u, Y_u|\theta)$$

- find the maximum likelihood estimate (MLE) of θ, the maximum a posteriori (MAP) estimate, or be Bayesian
- common mixture models used in semi-supervised learning:
  - Mixture of Gaussian distributions (GMM) – image classification
  - Mixture of multinomial distributions (Naïve Bayes) – text categorization
  - Hidden Markov Models (HMM) – speech recognition
- Learning via the Expectation-Maximization (EM) algorithm (Baum-Welch)

Gaussian Mixture Models

Binary classification with GMM using MLE.

- with only labeled data
  - $\log p(X_l, Y_l|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(x_i|y_i,\theta)$
  - MLE for θ trivial (sample mean and covariance)

- with both labeled and unlabeled data

$$\log p(X_l, Y_l, X_u|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(x_i|y_i,\theta) + \sum_{i=l+1}^{l+u} \log \left(\sum_{y=1}^{2} p(y|\theta)p(x_i|y,\theta) \right)$$

  - MLE harder (hidden variables): EM

EM for Gaussian Mixture Models

1. Start from MLE $\theta = \{w, \mu, \Sigma\}_{1:2}$ on (X_l, Y_l),
   - w_c=proportion of class c
   - μ_c=sample mean of class c
   - Σ_c=sample cov of class c

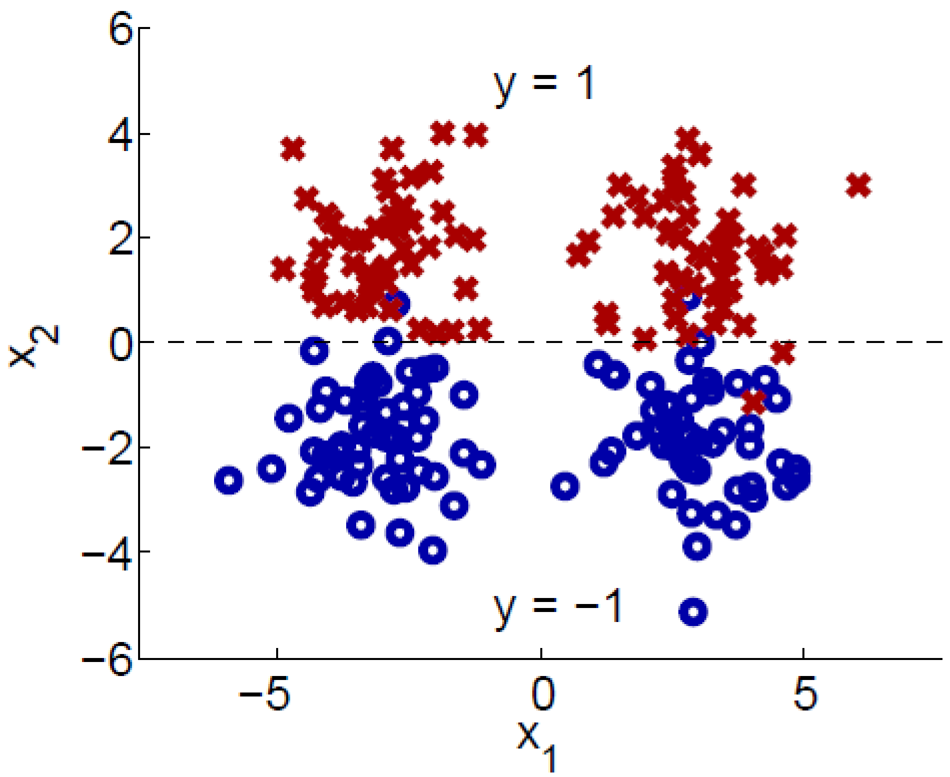
   repeat:

2. The E-step: compute the expected label $p(y|x, \theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)}$ for all $x \in X_u$
   - label $p(y = 1|x, \theta)$-fraction of x with class 1
   - label $p(y = 2|x, \theta)$-fraction of x with class 2

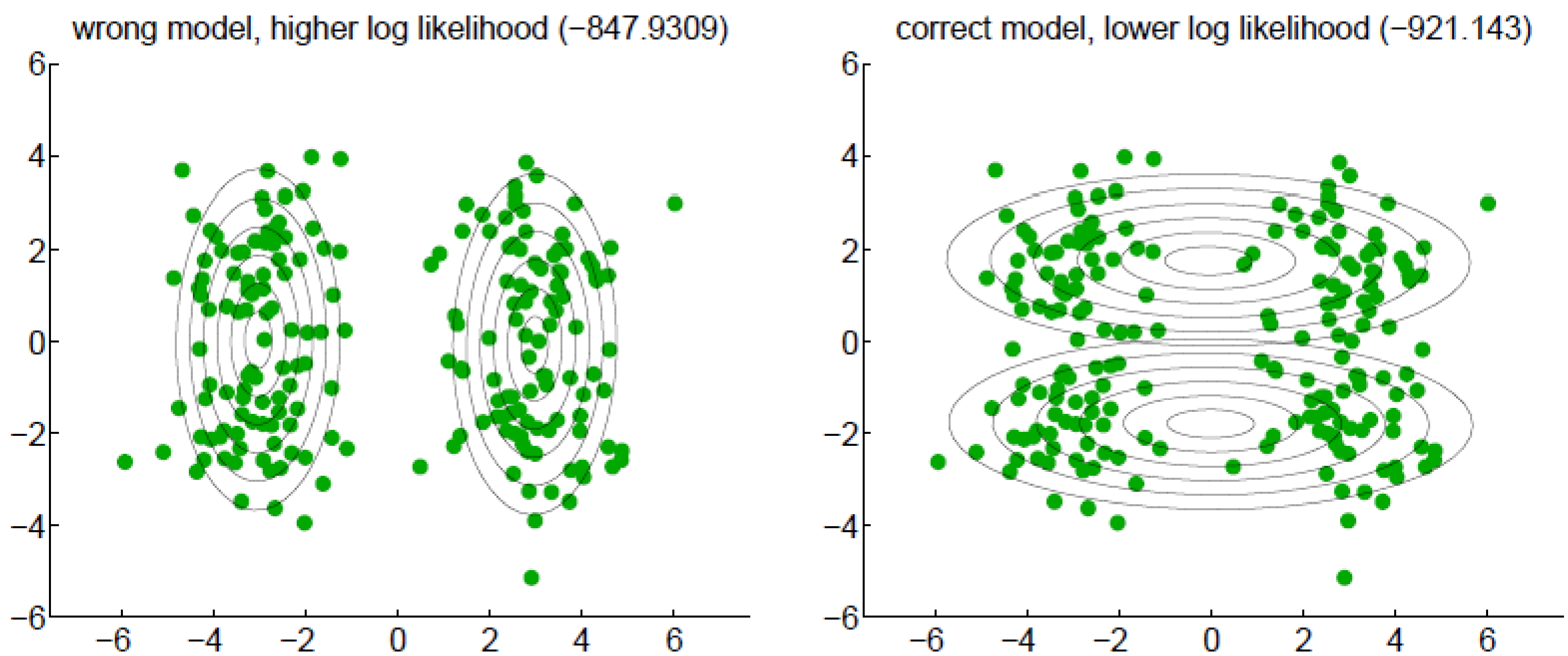
3. The M-step: update MLE θ with (now labeled) X_u

Assumption for GMMs

- **Assumption**: the data actually comes from the mixture model, where the number of components, prior $p(y)$, and conditional $p(\mathbf{x}|y)$ are all correct.
- When the assumption is wrong:



Assumption for GMMs



Assumption for GMMs

Heuristics to lessen the danger

- Carefully construct the generative model, e.g., multiple Gaussian distributions per class

- Down-weight the unlabeled data ($\lambda < 1$)

$$\log p(X_l, Y_l, X_u|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(x_i|y_i, \theta) + \lambda \sum_{i=l+1}^{l+u} \log \left(\sum_{y=1}^{2} p(y|\theta)p(x_i|y, \theta)\right)$$

Other issues: Identifiability, EM local optima

Related: Cluster and Label

Input: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l), \mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}$,
a clustering algorithm $\mathcal{A}$, a supervised learning algorithm $\mathcal{L}$

1. Cluster $\mathbf{x}_1, \ldots, \mathbf{x}_{l+u}$ using $\mathcal{A}$.
2. For each cluster, let S be the labeled instances in it:
3. Learn a supervised predictor from S: $f_S = \mathcal{L}(S)$.
4. Apply f_S to all unlabeled instances in this cluster.

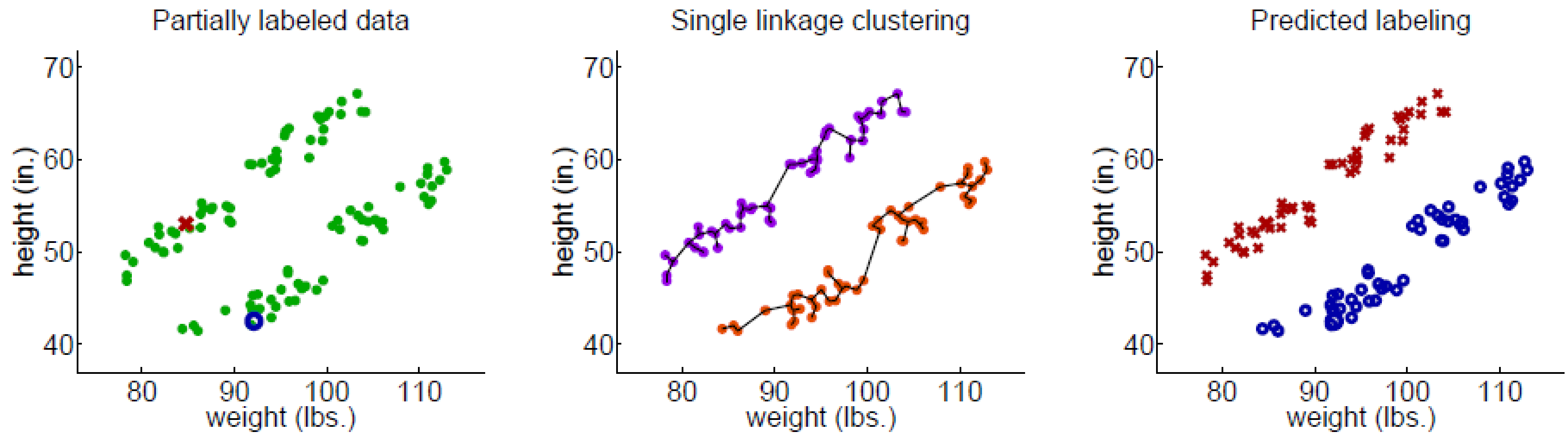
Output: labels on unlabeled data $y_{l+1}, \ldots, y_{l+u}$.

But again: **SSL sensitive to assumptions**—in this case, that the clusters coincide with decision boundaries. If this assumption is incorrect, the results can be poor.

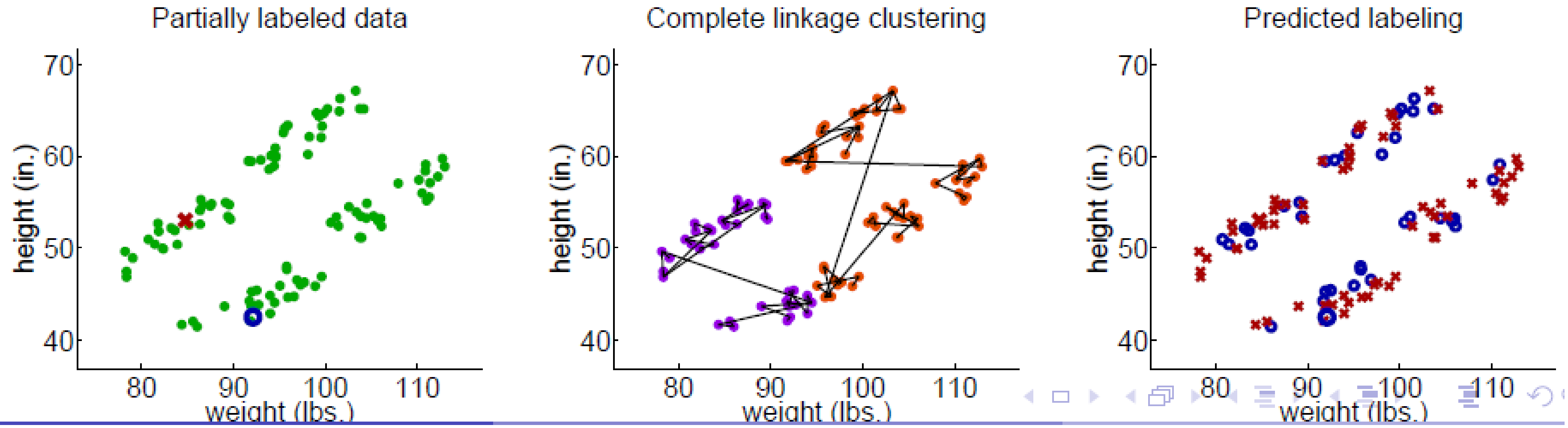
Cluster-and-label: now it works, now it doesn't

Example: $\mathcal{A}$=Hierarchical Clustering, $\mathcal{L}$=majority vote.

single linkage



complete linkage



Some SSL Algorithms

- Generative methods – assume a model for p(x,y) and maximize joint likelihood

  Mixture models

- Multi-view methods – multiple independent learners that agree on prediction for unlabeled data

  Co-training

- Graph-based methods – assume the target function p(y|x) is smooth wrt a graph or manifold

  Graph/Manifold Regularization

Two views of an Instance

Example: named entity classification `Person` (Mr. Washington) or `Location` (Washington State)

instance 1: ... headquartered in (Washington State) ...
instance 2: ... (Mr. Washington), the vice president of ...

- a named entity has two views (subset of features) $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]$
- the words of the entity is $\mathbf{x}^{(1)}$
- the context is $\mathbf{x}^{(2)}$

Two views of an Instance

instance 1:	. . . headquartered in (Washington State)^{L} . . .
instance 2:	. . . (Mr. Washington)^{P}, the vice president of . . .
test:	. . . (Robert Jordan), a partner at . . .
test:	. . . flew to (China) . . .

Two views of an Instance

With more unlabeled data

instance 1:	. . . headquartered in (Washington State)^L . . .
instance 2:	. . . (Mr. Washington)^P, the vice president of . . .
instance 3:	. . . headquartered in (Kazakhstan) . . .
instance 4:	. . . flew to (Kazakhstan) . . .
instance 5:	. . . (Mr. Smith), a partner at Steptoe & Johnson . . .
test:	. . . (Robert Jordan), a partner at . . .
test:	. . . flew to (China) . . .

Co-training Algorithm

Blum & Mitchell'98

Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$
each instance has two views $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}]$,
and a learning speed k.

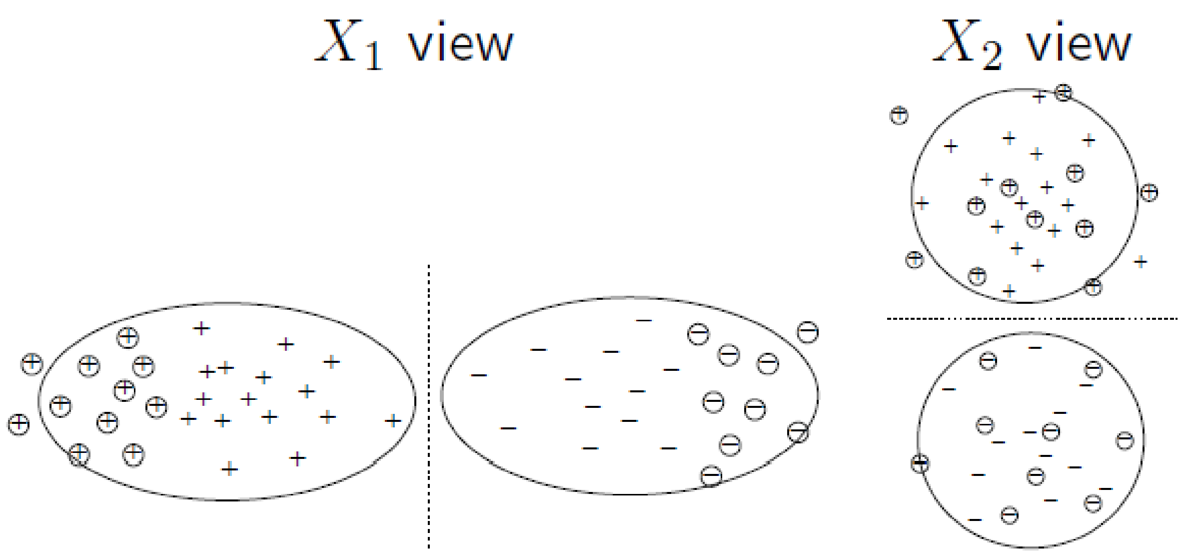
1. let $L_1 = L_2 = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$.
2. Repeat until unlabeled data is used up:
3. Train view-1 $f^{(1)}$ from L_1, view-2 $f^{(2)}$ from L_2.
4. Classify unlabeled data with $f^{(1)}$ and $f^{(2)}$ separately.
5. Add $f^{(1)}$'s top k most-confident predictions $(\mathbf{x}, f^{(1)}(\mathbf{x}))$ to L_2.
   Add $f^{(2)}$'s top k most-confident predictions $(\mathbf{x}, f^{(2)}(\mathbf{x}))$ to L_1.
   Remove these from the unlabeled data.

Like self-training, but with two classifiers teaching each other.

Co-training

Assumptions

- feature split $x = [x^{(1)}; x^{(2)}]$ exists
- $x^{(1)}$ or $x^{(2)}$ alone is sufficient to train a good classifier
- $x^{(1)}$ and $x^{(2)}$ are conditionally independent given the class



Multi-view learning

Extends co-training.

- Loss Function: $c(\mathbf{x}, y, f(\mathbf{x})) \in [0, \infty)$. For example,
  - squared loss $c(\mathbf{x}, y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$
  - 0/1 loss $c(\mathbf{x}, y, f(\mathbf{x})) = 1$ if $y \neq f(\mathbf{x})$, and 0 otherwise.
- Empirical risk: $\hat{R}(f) = \frac{1}{l} \sum_{i=1}^{l} c(\mathbf{x}_i, y_i, f(\mathbf{x}_i))$
- Regularizer: $\Omega(f)$, e.g., $\|f\|^2$
- Regularized Risk Minimization $f^* = \text{argmin}_{f \in \mathcal{F}} \hat{R}(f) + \lambda \Omega(f)$

Multi-view learning

A special regularizer $\Omega(f)$ defined on unlabeled data, to encourage agreement among multiple learners:

Each of the k learners is good

$$\underset{f_1,\ldots,f_k}{\operatorname{argmin}} \quad \sum_{v=1}^{k}\left(\sum_{i=1}^{l} c(\mathbf{x}_i, y_i, f_v(\mathbf{x}_i)) + \lambda_1 \Omega_{SL}(f_v)\right) + \lambda_2 \sum_{u,v=1}^{k} \sum_{i=l+1}^{l+u} c(\mathbf{x}_i, f_u(\mathbf{x}_i), f_v(\mathbf{x}_i))$$

The k learners agree on unlabeled data

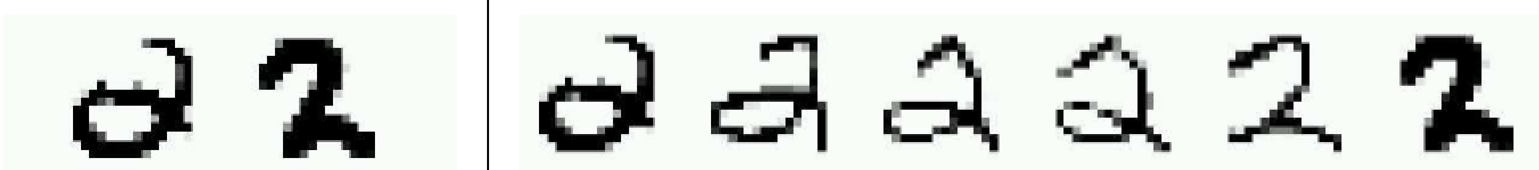
Some SSL Algorithms

- Generative methods – assume a model for p(x,y) and maximize joint likelihood
  - Mixture models
- Multi-view methods – multiple independent learners that agree on prediction for unlabeled data
  - Co-training
- Graph-based methods – assume the target function p(y|x) is smooth wrt a graph or manifold
  - Graph/Manifold Regularization

Graph Regularization

Assumption: Similar unlabeled data have similar labels.

Handwritten digits recognition with pixel-wise Euclidean distance



not similar

'indirectly' similar
with stepping stones

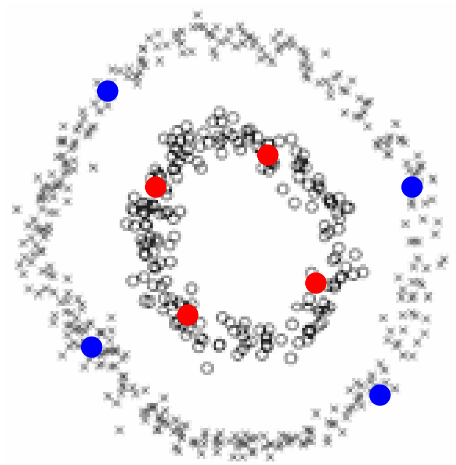
Graph Regularization

Similarity Graphs: Model local neighborhood relations between data points

- Nodes: $X_l \cup X_u$

- Edges: similarity weights computed from features, e.g.,
  - k-nearest-neighbor graph
  - fully connected graph, weight decays with distance

    $w_{ij} = \exp\left(-\|x_i - x_j\|^2/\sigma^2\right)$
  - ϵ-radius graph



Graph Regularization

Graph Prior: $$p(f) \propto e^{-\sum_{i,j} w_{ij}(f_i - f_j)^2}$$

If data points i and j are similar (i.e. weight w_{ij} is large), then their labels are similar $f_i = f_j$

$$\min_f \underbrace{\sum_{i \in l} (y_i - f_i)^2}_{\text{Loss on labeled data (mean square,0-1)}} + \lambda \underbrace{\sum_{i,j \in l,u} w_{ij}(f_i - f_j)^2}_{\text{Graph based smoothness prior on labeled and unlabeled data}}$$

Loss on labeled data
(mean square,0-1)

Graph based smoothness prior
on labeled and unlabeled data

Graph Regularization

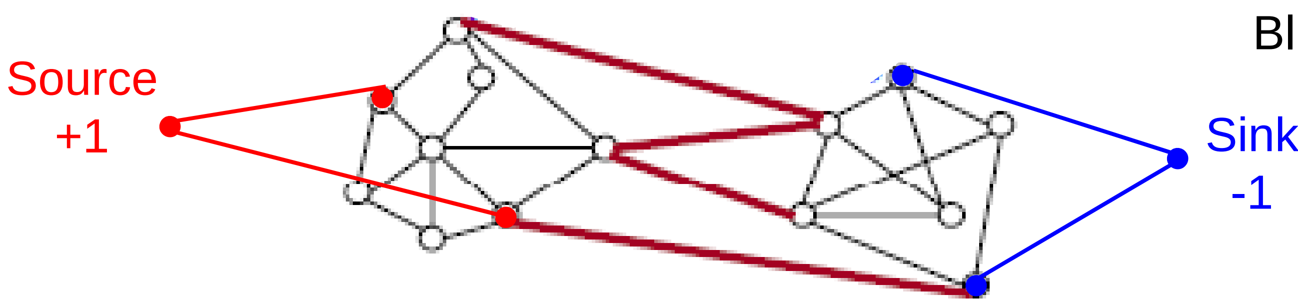
$$\min_f \sum_{i \in l} (y_i - f_i)^2 + \lambda \sum_{i,j \in l,u} w_{ij} (f_i - f_j)^2$$

Loss on labeled data

Graph based smoothness prior on labeled and unlabeled data

From previous lecture, recall the second term is simply the min-cut objective.

If binary label, can be solved by min-cut on a modified graph - add source and sink nodes with large weight to labeled examples.



Blum & Chawla'01

Semi-Supervised Learning

- Generative methods – Mixture models
- Multi-view methods – Co-training
- Graph-based methods – Manifold Regularization
- Semi-Supervised SVMs – assume unlabeled data from different classes have large margin
- Many other methods

SSL algorithms can use unlabeled data to help improve prediction accuracy **if data satisfies appropriate assumptions**