Semi-supervised Learning

Lorena Álvarez Pérez

Machine Learning Group (MLG)

November 15, 2017

Index

1. Motivation / Introduction
2. Semi-supervised learning
3. Self-training
4. Co-training and multi-view learners
5. Generative models
6. Semi-supervised learning with graphs
7. Transductive Support Vector Machines
8. Semi-supervised learning in Nature
9. Conclusions

1. Motivation

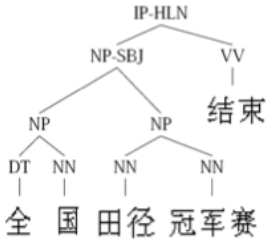
- Supervised Learning models require labeled data
- Learning a reliable model usually requires plenty of labeled data
- Labeled data: expensive and scarce

Examples of hard-to-get-labels

1) Speech Analysis: Switchboard dataset
   - Telephone transcription dataset
   - 400 hours annotation time for each hour of conversation

     Film ➔ f ih_n uh_gl_n m

2) Natural Language Parsing: Penn Chinese Treebank
   - 2 years for 4000 sentences



"The National Track and Field Champions has finished"

1. Motivation. Semi-supervised Learning

- Unlabeled data: abundant and free/cheap
  - E.g., webpage classification: easy to get unlabeled webpages

Semi-supervised (SSL) learning: Devising ways of utilizing unlabeled data with labeled data to learn better models

1. Motivation. Unlabeled data

How can unlabeled data ever help?

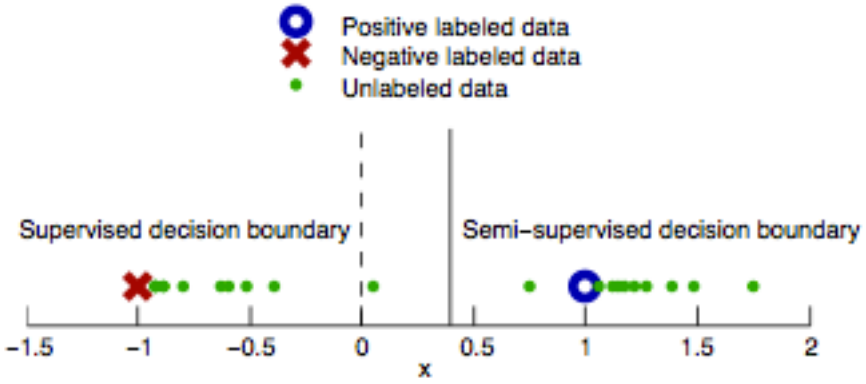


*Figure taken from: X. Zhu, "Tutorial on Semi-Supervised Learning", Theory and Practice of Computational Learning, 2009

- Assuming each class is a coherent distribution (e.g. Gaussian)
- With and without unlabeled data: decision boundary shift
  - Unlabeled data can give a better sense of the class separation boundary

2. Semi-supervised learning: Formally

- General idea: Learning from both labeled and unlabeled data
- Semi-supervised Classification/Regression
  - Given: Labeled training data $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{L}$, unlabeled data $\mathcal{U} = \{\mathbf{x}_j, y_i\}_{j=L+1}^{L+U}$ (usually U>> L)
  - Goal: Learning a classifier f better than using labeled data alone
- Constrained clustering
  - Given: Unlabeled training data $\{\mathbf{x}_i\}_{i=1}^{N}$ and "supervised information", e.g., must-links, cannot-links, or both, and the **goal** is to better cluster than from unlabeled data alone.

We will mainly discuss semi-supervised classification!

2. Semi-supervised learning

Inductive vs. transductive learning

- Inductive learning:

  Given $\{\mathbf{x}_i, y_i\}_{i=1}^{L}, \{\mathbf{x_j}\}_{j=L+1}^{L+U}$, learn $f : \mathcal{X} \rightarrow \mathcal{Y}$ so that f is expected to be a good predictor on future data, beyond $\{\mathbf{x}_j\}_{j=L+1}^{U}$

- Transductive learning

  Given $\{\mathbf{x}_i, y_i\}_{i=1}^{L}, \{\mathbf{x_j}\}_{j=L+1}^{L+U}$, learn $f : \mathcal{X}^{L+U} \rightarrow \mathcal{U}^{L+U}$ so that f is expected to be a good predictor on the unlabeled data $\{\mathbf{x}_j\}_{j=L+1}^{U}$. Note f is defined only on the given training sample, and is not required to make predictions outside them.

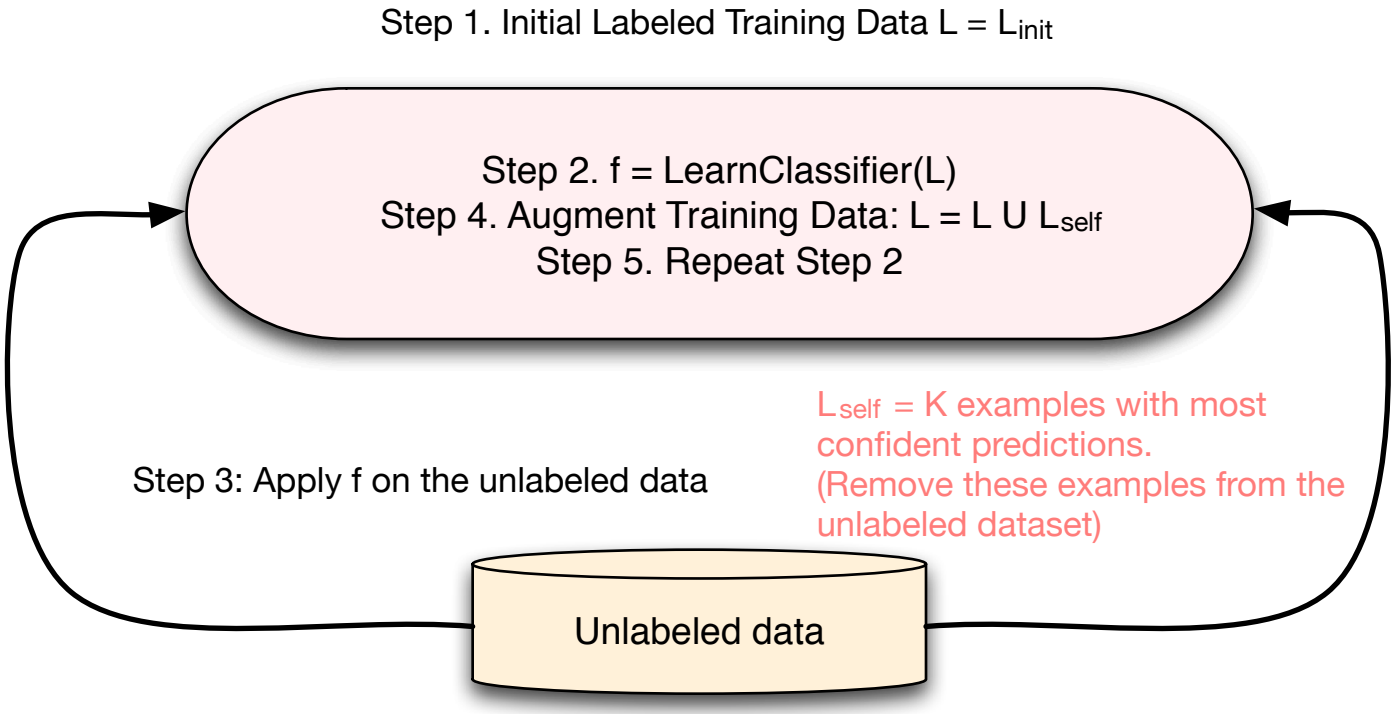
2. Semi-supervised learning

- **How many semi-supervised learning methods are there?**
  - Many. Some of them include:
    - Self-training
    - Co-training and multi-view learning
    - Generative mixture models
    - Graph-based methods
    - Transductive Support Vector Machines
    - …

We will study some of them!

3. Self-Training

- **Given**: Small amount of initial labeled training data
- **Idea**: Train, predict, re-train using your own (best) prediction, repeat



- Can be used in any supervised learner. Often works well in practice
- Caution: Prediction mistake can reinforce itself!

3. Self-Training: A Good Case

- **Base learner**: 1-NN classifier

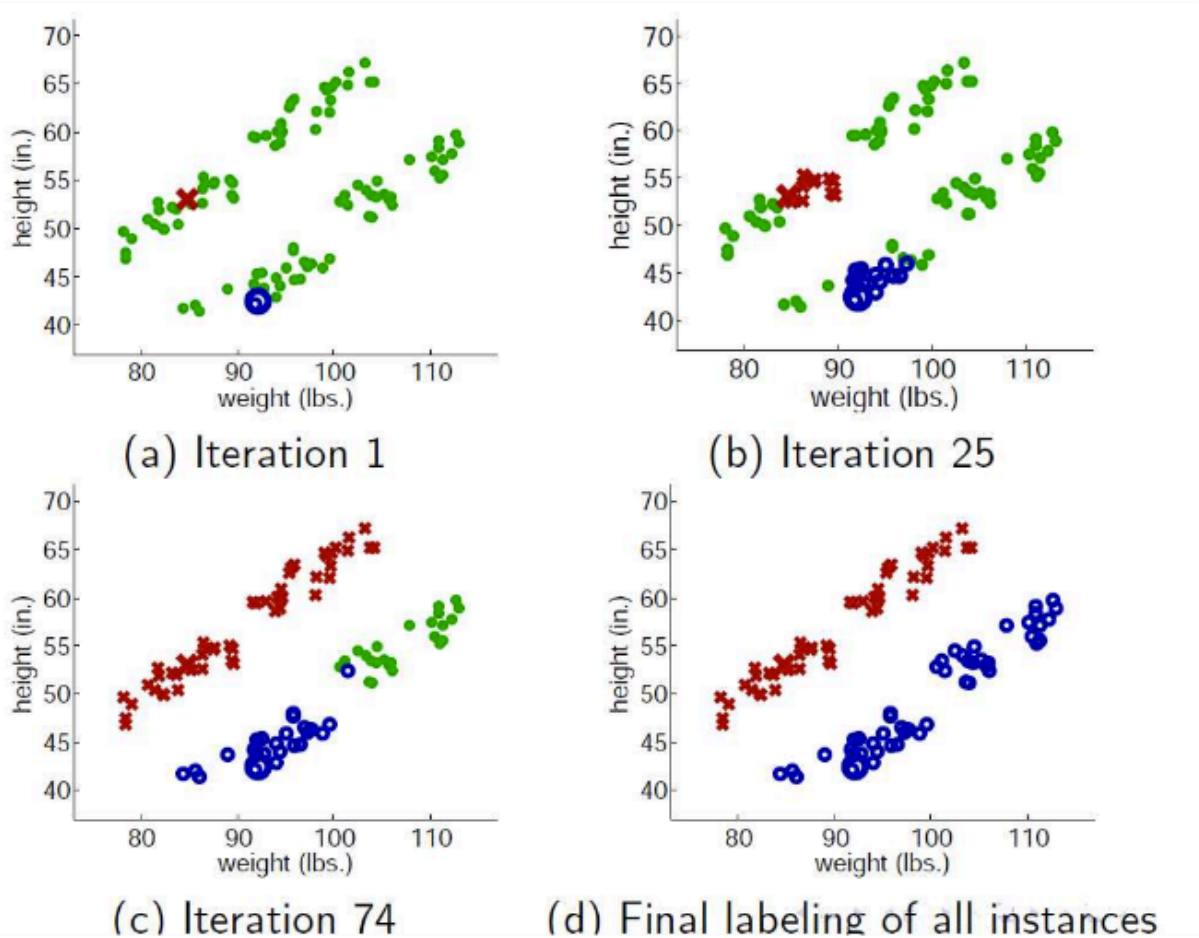


*Figure taken from: Piyush Rai, "Semi-supervised Learning", CS5350/6350: Machine Learning, November 2011

3. Self-Training: A Good Case

- **Base learner**: 1-NN classifier
- Things can go wrong if there are **outliers**. Mistakes get reinforced

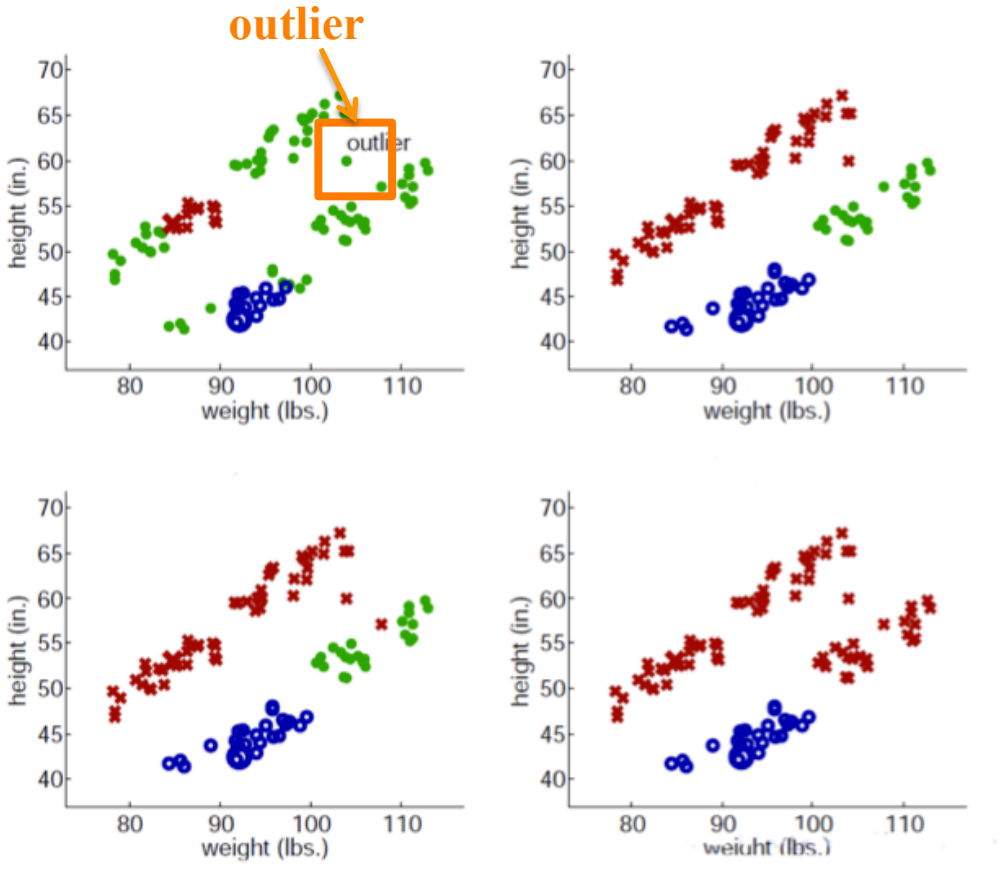


*Figure taken from: Piyush Rai, "Semi-supervised Learning", CS5350/6350: Machine Learning, November 2011

4. Co-training and Multi-view Learning

4.1 Co-training: Idea

- **Given**: Labeled data $\{\mathbf{x}_i, y_i\}_{i=1}^{L}$, unlabeled data $\{\mathbf{x}_j\}_{j=L+1}^{L+U}$
- Each sample has 2 views: $[\mathbf{x}^{(1)}\ \ \mathbf{x}^{(2)}]$
- How do we get different views?
  - Naturally available (different types of features for the same object/ pattern)
    - Webpages: view 1 from page text; view 2 from social tags
    - Images: view 1 from features pixel; view 2 from Fourier coefficients
  - … or by splitting the original features into two groups

4. Co-training and Multiview-Learning

4.1 Co-training: Idea

- Assumption: Given sufficient data, each view is good enough to learn from
- Co-training: Utilize both views to learn better with fewer labeled examples
- **Idea:** Each view teaching (training) the other view
- Technical Condition: Views should be conditionally independent given the class
  - Intuitively, we don't want redundancy between the views

4. Co-training and Multiview-Learning

4.1 Co-training: Algorithm

- Given labeled data L and unlabeled data U
- Create two labeled datasets L_1 and L_2 from L using views 1 and 2
- Learn classifier $f^{(1)}$ using L_1 and classifier $f^{(2)}$ using L_2
- Apply $f^{(1)}$ and $f^{(2)}$ on labeled data pool U to predict labels
  - Predictions are made only using their own set (view) of features
- Add K most confident predictions $((\mathbf{x}, f^{(1)}(\mathbf{x}))$ of $f^{(1)}$ to L_2
- Add K most confident predictions $((\mathbf{x}, f^{(2)}(\mathbf{x}))$ of $f^{(2)}$ to L_1
- Remove theses examples from the unlabeled data U
- Re-train $f^{(1)}$ using L_1, $f^{(2)}$ using L_2
- Like self-training but two classifiers teaching each other

4. Co-training and Multiview-Learning

4.1 Co-training: Variants

- co-EM (Nigam&Ghani [2000])
  - Hybrid algorithm of expectation-maximization (EM) and co-training
  - co-EM tries to divide the instance space into two conditional independent views and train two EM classifiers based on these two views
  - Unlike co-training, co-EM uses all the unlabeled data every time, instead of incrementally selecting some confident predictions to update the training set
- Goldman&Zhou [2000]
  - Co-training and co-EM suffer from the conditional independent assumption
  - They use two different algorithms in the paradigm of co-training without splitting the attribute set
- …

4. Co-training and Multi-view Learning

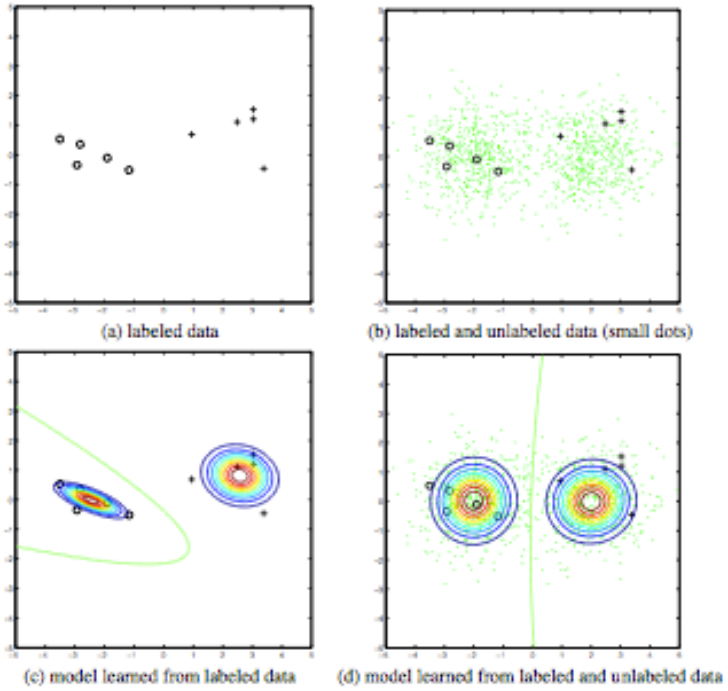
4.2 Multi-view Learning

- A general class of algorithms for semi-supervised learning
- Based on using multiple views (feature representations) of the data
- General Idea: Train multiple classifier, each using a different view
- Modus Operandi: Multiple classifiers must agree on the unlabeled data
- How might it help learn better?
  - Learning is essentially searching for the best classifier
  - By enforcing agreement among classifiers, we are reducing the search space
    - Hope is that the best classifier can be found easily with little labeled data
- For test data, these multiple classifiers can be combined
  - E.g., voting, consensus, etc.

5. Generative models

5.1 Introduction

- It assumes a model $p(x, y) = p(y)p(x|y)$, where $p(x|y)$ is an identifiable mixture distribution (e.g., Gaussian mixture model)
- With large amount of unlabeled data, the mixture components can be identified (ideally, only one labeled sample is needed)



Use unlabeled data help parameter estimation!

*Figure taken from: X. Zhu, "Tutorial on Semi-Supervised Learning", Theory and Practice of Computational Learning, 2009

- They are different because they maximize different quantities

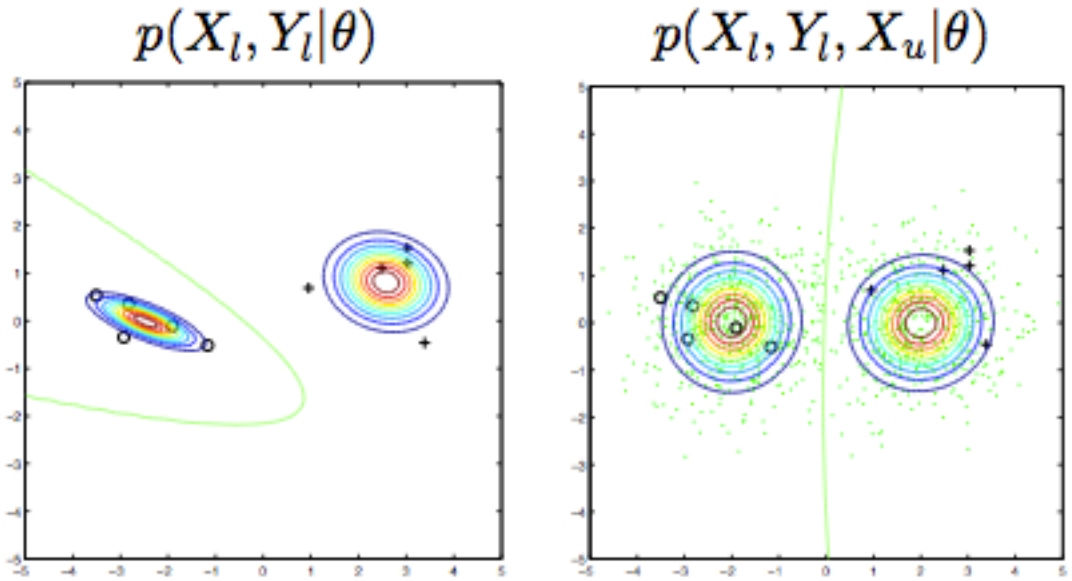


*Figure taken from: X. Zhu, "Tutorial on Semi-Supervised Learning", Theory and Practice of Computational Learning, 2009

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 $y_{l+1}, \ldots, y_{l+u}$

- Assumption: Clusters coincide with decision boundaries
  - Poor results if this assumption is wrong

- **Given:** Labeled data $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{L}$, unlabeled data $\mathcal{U} = \{\mathbf{x}_j, y_i\}_{j=L+1}^{L+U}$
- Expectation-Maximization based SSL:
  - Train an initial model using just $\mathcal{L}$

$$\widehat{\theta} = \text{TrainModel}(\mathcal{L})$$

  - Use this model to "guess" the label of each $\mathbf{x_j} \in \mathcal{U}$ (compute expected label). Assuming binary labels (+1/-1), we can compute:

$$\text{E}[y_j] = +1 \times P(y_j = +1|\widehat{\theta}, \mathbf{x_j}) + (-1) \times P(y_j = -1|\widehat{\theta}, \mathbf{x_j})$$

  - Re-train the model using $\mathcal{L} + \mathcal{U}$ with its guessed labels
  - Use the new model $\widehat{\theta}$ to refine the guesses of the labels $\text{E}[y_j]$ of $\mathcal{U}$
  - Repeat until converged
- A general scheme can be used with any probabilistic learning model
  - E.g. naïve Bayes, logistic regression, linear regression, etc.

- This kind of algorithm tends to converge to a local maximum, specially in classification problems in which there are few labeled samples
- Some possible solutions:
  - Assign weights for the unlabeled samples lower than for labeled samples
  - Smart choice of starting point by active learning

6. Semi-supervised Learning with graphs

6.1 Introduction

- Graph based approaches exploit the property of label smoothness
- Idea: Represent each example (labeled/unlabeled) as vertices of some graph
  - The labels should vary smoothly along the graph
    - Nearby vertices should have similar labels

Example: Handwritten digits recognition with pixel-wise Euclidean distance

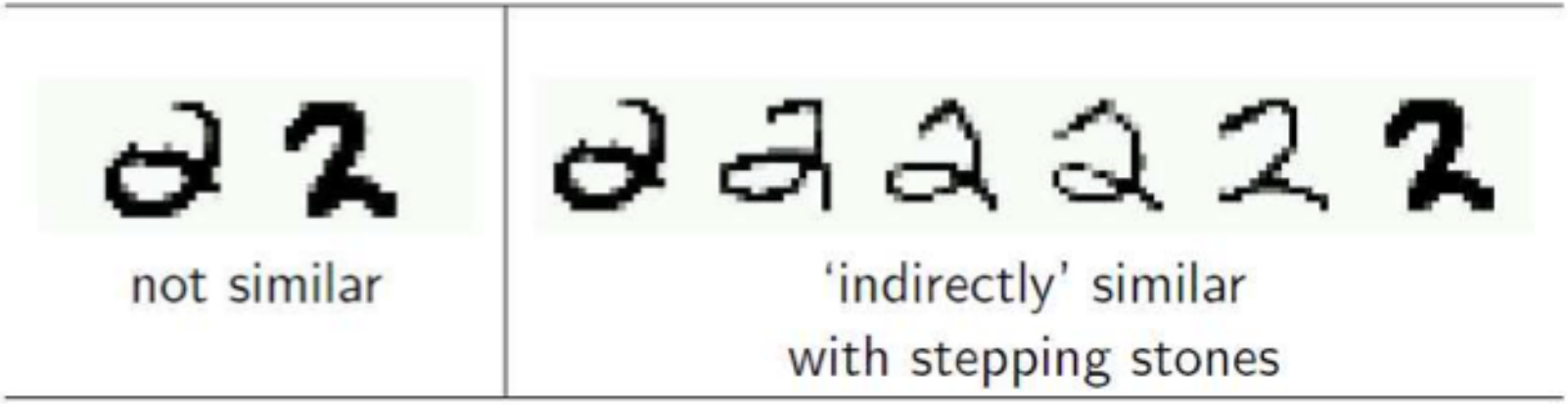


*Figure taken from: Piyush Rai, "Semi-supervised Learning", CS5350/6350: Machine Learning, November 2011

- Nodes $X_l \cup X_u$
- Edges: similarity weights computed from features, e.g.,
  - K-nearest neighbor graph, unweighted (0, 1 weights)
  - Fully connected graph, weight decays with distance

$$w = \exp\left(-\frac{||x_i - x_j||^2}{\sigma^2}\right)$$

- Assumption: Instances connected by heavy edge tend to have the same label

(a) Before SSL

(b) After SSL

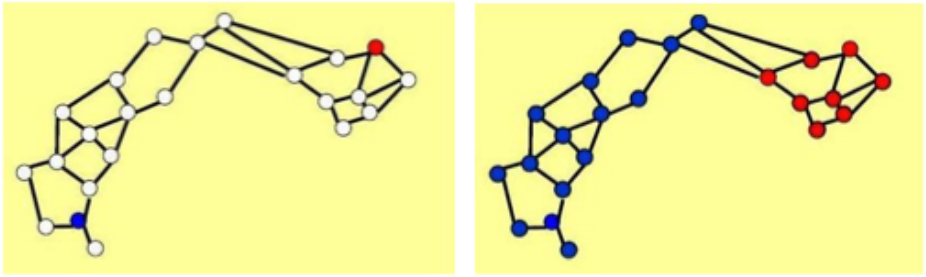


*Figure taken from: *M. Farouk et. al.*, "Semi-supervised Learning", Handbook on Neural Information Processing (Chapter 7), Springer, 2013.

- Assume the predictions on the entire data $\mathcal{L} + \mathcal{U}$ to be defined by function f
- Graph regularization assumes that the function f is smooth
  - Similar examples i and j should have similar predictions f_i and f_j
- Graph regularization optimizes the following objective

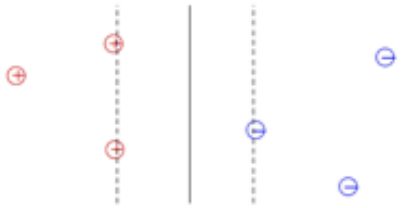
$$\min_f \sum_{i \in \mathcal{L}} (y_i - f_i)^2 + \lambda \sum_{i,j \in \mathcal{L},\mathcal{U}} w_{ij} (f_i - f_j)^2$$

- First part is minimizing the loss on labeled data, second part ensures smoothness of labels of labeled and unlabeled data
  - Minimization makes f_i and f_j to be very similar if w_{ij} is large
- λ is a trade-off parameter
- Several variants and ways to solve the above problem

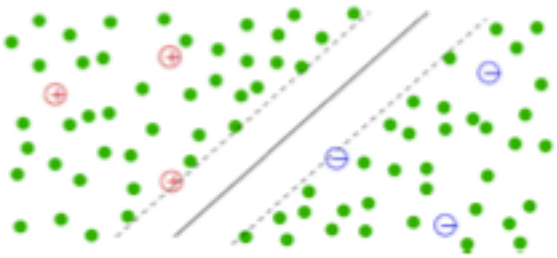
7. Semi-supervised Support Vector Machines

7.1 Idea

- In standard SVMs



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



- Assumption: Unlabeled data from different classes are separated with large margin

- The S3VM objective function
  - To incorporate unlabeled points:
    - assign putative labels $\text{sign}(f(\mathbf{x}))$ to $\mathbf{x} \in \mathcal{U}$
    - the hinge loss on unlabeled points becomes

$$(1 - \text{sign}(f(\mathbf{x}_i))f(\mathbf{x}_i))_+ = (1 - |(f(\mathbf{x}_i|)_+$$

where $(z)_+ = \max(z, 0)$

- S3VM objective:

$$\min_f \sum_{i=1}^{L}(1 - y_i f(\mathbf{x}_i)) + \lambda_1 ||h||^2_{\mathcal{H}_K} + \lambda_2 \sum_{j=L+1}^{L+U} (1 - |(f(\mathbf{x}_i|)_+$$

7. Semi-supervised Support Vector Machines
7.2 Algorithm

- The S3VM objective function
  - To incorporate unlabeled points:
    - assign putative labels $\text{sign}(f(\mathbf{x}))$ to $\mathbf{x} \in \mathcal{U}$
    - the hinge loss on unlabeled points becomes

$$(1 - \text{sign}(f(\mathbf{x}_i))f(\mathbf{x}_i))_+ = (1 - |(f(\mathbf{x}_i|)_+$$

where $(z)_+ = \max(z, 0)$

- S3VM objective:

$$\min_f \sum_{i=1}^{L}(1 - y_i f(\mathbf{x}_i)) + \lambda_1 ||h||^2_{\mathcal{H}_K} + \lambda_2 \sum_{j=L+1}^{L+U} (1 - |(f(\mathbf{x}_i|)_+$$

7. Semi-supervised Support Vector Machines
7.2 Algorithm

- The hat loss on unlabeled data

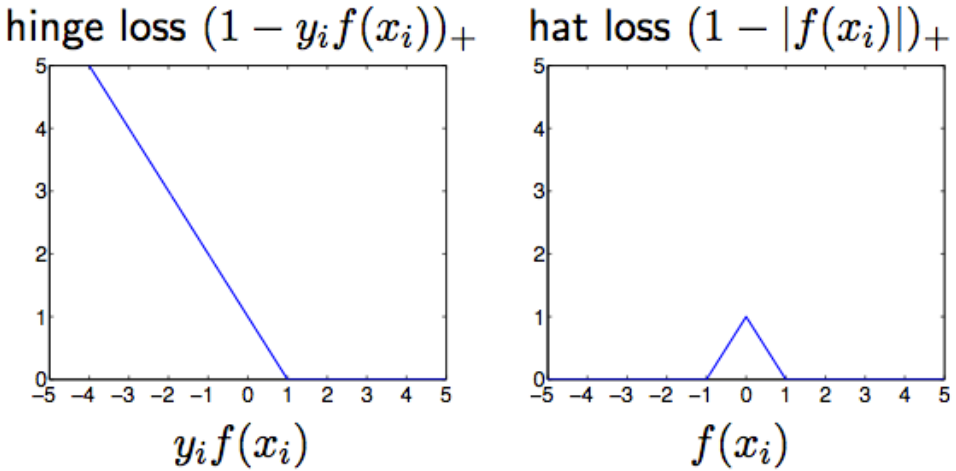


*Figure taken from: X. Zhu, "Tutorial on Semi-Supervised Learning", Theory and Practice of Computational Learning, 2009

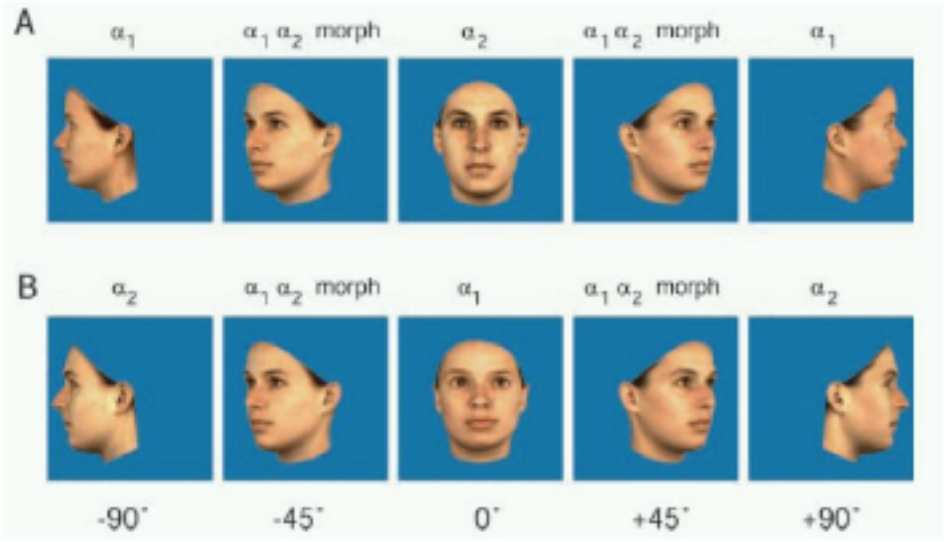
Prefers $f(x) \geq 1$ or $f(x) \leq -1$, i.e., unlabeled instance away from decision boundary $f(x) = 0$

8. Semi-supervised learning in Nature

- Learning exists long before machine learning
  - Do humans perform semi-supervised learning?
  - Yes, it seems. We discuss two human experiments.
    - Visual recognition with temporal association
    - Infant word-object mapping

8. Semi-supervised learning in Nature

- Visual recognition with temporal association
  - A face from two angles are very different, but we can easily associate it
  - The image sequence (unlabeled data) might be the glue
  - Artificial wrong sequences (person A's profile morphs to B's frontal) damage people's ability to match test profile and frontal images



8. Semi-supervised learning in Nature

- Infant word-object mapping
  - 17-month infants listen to a word, see an object
  - Measure their ability to associate the word and object
    - If the word heard many times before (without seeing the object; unlabeled data), association is stronger
    - If the word not heard before, association is weaker
  - Similar to cluster-and-label

9. Conclusions

- **Can we really learn anything from unlabeled data?**
  - Yes, we can. Unlabeled data can help if the model assumptions are appropriate
  - Spend reasonable amount of effort to design good models/features/kernels/ similarity functions for semi-supervised learning.
- **Does unlabeled data always help?**
  - No, there's no free lunch. Bad matching of problem with model assumption can lead to degradation in classifier performance.

9. Conclusions

- **Which method should I use / is the best?**
  - There is no direct answer to this question.
  - SSL methods make strong model assumptions because the labeled data scarce and there is no guarantee that unlabeled data will always help

Method	Assumption
Co-training	Conditionally independent Each view is good enough to learn from
Mixture model	Identifiable mixture distribution
Graph methods	Labels smooth on graph
Transductive SVM	Low density region between classes

9. Conclusions

- **Which method should I use / is the best?**
  - Do the classes produce well clustered data?
    - EM with generative models
  - Do the features natural split into two sets
    - Co-training
  - Is that true that two points with similar features tend to be in the same class?
    - Graph-based methods
  - Already using SVM?
    - Transductive SVM is a natural extension
  - In all cases
    - Self-training is a practical wrapper method