

DDA4210 Advanced Machine Learning

Lecture 05-II Semi-Supervised Learning

Jicong Fan

School of Data Science, CUHK-Shenzhen

February 27

Overview

- 1 Introduction
- 2 Self-training algorithm
- 3 Graph based SSL methods

Slides Courtesy: Jerry Zhu

- 1 Introduction
- 2 Self-training algorithm
- 3 Graph based SSL methods

Three Types of Learning

- Supervised learning (SL)
 - Classification
 - Regression
- Unsupervised learning (USL)
 - Clustering
 - Dimensionality reduction
 - Probability distribution estimation
 - Generative models
- Semi-supervised learning (SSL)

Why Semi-Supervised Learning?

- Labeled data are rare or expensive
 - Human annotation is boring
 - Labels may require experts
 - Labels may require special devices or money

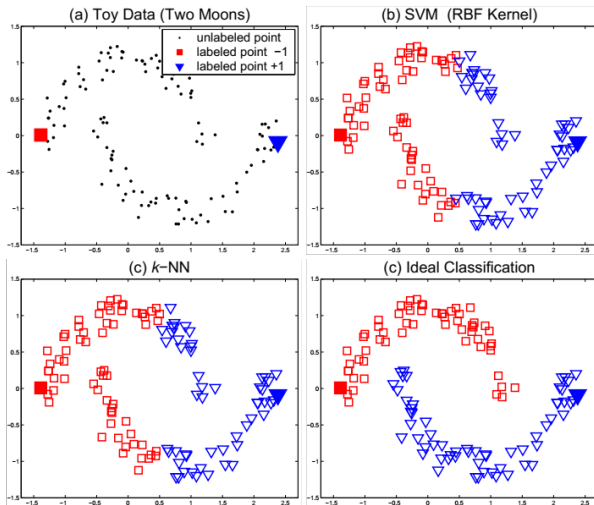
Why Semi-Supervised Learning?

- Labeled data are rare or expensive
 - Human annotation is boring
 - Labels may require experts
 - Labels may require special devices or money
- Unlabeled data are prevalent and cheap
- Unlabeled data are helpful
 - Using both labeled and unlabeled data to build better learners, than using each one alone.

Why Semi-Supervised Learning?

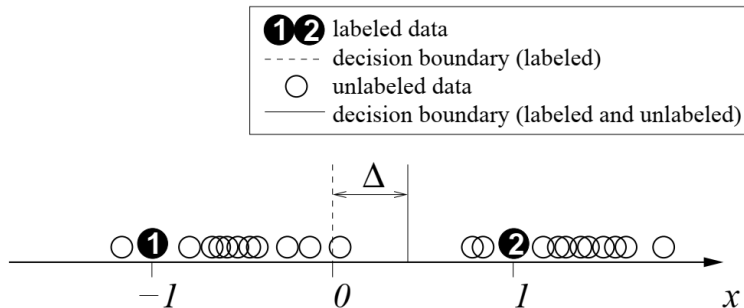
Classification on the two moons pattern [Zhou et al. 04]:

(a) two labeled points; (b) SVM with a RBF kernel; (c) k -NN with $k = 1$.



- Input samples \mathbf{x} , label \mathbf{y}
- Learner $f : \mathcal{X} \mapsto \mathcal{Y}$
- Labeled data $(X_l, Y_l) = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_l, \mathbf{y}_l)\}$
- Unlabeled data $X_u = \{\mathbf{x}_{l+1}, \dots, \mathbf{x}_N\}$, available during training
- Usually, $l \ll N$
- Test data $X_{\text{test}} = \{\mathbf{x}_{N+1}, \dots\}$, not available during training

How Can Unlabeled Data Help?



- Assuming each class is a coherent group (e.g. Gaussian)
- With and without unlabeled data: decision boundary shift
- This is only one of many ways to use unlabeled data.

- **Self-training algorithm**
- **Graph based algorithms**
- **Graph convolutional network based SSL** ([next lecture](#))
- Other algorithms

- 1 Introduction
- 2 Self-training algorithm**
- 3 Graph based SSL methods

Self-Training Algorithm

- Assumption: One's own high confidence predictions are correct.
- Self-training algorithm
 1. Train f from (X_l, Y_l)
 2. Predict on $\mathbf{x} \in X_u$
 3. Add $(\mathbf{x}, f(\mathbf{x}))$ to labeled data
 4. Repeat

- Some variations

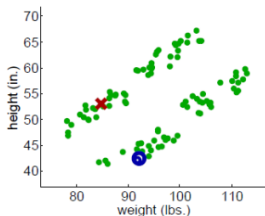
- Add a few most confident $(\mathbf{x}, f(\mathbf{x}))$ to labeled data
- Add all $(\mathbf{x}, f(\mathbf{x}))$ to labeled data
- Add all $(\mathbf{x}, f(\mathbf{x}))$ to labeled data, but with different weights according to the confidence

Self-Training Algorithm: Propagating 1-NN

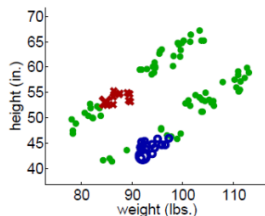
1. Classify \mathbf{x} with 1-NN
2. Add $(\mathbf{x}, f(\mathbf{x}))$ to labeled data, and repeat

Self-Training Algorithm: Propagating 1-NN

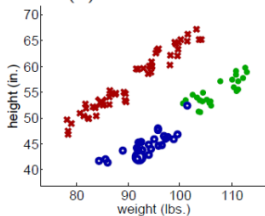
1. Classify \mathbf{x} with 1-NN
2. Add $(\mathbf{x}, f(\mathbf{x}))$ to labeled data, and repeat



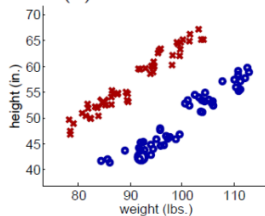
(a) Iteration 1



(b) Iteration 25



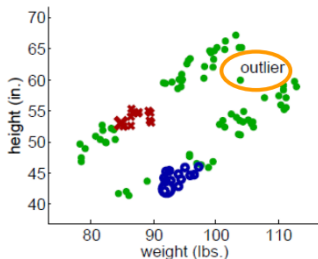
(c) Iteration 74



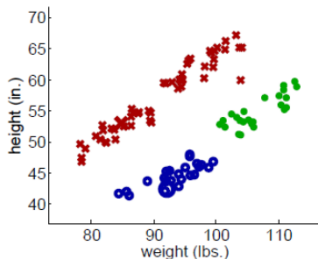
(d) Final labeling of all instances

Self-Training Algorithm: Propagating 1-NN

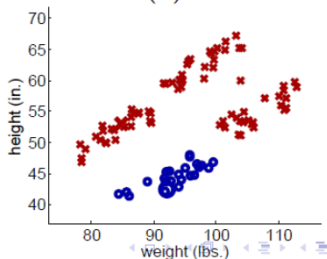
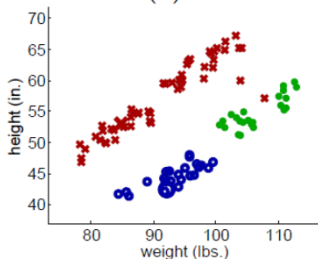
It is sensitive to outlier!



(a)



(b)



Advantage and Disadvantage of Self-Training

- Advantage

- The simplest semi-supervised learning method.
- A wrapper method, applies to existing (complex) classifiers.
- Often used in real tasks like natural language processing.

Advantage and Disadvantage of Self-Training

- Advantage

- The simplest semi-supervised learning method.
- A wrapper method, applies to existing (complex) classifiers.
- Often used in real tasks like natural language processing.

- Disadvantage

- Early mistakes could reinforce themselves

- 1 Introduction
- 2 Self-training algorithm
- 3 Graph based SSL methods**

Example 1

- Classify astronomy v.s. travel articles
 - Articles d_1 and d_2 are training data (labeled)
 - Classify articles d_3 and d_4 (test data)
 - Use similarity measured by content word overlap
- Case A: successful classification

	d_1	d_3	d_4	d_2
asteroid	•	•		
bright	•	•		
comet		•		
year				
zodiac				
.				
.				
airport				
bike				
camp			•	
yellowstone			•	•
zion				•

Example 1

- Classify astronomy v.s. travel articles
 - Articles d_1 and d_2 are training data (labeled)
 - Classify articles d_3 and d_4 (test data)
 - Use similarity measured by content word overlap
- Case B: failed classification (since there is no overlapping words!)

	d_1	d_3	d_4	d_2
asteroid	•			
bright	•			
comet				
year				
zodiac		•		
.				
.				
.				
airport			•	
bike			•	
camp				
yellowstone				•
zion				•



Example 1

- Case C: Take advantages of unlabeled data
 - d_5, d_6, d_7, d_8, d_9 are unlabeled articles
 - Labels “propagate” via similar unlabeled articles

	d_1	d_5	d_6	d_7	d_3	d_4	d_8	d_9	d_2
asteroid	•								
bright	•	•							
comet		•	•						
year			•	•					
zodiac				•	•				
.									
.									
.									
airport						•			
bike						•	•		
camp							•	•	
yellowstone								•	•
zion									•

Example 2

Handwritten digits recognition with pixel-wise Euclidean distance

 <p>not similar</p>	 <p>'indirectly' similar with stepping stones</p>
--	---

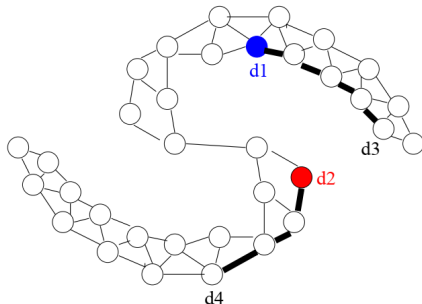
- **Assumption:** A graph is given on the labeled and unlabeled data. Instances connected by heavy edge tend to have the same label

- **Assumption:** A graph is given on the labeled and unlabeled data. Instances connected by heavy edge tend to have the same label

Question: Any other graph-based methods we have learnt?

Graph

- Nodes $X_I \cup X_U$
- Edges: similarity weights computed from features, e.g.,
 - k-nearest-neighbor graph, unweighted (0, 1 weights)
 - fully connected graph, weight decays with distance
$$w_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / (2\sigma^2))$$
- Want: implied similarity via all paths



Graph Regularization

- Regularized classifier
- Learn a classifier that minimize
 - Loss term + regularization
 - Example: regularized least squares, LASSO

Graph Regularization

- Regularized classifier
- Learn a classifier that minimize
 - Loss term + regularization
 - Example: regularized least squares, LASSO
- Can we use unlabeled data for regularization?
 - If data points i and j are similar (i.e. weight w_{ij} is large), then their predicted labels f_i and f_j are similar

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

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

Graph based smoothness prior
on labeled and unlabeled data

Graph Regularization

- Specific examples of graph regularization based SSL?

Label Propagation Algorithm

Algorithm 11.1 Label propagation (Zhu and Ghahramani [2002])

Compute affinity matrix \mathbf{W} from (11.1)

Compute the diagonal degree matrix \mathbf{D} by $\mathbf{D}_{ii} \leftarrow \sum_j W_{ij}$

Initialize $\hat{Y}^{(0)} \leftarrow (y_1, \dots, y_l, 0, 0, \dots, 0)$

Iterate

1. $\hat{Y}^{(t+1)} \leftarrow \mathbf{D}^{-1} \mathbf{W} \hat{Y}^{(t)}$

2. $\hat{Y}_l^{(t+1)} \leftarrow Y_l$

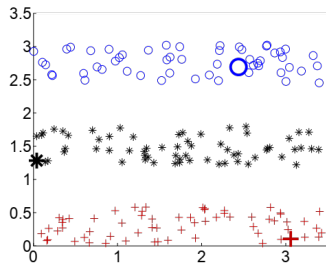
until convergence to $\hat{Y}^{(\infty)}$

Label point x_i by the sign of $\hat{y}_i^{(\infty)}$

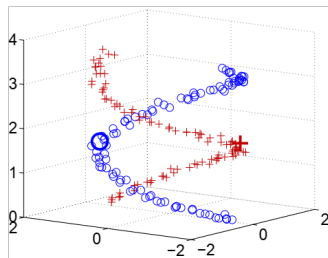
- The algorithm forces the labels on the labeled data
- The algorithm tries to maximize the consistency of the unlabeled examples with the topology of the graph

Label Propagation: Example

Label propagation on two synthetic datasets



(a) 3-Bands



(b) Springs

Label Spreading Algorithm

Algorithm 11.3 Label spreading (Zhou et al. [2004])

Compute the affinity matrix \mathbf{W} from (11.1) for $i \neq j$ (and $\mathbf{W}_{ii} \leftarrow 0$)

Compute the diagonal degree matrix \mathbf{D} by $\mathbf{D}_{ii} \leftarrow \sum_j \mathbf{W}_{ij}$

Compute the normalized graph Laplacian $\mathcal{L} \leftarrow \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$

Initialize $\hat{\mathbf{Y}}^{(0)} \leftarrow (y_1, \dots, y_l, 0, 0, \dots, 0)$

Choose a parameter $\alpha \in [0, 1]$

Iterate $\hat{\mathbf{Y}}^{(t+1)} \leftarrow \alpha \mathcal{L} \hat{\mathbf{Y}}^{(t)} + (1 - \alpha) \hat{\mathbf{Y}}^{(0)}$ until convergence to $\hat{\mathbf{Y}}^{(\infty)}$

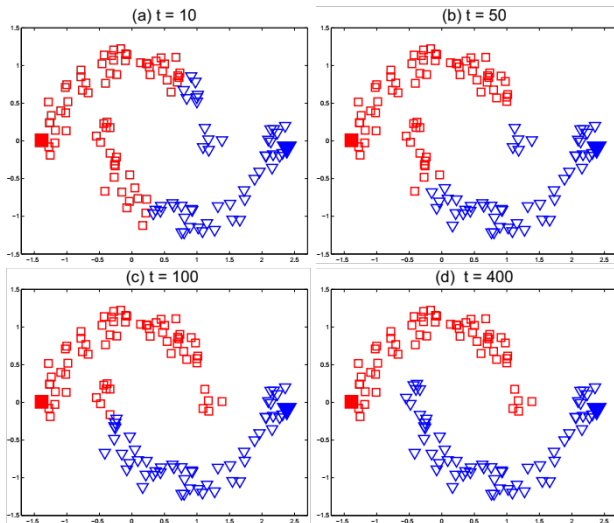
Label point x_i by the sign of $\hat{y}_i^{(\infty)}$

- Similar to the label propagation algorithm, but use the symmetric normalized graph Laplacian¹ instead
- The algorithm does not force the labeled data (useful with noisy data)
- At each step a contribution of the initial labeling is considered (convex combination)

¹The definition is a little bit different from that in spectral clustering.

Label Spreading: Example

Label spreading on the two moon dataset



Real Applications

Classification on Extended Yale Face B dataset



p_L	SRC	G_{ALRR}
50%	97.02	95.42
30%	94.81	94.86
10%	85.08	94.25
5%	74.52	93.41
3%	51.02	91.03

SRC : a sparse representation based classification method

G_{ALRR} : label propagation on a graph constructed by ALRR (Fan et al. 2018)

Real Applications

Classification on MNIST dataset



p_L	CNN	G_{LLE}	G_{ALRR}
50%	98.26	97.74	98.63
30%	97.04	96.33	98.01
10%	95.33	94.52	97.27
5%	93.97	93.11	96.23
3%	91.08	92.26	95.86
1%	83.18	88.75	93.53

G_{LLE} : label propagation on LLE ([lecture 07](#)) graph

G_{ALRR} : label propagation on a graph constructed by ALRR (Fan et al. 2018)

More about label propagation:

Fujiwara, Y., & Irie, G. (2014). Efficient label propagation. In Proceedings of the 31st international conference on machine learning (pp. 784-792).