DDA4210/MAIR6002 Advanced Machine Learning
Lecture 05-II Semi-Supervised Learning

Tongxin Li

School of Data Science, CUHK-Shenzhen

Spring 2024

Overview

Slides Courtesy: Jerry Zhu

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?

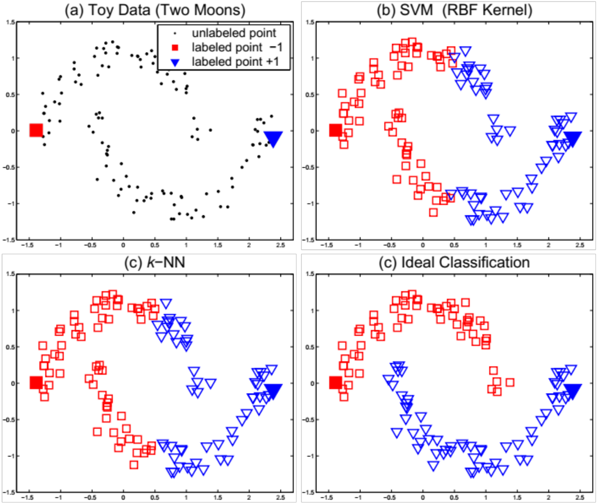
- Labeled data are rare or expensive
  - Human annotation is boring & expensive (fine-tuning GPTs, etc)
  - Labels may require experts
  - Labels may require special devices or money

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.

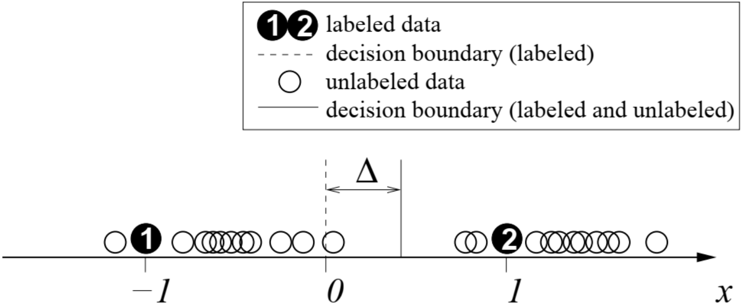


Notation~~s~~

$\{(x_l, y_l)\} \cup \{x_u\}$

- Input (or feature) $\boldsymbol{x} \in \mathcal{X}$, output (or label) $\boldsymbol{y} \in \mathcal{Y}$
- Learner $f : \mathcal{X} \to \mathcal{Y}$
- Labeled data $(X_l, Y_l) = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_l, \mathbf{y}_l)\}$
- Unlabeled data $X_u = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_N\}$, available during training
- Loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$
- Usually, $l \ll N$
- Test data $X_{\text{test}} = \{\mathbf{x}_{N+1}, \ldots\}$, 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 may shift
- This is only one of many ways to use unlabeled data.

SSL Algorithms

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

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 $\boldsymbol{x} \in X_u$
  3. Add $(\boldsymbol{x}, f(\boldsymbol{x}))$ to labeled data
  4. Repeat

Self-Training Algorithm

How to characterize the confidence?

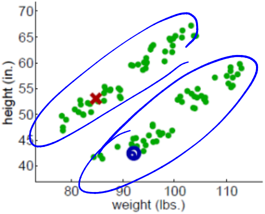
- Some variations
  - Add a few most confident $(\boldsymbol{x}, f(\boldsymbol{x}))$ to labeled data
  - Add all $(\boldsymbol{x}, f(\boldsymbol{x}))$ to labeled data
  - Add all $(\boldsymbol{x}, f(\boldsymbol{x}))$ to labeled data, but with different weights according to the confidence

Self-Training Algorithm: Propagating 1-NN

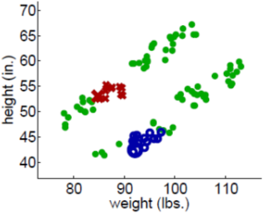
1. Classify $\boldsymbol{x}$ with 1-NN
2. Add $(\boldsymbol{x}, f(\boldsymbol{x}))$ to labeled data, and repeat

Self-Training Algorithm: Propagating 1-NN

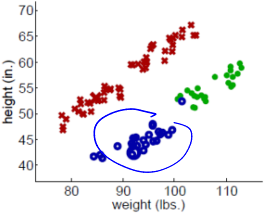
1. Classify $\boldsymbol{x}$ with 1-NN → 1- Nearest Neighbor !
2. Add $(\boldsymbol{x}, f(\boldsymbol{x}))$ to labeled data, and repeat



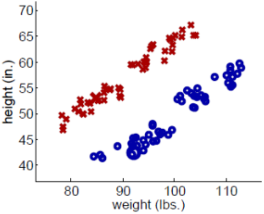
(a) Iteration 1



(b) Iteration 25



(c) Iteration 74

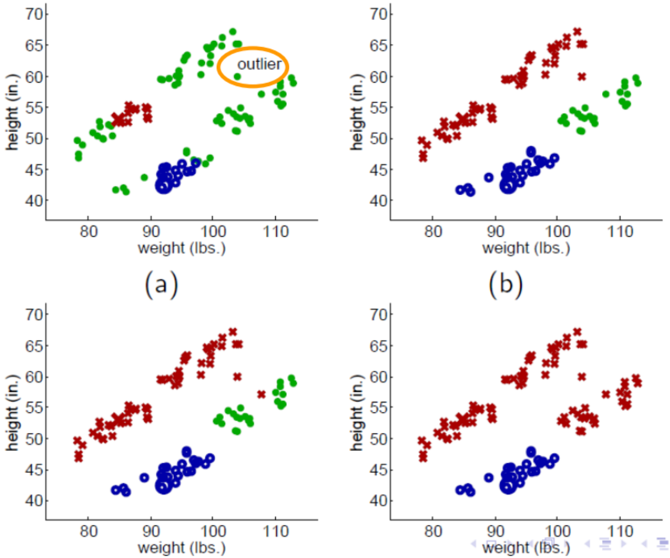


(d) Final labeling of all instances

Q: Any Issues?

Self-Training Algorithm: Propagating 1-NN

It is sensitive to outlier!



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

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!)

(features)

	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									●

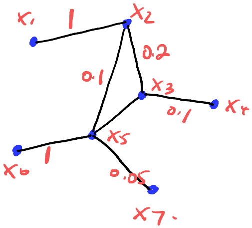
Example 2

Handwritten digits recognition with pixel-wise Euclidean distance

	
not similar	‘indirectly’ similar with stepping stones

Graph-Based Semi-Supervised Learning

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



Graph-Based Semi-Supervised Learning

- **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

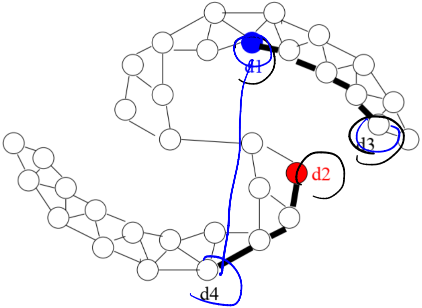
- Nodes $X_l \cup X_u$
- Edges: similarity weights computed from features, e.g.,
  - k-nearest-neighbor graph, unweighted (0, 1 weights)
  - fully connected graph, weight decays with distance

$$w_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2)\right)$$

$W = (w_{ij})_{i,j=1,\cdots,N}$

$N = ? \; |X_{l}| + |X_{u}|$

- 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

let $L = D - W$ be the laplacian.

$f \in \mathbb{R}$

- $\sum_{i,j} w_{ij} (f_i - f_j)^2 = 2 f^T L f$

$f \in \mathbb{R}^d$

- $\sum_{i,j} w_{ij} \| f_i - f_j \|^2 = 2 tr(F^T L F)$

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

- Can we use unlabeled data for regularization?
  - If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar (i.e. weight w_{ij} is large), then their predicted labels (or responses more generally) $f(\boldsymbol{x}_i)$ and $f(\boldsymbol{x}_j)$ are similar.
  - Thus we can solve the following problem

$$\min_f \underbrace{\sum_{i=1}^{l} \ell(y_i, f(\boldsymbol{x}_i))}_{\text{loss on labeled data}} + \lambda \underbrace{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\|^2}_{\text{graph-based regularization on both labeled and unlabeled data}}$$

Graph Regularization

- Specific examples of graph regularization based SSL?

$$\min_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log f_{ik} + \lambda \, \mathrm{tr}(F^T L F)$$

• can be used as a regularizer in deep learning

Label Propagation Algorithm

can be obtained in many ways (similar to the spectral clustering)

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

Compute affinity matrix $\mathbf{W}$

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

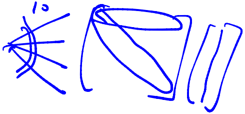
Initialize $\hat{Y}^{(0)} \leftarrow (y_1, \ldots, \underbrace{y_l, 0, 0, \ldots, 0}_{u+k})$

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 $\hat{y}_i^{(\infty)}$

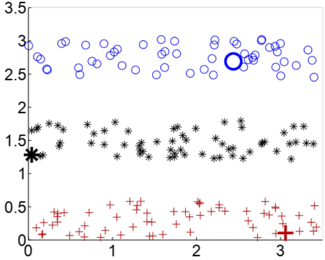


- The algorithm forces the labels on the labeled data
- The algorithm tries to maximize~~s~~ the consistency of the unlabeled examples with the topology of the graph

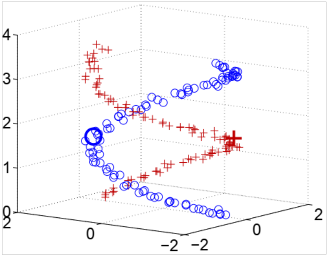
W

Label Propagation: Example

Label propagation on two synthetic datasets



(a) 3-Bands



(b) Springs

Real Applications

Classification on Extended Yale Face B dataset

- SRC: State-of-the-Art supervised classifier
- G_{ALRR}: semi-supervised

percentage of labeled data.

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)

"Low-Rank Representation"

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).