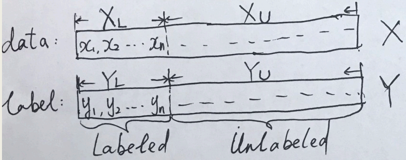
SEMI-SURPERVISED LEARNING

Chris Ding
UT Arlington

(1)

SSL (semi-surpervised Learning)



Labeled training data $\mathcal{L} = \{x_i, y_i\}_{i=1}^{L}$

Unlabeled data $U = \{x_i\}_{i=L+1}^{L+U}$

(usually $U >> L$)

Inductive Learning

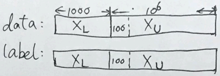
Our standard classification.

Learn the classifier using (X_L, Y_L)

classify $X_U \rightarrow Y_U$

Transductive Learning

Model $(X_L, X_U, Y_L) \rightarrow Y_U$



A simple Algorithm: Self-training

steps: 1. Learn classifier using (X_L, Y_L)

2. Classify $X_U \to Y_U$ (Scores: $S_{L+1}, S_{L+2} \cdots S_{L+U}$)

3. pick most "confidently predicted labels" $(X_L^{(1)}, Y_L^{(1)})$ (1): first Interation

Sort $Y_U = \{y_{l+1}, y_{l+2}, \cdots, y_{l+u}\}$ to $Y_U = \{\tilde{y}_{l+1}, \tilde{y}_{l+2}, \cdots \tilde{y}_{l+u}\}$ according to ~~according~~ the Scores in descenting order

here we assume ~~we~~ that pick up the first 100 data in X_U

4. Enlarge Training data $(X_L + X_L^{(1)}, Y_L + Y_L^{(1)})$. remove those data from (X_U, Y_U)

5. Repeat from step 1

Caution: Prediction mistake can reinforce itself!

Examples: Pick most confidently predicted labels: build each $X_t \rightarrow y_t$ with <u>score</u>

(1) KNN ($K=3$) Score

	KNN labels	
X_{t+1}	1, 1, 3	$\longrightarrow y_{t+1}=\text{"1"}$ with $Prob=\frac{2}{3} \Rightarrow Score=\frac{2}{3}$
X_{t+2}	1, 1, 1.	$\longrightarrow y_{t+2}=\text{"1"}$ with $Prob=1 \Rightarrow Score=1$ ✓

(2) Centroid method score

X_{t+1} $\quad \|X_{t+1}-\mu_1\| = 3.5 \leftarrow$

$\qquad\quad \|X_{t+1}-\mu_2\| = 4.6$

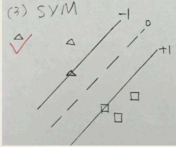
X_{t+2} $\quad \|X_{t+2}-\mu_1\| = 2.1 \leftarrow$ ✓

$\qquad\quad \|X_{t+2}-\mu_2\| = 4.7$

prediction X_{t+2} is more confident compare to X_{t+1},

in this case we can use some monotonically decreasing function to calculate the score

(3) SVM



confident depends on distance from boundary $f=0$

we can calculate the score according the distance.

(5)

Solve $\min_{\{B, Y_U\}} \| [Y_L, Y_U] - B^T [X_L, X_U] \|_F^2$

initialize Y_U using inductive learning $\Rightarrow$ $\left\{ \begin{array}{l} \text{Solve } \min_B \|Y_L - B^T X_L\|_F^2 \\ Y_U = B^{*T} X_U \\ Y = [Y_L, Y_U] \text{ , } X = [X_L, X_U] \end{array} \right.$

M-step : solve $\min_B \|Y - B^T X\|_F^2$

get B^* (*: the optimal solution)

E-step : Solve $\min_{Y_U} \|[Y_L, Y_U] - B^{*T} X\|_F^2$

$Y_U^* = B^T X_U$

repeat M, E step until convergence.

(4)

Transdutive learning vs Inductive learning

(linear Regression) : $\min_{B} \|Y - B^T X\|_F^2$

transductive learn : $\min_{\{B, Y_U\}} \|[Y_L, Y_U] - B^T[X_L, X_U]\|_F^2$

Inductive learn : $\min_{\{B\}} \|Y_L - B^T X_L\|_F^2$, $Y_U = B^{*T} X_U$

Graph Regularization

Assume the predictions on the entire data $L \cup U$ to be defined by f

Graph regularization assumes that function f is smooth

(similar examples i and j should have similar prediction f_i and f_j)

objective: $$\min_f \underbrace{\sum_{i \in L} (y_i - f_i)^2}_{\text{minimizing the loss on labeled data}} + \lambda \underbrace{\sum_{i,j \in L,U} W_{ij} (f_i - f_j)^2}_{\text{ensures smoothess of labels of labeled and unlabeled data}}$$

λ: trade-off parameter.

(Minimization make f_i and f_j to be very similar if W_{ij} is large)