Discussions on Semi-Supervised Learning and Active Learning

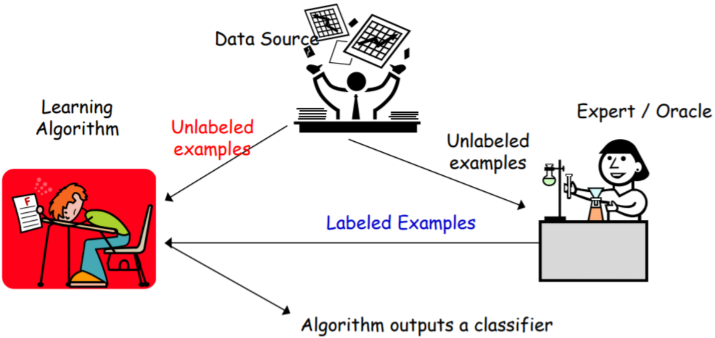
Xin Deng

ShanghaiTech University

dengxin1@shanghaitech.edu.cn

April 20, 2020

Semi-Supervised Learning



$S_l = \{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$

x_i drawn i.i.d from D, $y_i = c^*(x_i)$

$S_u = \{x_1, \ldots, x_{m_u}\}$ drawn i.i.d from D

Goal: h has small error over D.

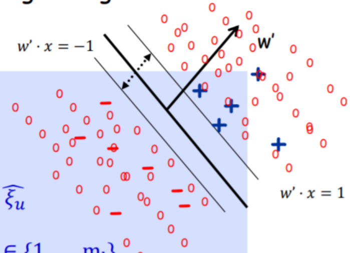
$$\mathrm{err}_D(h) = \Pr_{x \sim D}(h(x) \neq c^*(x))$$

Semi-Supervised Learning

Outline:

1. Semi-supervised SVM
2. Co-training
3. Graph-based methods

Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]



Input: $S_l=\{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$

$S_u=\{x_1, \ldots, x_{m_u}\}$

$$\mathrm{argmin}_w \, ||w||^2 + C \sum_i \xi_i + C \sum_u \widehat{\xi_u}$$

- $y_i \, w \cdot x_i \geq 1-\xi_i$, for all $i \in \{1, \ldots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1 - \widehat{\xi_u}$, for all $u \in \{1, \ldots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$

It's a convex problem, but NP-hard.

Semi-supervised SVM

Heuristic (Joachims) high level idea:

- First maximize margin over the labeled points
- Use this to give initial labels to unlabeled points based on this separator.
- Try flipping labels of unlabeled points to see if doing so can increase margin

Co-training

Assumptions between two parts:

1. examples contain two sufficient sets of features, $x = < x_1, x_2 >$
2. belief: the parts are consistent, i.e., $\exists c_1, c_2$ s.t. $c_1(x_1) = c_2(x_2) = c^*(x)$

Co-training

Training **2** classifiers, one on each type of info.
Using each to help train the other.

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.

Co-training/Multi-view SSL: Direct Optimization of Agreement

Input: $S_l=\{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$
$S_u=\{x_1, \ldots, x_{m_u}\}$

$$\mathrm{argmin}_{h_1,h_2} \sum_{l=1}^{2}\sum_{i=1}^{m_l} l(h_l(x_i), y_i) + C\sum_{i=1}^{m_u} \mathrm{agreement}(h_1(x_i), h_2(x_i))$$

Each of them has small labeled error

Regularizer to encourage agreement over unlabeled dat

Graph-based methods

Main idea:

- Construct graph G with edges between very similar examples
- Might have also glued together in G examples of different classes.
- Run a graph partitioning algorithm to separate the graph into pieces.

How to create the graph:

$$G =< V, E >= \begin{cases} V : datapoints\ (S_l \cup S_u) \\ E : Similarity/Weights \end{cases}$$

1. Adjacency graph:
   - K-NN
   - $\sum$-NN, where $\sum$ is the distance of two data points
2. Graph weights
   - Simple formulation: $\{0,1\}$
   - Gaussian kernel function

Minimum "soft cut"

Initialization:

$$f_i = \begin{cases} \pm 1, x_i \in S_l \\ 0, \quad x_i \in S_u \end{cases}$$

Prediction:

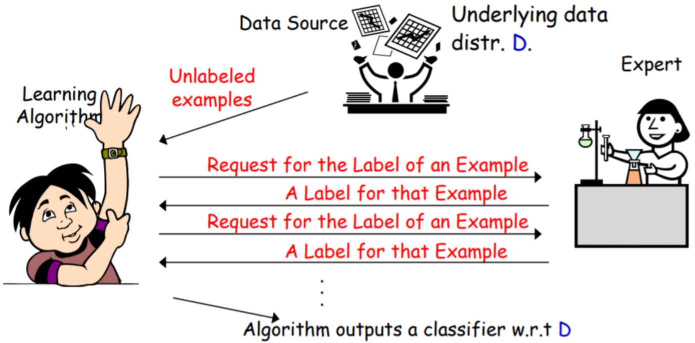
$$y_i = sign(f_i)$$

Training:

$$\min \sum_{i,j} w_{i,j} (f_i - f_j)^2$$

$$\text{s.t.} \quad f_i = y_i, \qquad x_i \in S_l$$

Active Learning



- Learner can choose specific examples to be labeled.
- Goal: use fewer labeled examples [pick informative examples to be labeled].

What Makes a Good Active Learning Algorithm?

- Guaranteed to output a relatively good classifier for most learning problems.
- Doesn't make too many label requests.
- Need to choose the label requests carefully, to get informative labels.

Disagreement Based Active Learning Hypothesis Space Search

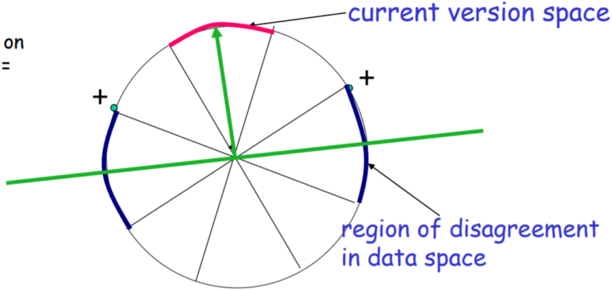
Definition (CAL'92)

Version space: part of H consistent with labels so far.

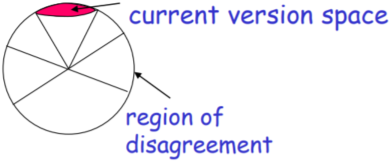
Region of disagreement = part of data space about which there is still some uncertainty (i.e. disagreement within version space)

$x \in X, x \in DIS(VS(H))$ iff $\exists h_1, h_2 \in VS(H), h_1(x) \neq h_2(x)$

E.g.,: data lies on circle in R^2, H = homogeneous linear seps.



A^2 Agnostic Active Learner



Algorithm:

Careful use of generalization bounds;
Avoid the sampling bias!!!!

Let $H_1 = H$.

For $t = 1, \ldots,$

- Pick a few points at random from the current region of disagreement $DIS(H_t)$ and query their labels.
- Throw out hypothesis if you are statistically confident they are suboptimal.

The DHN Agnostic Active Learner

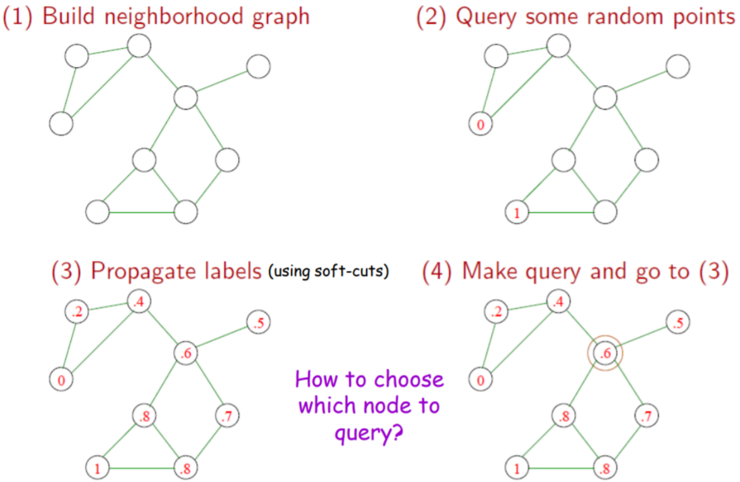
$S = \emptyset$ (points with inferred labels)
$T = \emptyset$ (points with queried labels)
For $t = 1, 2, \ldots$:
  Receive x_t
  If $(h_{+1} = \texttt{learn}(S \cup \{(x_t, +1)\}, T))$ fails: Add $(x_t, -1)$ to S and break
  If $(h_{-1} = \texttt{learn}(S \cup \{(x_t, -1)\}, T))$ fails: Add $(x_t, +1)$ to S and break
  If $\mathrm{err}(h_{-1}, S \cup T) - \mathrm{err}(h_{+1}, S \cup T) > \Delta_t$: Add $(x_t, +1)$ to S and break
  If $\mathrm{err}(h_{+1}, S \cup T) - \mathrm{err}(h_{-1}, S \cup T) > \Delta_t$: Add $(x_t, -1)$ to S and break
  Request y_t and add (x_t, y_t) to T

Figure 16: The DHM selective sampling algorithm. Here, $\mathrm{err}(h, A) = (1/|A|) \sum_{(x,y)\in A} 1(h(x) \neq y)$. A possible setting for Δ_t is shown in Equation 1. At any time, the current hypothesis is $\texttt{learn}(S, T)$.

$\texttt{learn}(A, B)$ returns a hypothesis $h \in \mathcal{H}$ consistent with A, and with minimum error on B. If there is no hypothesis consistent with A, a failure flag is returned.

$$\Delta_t = \beta_t^2 + \beta_t \left(\sqrt{\mathrm{err}(h_{+1}, S \cup T)} + \sqrt{\mathrm{err}(h_{-1}, S \cup T)} \right), \qquad \beta_t = C\sqrt{\frac{d \log t + \log(1/\delta)}{t}}$$

Active learning with label propagation



Active learning with label propagation

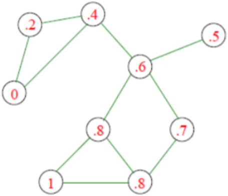
Instead, use a 1-step-lookahead heuristic:

- For a node with label p, assume that querying will have prob p of returning answer 1, $1-p$ of returning answer 0.
- Compute "average confidence" after running soft-cut in each case:

$$p\frac{1}{n}\sum_{x_i}\max(f_1(x_i), 1-f_1(x_i)) + (1-p)\frac{1}{n}\sum_{x_i}\max(f_0(x_i), 1-f_0(x_i))$$

- Query node s.t. this quantity is highest (you want to be more confident on average).

(3) Propagate labels (using soft-cuts)



(4) Make query and go to (3)