Semi-supervised learning II

CS534
Semi-supervised SVM ($S^3VM$)

Assumption: Unlabeled data from different classes are separated with large margin
Standard soft margin SVMs

• keep labeled points outside the margin, while maximizing the margin:

\[
\min_{h, b, \xi} \sum_{i=1}^{l} \xi_i + \lambda \|h\|^2_{HK}
\]

subject to \(y_i(h(x_i) + b) \geq 1 - \xi_i\), \(\forall i = 1 \ldots l\)

\(\xi_i \geq 0\)

• Equivalent to

\[
\min_{f} \sum_{i=1}^{l} (1 - y_i f(x_i))^+ + \lambda \|h\|^2_{HK}
\]

\(y_i f(x_i)\) known as the margin, \((1 - y_i f(x_i))^+\) the hinge loss
To incorporate unlabeled points,
- assign putative labels \( \text{sign}(f(x)) \) to \( x \in X_u \)
- Hinge loss on unlabeled points becomes
  \( (1 - |f(x)|)_+ \)

New objective:

\[
\min_f \sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \lambda_1 \|h\|_{\mathcal{H}_K}^2 + \lambda_2 \sum_{i=l+1}^{n} (1 - |f(x_i)|)_+ 
\]
The hat loss on unlabeled data

hinge loss \((1 - y_i f(x_i))_+\)

hat loss \((1 - |f(x_i)|)_+\)

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

• often unbalanced – most points classified into one class

• Heuristic for encouraging class balance

\[
\frac{1}{n-l} \sum_{i=l+1}^{n} f(x_i) = \frac{1}{l} \sum_{i=1}^{l} y_i.
\]
Putting everything together

\[
\min_{f} \sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \lambda_1 \|f\|_{\mathcal{H}_k}^2 + \lambda_2 \sum_{i=l+1}^{n} (1 - |f(x_i)|)_+
\]

s.t. \[
\frac{1}{n-l} \sum_{i=l+1}^{n} f(x_i) = \frac{1}{l} \sum_{i=1}^{l} y_i
\]

- \(S^3VM\) objective is non-convex
- NP-hard. Algorithm instead does local optimization.
  - Start with large margin over labeled data. Induces labels on U.
  - Then try flipping labels in greedy fashion
- Quite successful on text classification
Graph-based methods

• Suppose we believe that very similar examples probably have the same label.
• If you have a lot of labeled data, this suggests a Nearest-Neighbor type of alg.
• If you have a lot of unlabeled data, perhaps can use them as “stepping stones”
High level idea

• Construct a graph with edges between very similar examples

• Unlabeled data can help “glue” the objects of the same class together
Graph-based semi-supervised learning

• 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, weighted ($w = \frac{\exp(-|x_i-x_j|^2)}{\sigma^2}$)
  – $\epsilon$-radius graph

• Assumption: instances that are connected by heavy edges tend to have the same label
The Graph Mincut algorithm
(Blum & Chawla 2001)

• Fix $Y_l$, find $Y_u \in \{0,1\}^{n-l}$ to minimize $\sum_{ij} w_{ij} |y_i - y_j|^2$
  
  – This encourages similar examples to have the same class label

• This is equivalent to the following optimization problem:

$$\min_{Y \in \{0,1\}^n} \sum_{i=1}^{l} (y_i - Y_{l_i})^2 + \sum_{ij} w_{ij} (y_i - y_j)^2$$

• Minimizing this objective corresponds to finding a labeling of the unlabeled examples s.t. 1-nearest neighbor classifier has the smallest leave-one-out cross-validation error
  
  – A commonly used principle: labeling the unlabeled data in a way that make the underlying learning algorithm “happiest”
The Graph Mincut algorithm (cont)

• If binary label, can be solved by min-cut on a modified graph – adding source and sink nodes with large weights to labeled examples

• A polynomial-time max-flow algorithm can be used to solve the optimization problem optimally
Co-training for Semi-Supervised Learning
(Blum and Mitchell 1998)

- Assumes feature \( X \) is very expressive and has redundant information
- Exploits redundant information for semi-supervised learning
- Redundant info:
  - Text in the document
  - Anchor text for hyperlinks
Two view of an instance – another example

Example: named entity classification Person (Mr. Washington) or Location (Washington State)

instance 1: ... headquartered in (Washington State) ...
instance 2: ... (Mr. Washington), the vice president of ...

- a named entity has two views (subset of features) $x = [x^{(1)}, x^{(2)}]$
- the words of the entity is $x^{(1)}$
- the context is $x^{(2)}$
More data

With more unlabeled data

instance 1: ... headquartered in (Washington State)$^L$ ...

instance 2: ... (Mr. Washington)$^P$, the vice president of ...

instance 3: ... headquartered in (Kazakhstan) ...

instance 4: ... flew to (Kazakhstan) ...

instance 5: ... (Mr. Smith), a partner at Steptoe & Johnson ...

test: ... (Robert Jordan), a partner at ...

test: ... flew to (China) ...
The Co-training algorithm

**Input:** labeled data $\{(x_i, y_i)\}_{i=1}^l$, unlabeled data $\{x_j\}_{j=l+1}^{l+u}$

- each instance has two views $x_i = [x_i^{(1)}, x_i^{(2)}]$
- and a learning speed $k$.

1. let $L_1 = L_2 = \{(x_1, y_1), \ldots, (x_l, y_l)\}$.

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 $(x, f^{(1)}(x))$ to $L_2$.
   Add $f^{(2)}$'s top $k$ most-confident predictions $(x, f^{(2)}(x))$ to $L_1$.
   Remove these from the unlabeled data.
Experimental Results

- 12 labeled web pages
- 1,000 additional unlabeled web pages
- Learning algorithm: Naïve Bayes
- Average error:
  - learning from labeled data only using combined classifier: ~11%
  - Co-training: ~5%
Co-training Theory

Assumptions

- feature split $x = [x^{(1)}; x^{(2)}]$ exists
- $x^{(1)}$ or $x^{(2)}$ alone is sufficient to train a good classifier
- $x^{(1)}$ and $x^{(2)}$ are conditionally independent given the class
Understanding Co-Training: A Simple Setting

• Unlabeled data defines the connected components
• Suppose we have infinite unlabeled data, we obtain the correct bipartite graph
• Labeled data provides labels to the connected components
• Each component will only need one labeled data
• Co-training with unlabeled data reduces the number of labeled data points needed
Co-Training Theoretical Result
(Blum and Mitchell COLT1998)

• If
  – \( x^{(1)}, x^{(2)} \) are conditionally independent given \( y \)
  – and \( f \) is PAC learnable from noisy \textit{labeled} data
    • e.g., give me an \( \varepsilon \) and \( \delta \), I give you \( h \) such that error \( \leq \varepsilon \) with prob. at least \( 1-\delta \)
• Then
  – \( f \) is PAC learnable from \textbf{weak initial classifier} plus \textit{unlabeled} data
    • Basic idea: \( f_1(x^{(1)}) \) can be considered as the noisy label for \( x^{(2)} \) and vice versa
Multiview learning – extending co-training

- Loss Function: $c(x, y, f(x)) \in [0, \infty)$. For example,
  - squared loss $c(x, y, f(x)) = (y - f(x))^2$
  - 0/1 loss $c(x, y, f(x)) = 1$ if $y \neq f(x)$, and 0 otherwise.

- Empirical risk: $\hat{R}(f) = \frac{1}{l} \sum_{i=1}^{l} c(x_i, y_i, f(x_i))$

- Regularizer: $\Omega(f)$, e.g., $\|f\|^2$

- Regularized Risk Minimization $f^* = \arg\min_{f \in \mathcal{F}} \hat{R}(f) + \lambda \Omega(f)$

Include an additional regularizer defined on unlabeled data to encourage agreement among multiple learners:

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{v=1}^{k} \left( \sum_{i=1}^{l} c(x_i, y_i, f_v(x_i)) + \lambda_1 \Omega_{SL}(f_v) \right)$$

$$+ \lambda_2 \sum_{u,v=1}^{k} \sum_{i=l+1}^{l+u} c(x_i, f_u(x_i), f_v(x_i))$$
Summary: Semi-Supervised Learning

• Generative methods – Mixture models, the underlying components correspond to the classes
• Semi-Supervised SVMs – assume unlabeled data from different classes have large margin
• Graph based methods – assume similar examples have similar labels
• Co-training, multi-view methods – assume different views should give self consistent predictions
• Many others ...
SSL algorithms can use unlabeled data to help improve prediction accuracy
• if data satisfies the underlying assumptions