Semi-supervised Learning

CS534

Adapted from “Tutorial on Semi-supervised learning” by Xiaojin Zhu
Why Semi-Supervised Learning

• Unsupervised and Supervised learning
  – Two extreme learning paradigms
  – Unsupervised learning
    • e.g., collection of documents without any labels
    • easy to collect
  – Supervised learning
    • each object labeled with a class.
    • Expensive to do

• Real life applications are somewhere in between – Semi-supervised Learning
Semi-supervised learning

• Traditional supervised learning
  – Given labeled data $D_L = \{(x_i, y_i)\}, i = 1, \ldots, n$, learn classifier $h$

• Semi-supervised learning
  – Given labeled data $D_L = \{(x_i, y_i)\}, i = 1, \ldots, n$, and unlabeled data $D_U = \{x_i\}, i = 1, \ldots, m$, learn a classifier $h$
  – Often have $m \gg n$
Why can unlabeled data help?

- Assume that the class boundary should go through low density areas
- Having the unlabeled points help identify better decision boundary
Why can unlabeled data help?

- Assume that each class contains a coherent group of points (e.g., Gaussian)
- Having the unlabeled points can help learn the distribution more accurately
Major SSL methods

• Generative models
  – Use unlabeled data to more accurately estimate the models

• Discriminative models
  – Assume that $p(y|x)$ is locally smooth
  – Graph/manifold regularization

• Multi-view approach – multiple independent learners that agrees on unlabeled data
  – Cotraining
Bayes Gaussian Classifier

• Assume that each class is a Gaussian, what is the decision boundary?
Adding Unlabeled Data

• What is the decision boundary now?
Semi-supervised Learning

- Different learning goals
- In SSL, the learned $\theta$ needs to explain the unlabeled data well too
Generative model for semi-supervised learning

• The joint and marginal likelihood

\[ p(X_l, Y_l, X_u|\theta) = \sum_{Y_u} p(X_l, Y_l, X_u, Y_u|\theta) \]

• Find the maximum likelihood estimate, or MAP estimate

• Common mixture models used in semi-supervised learning
  – GMM – image classification
  – Mixture of Multinomials (naïve bayes) – text classification

• Learning through Expectation Maximization (EM)
Gaussian Mixture model

• Binary classification with GMM using MLE.
  – with only labeled data

\[
\log p(X_i, Y_i | \theta) = \sum_{i=1}^{l} \log p(y_i | \theta) p(x_i | y_i, \theta)
\]

• MLE is trivial (sample mean and covariance)
  – With both labeled and unlabeled data

\[
\log p(X_l, Y_l, X_u | \theta) = \sum_{i=1}^{l} \log p(y_i | \theta) p(x_i | y_i, \theta) + \sum_{i=l+1}^{l+u} \log \left( \sum_{y=1}^{2} p(y | \theta) p(x_i | y, \theta) \right)
\]

• MLE harder (hidden variables) – use EM
Semi-Supervised EM for GMM

1. Start from MLE $\theta = \{w, \mu, \Sigma\}_{1:2}$ on $(X_l, Y_l)$,
   - $w_c =$ proportion of class $c$
   - $\mu_c =$ sample mean of class $c$
   - $\Sigma_c =$ sample cov of class $c$

   repeat:

2. The E-step: compute the expected label $p(y|x, \theta) = \frac{p(x, y|\theta)}{\sum_{y'} p(x, y'|\theta)}$ for all $x \in X_u$
   - label $p(y = 1|x, \theta)$-fraction of $x$ with class 1
   - label $p(y = 2|x, \theta)$-fraction of $x$ with class 2

3. The M-step: update MLE $\theta$ with (now labeled) $X_u$
Assumption for SSL GMM

- Assumption: the data actually come from the mixture model, and the mixture components and classes correspond well with one another
- When the assumption is wrong
When assumption is wrong

- Down-weight the unlabeled data ($\lambda < 1$)

\[
\log p(X_l, Y_l, X_u|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(x_i|y_i, \theta) \\
+ \lambda \sum_{i=l+1}^{l+u} \log \left( \sum_{y=1}^{2} p(y|\theta)p(x_i|y, \theta) \right)
\]
Input: \((x_1, y_1), \ldots, (x_l, y_l), x_{l+1}, \ldots, x_{l+u}\),
a clustering algorithm \(\mathcal{A}\), a supervised learning algorithm \(\mathcal{L}\)

1. Cluster \(x_1, \ldots, x_{l+u}\) using \(\mathcal{A}\).
2. For each cluster, let \(S\) be the labeled instances in it:
3. Learn a supervised predictor from \(S\): \(f_S = \mathcal{L}(S)\).
4. Apply \(f_S\) to all unlabeled instances in this cluster.

Output: labels on unlabeled data \(y_{l+1}, \ldots, y_{l+u}\).

But again: **SSL sensitive to assumptions**—in this case, that the clusters coincide with decision boundaries. If this assumption is incorrect, the results can be poor.
Cluster-then-label: success

Example: $\mathcal{A}$=Hierarchical Clustering, $\mathcal{L}$=majority vote.
Cluster-then-label: failure

Example: $A =$ Hierarchical Clustering, $L =$ majority vote.
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 \( x^{(2)} \)
With more unlabeled data

instance 1: ... headquartered in \((\text{Washington State})^L\) ...
instance 2: ... \((\text{Mr. Washington})^P\), the \underline{vice president} of ...
instance 3: ... headquartered in \((\text{Kazakhstan})\) ...
instance 4: ... \underline{flew to} \((\text{Kazakhstan})\) ...
instance 5: ... \((\text{Mr. Smith})\), a \underline{partner} at Steptoe & Johnson ...

test: ... \((\text{Robert Jordan})\), a \underline{partner} at ...


test: ... \underline{flew to} \((\text{China})\) ...
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 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 weak initial classifier plus 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

• Unlabeled data can significantly help improve classification accuracy

• Combining generative probabilistic models and EM leads to natural use of unlabeled data
  – Unlabeled data don’t always lead to performance gain
  – Depend on whether the generative model is correct or not

• Co-training assumes that there are two redundant and conditionally independent feature sets
  – In practice there is often no natural split of features
  – Random splits can help as well

• Multi-view learning introduces regularization terms that encourages consistency of different learners on unlabeled data