Semi-Supervised Learning

- The main models we have been studying (PAC, mistake-bound) are for supervised learning.
  - Given labeled examples $S = \{(x_i, y_i)\}$, try to learn a good prediction rule.
- Unfortunately, labeled data is often expensive.
- On the other hand, unlabeled data is often plentiful and cheap.
  - Documents, images, OCR, web-pages, protein sequences, ...

Can we use unlabeled data to help?
**Semi-Supervised Learning**

- Two scenarios: active learning and semi-supervised learning.
  - **Active learning**: have ability to ask for labels of unlabeled points of interest.
    - Can you do better than just ask for labels on random subset?
  - **Semi-supervised learning**: no querying. Just have lots of additional unlabeled data.
    - Will look today at SSL. This is the most puzzling one since unclear what unlabeled data can do for you.

---

**Semi-Supervised Learning**

Given a set $L$ of labeled data and set $U$ of unlabeled data. Can we use $U$ to help?

- What can the unlabeled data possibly do for us?
- Abstract high-level answer we will get to is:
  - Going back to “Occam’s razor”, unlabeled data can help us improve our notion of what is simpler than what, by identifying regularities that appear in the data.

- But first:
  - Discuss several methods that have been developed for using unlabeled data to help.
  - Then will give an extension of PAC model to make sense of what’s going on.
Plan for today

Methods:
• Co-training
• Transductive SVM
• Graph-based methods

Model:
• Augmented PAC model for SSL.

There’s also a book “Semi-supervised learning” on the topic.

Co-training
[B&Mitchell’98] motivated by [Yarowsky’95]

Yarowsky’s Problem & Idea:
• Some words have multiple meanings (e.g., “plant”). Want to identify which meaning was intended in any given instance.

• Standard approach: learn function from local context to desired meaning, using labeled data. “…nuclear power plant generated…”

• Idea: use fact that in most documents, multiple uses have same meaning. Use to transfer confident predictions over.
Co-training
Actually, many problems have a similar characteristic.
• Examples \(x\) can be written in two parts \((x_1, x_2)\).
• Either part alone is in principle sufficient to produce a good classifier.
• E.g., speech+video, image and context, web page contents and links.
• So if confident about label for \(x_1\), can use to impute label for \(x_2\), and vice versa. Use each classifier to help train the other.

"Multi-view learning"

Example: classifying webpages
• Co-training: Agreement between two parts
  - examples contain two sets of features, i.e. an example is \(x=(x_1, x_2)\) and the belief is that the two parts of the example are sufficient and consistent, i.e. \(\exists c_1, c_2\) such that \(c_1(x_1)=c_2(x_2)=c(x)\)
**Example: intervals**

Suppose $x_1 \in \mathbb{R}$, $x_2 \in \mathbb{R}$. $c_1 = [a_1, b_1]$, $c_2 = [a_2, b_2]$

---

**Co-Training Theorems**

- [BM98] if $x_1, x_2$ are independent given the label: $D = p(D_1^+ \times D_2^+) + (1-p)(D_1^- \times D_2^-)$, and if $C$ is SQ-learnable (or from random class noise), then can learn from an initial “weakly-useful” $h_1$ plus unlabeled data.

- **Def:** $h$ is weakly-useful if

$$\Pr[h(x)=1|c(x)=1] > \Pr[h(x)=1|c(x)=0] + \varepsilon.$$  

(same as weak hyp if target $c$ is balanced)

- E.g., say “syllabus” appears on 1/3 of course pages but only 1/6 of non-course pages.

- **Idea:** use as a noisy label of other view. (helpful trick: balance data so observed labels are 50/50)
**Co-Training Theorems**

- [BB] in some cases (e.g., LTFs), you can use this to learn from a single labeled example.
  - Pick random hyperplane and boost (using above).
  - Repeat process multiple times.
  - Get 4 kinds of hyps: \{close to $c$, close to $\neg c$, close to 1, close to 0\}
  - Just need one labeled example to choose right one.
- [BBY] if don’t want to assume independence, and $C$ is learnable from positive data only, then suffices for $D^+$ to have *expansion*.

**Co-Training and expansion**

Want initial sample to expand to full set of positives after limited number of iterations.
Transductive SVM [Joachims99]

- Suppose we believe target separator goes through low density regions of the space/large margin.
- Aim for separator with large margin wrt labeled and unlabeled data. (L+U)

Unfortunately, optimization problem is now 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.
Transductive SVM [Joachims99]

- Suppose we believe target separator goes through low density regions of the space/large margin.
- Aim for separator with large margin wrt labeled and unlabeled data. (L+U)
- Unfortunately, optimization problem is now NP-hard. Algorithm instead does local optimization.
  - Also, work on polynomial-time approximation algorithms. (“furthest hyperplane problem”)

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, suggests a graph-based method.
**Graph-based methods**

- Transductive approach. *(Given L + U, output predictions on U).*
- Construct a graph with edges between very similar examples.

- Solve for:
  - Minimum cut
  - Minimum “soft-cut” *[ZhuGhahramaniLafferty]*
  - Spectral partitioning

---

**Graph-based methods**

- Suppose just two labels: 0 & 1.
- Solve for labels $f(x)$ for unlabeled examples $x$ to minimize:
  - $\sum_{e=(u,v)}|f(u)-f(v)|$ *[soln = minimum cut]*
  - $\sum_{e=(u,v)}(f(u)-f(v))^2$ *[soln = electric potentials]*
- In case of min-cut, can use counting/VC-dim results to get confidence bounds.
  - VC-dimension of class of cuts of size $k$ is $O(k/\lambda_{\text{min}})$, where $\lambda_{\text{min}}$ is the minimum nontrivial cut in the graph. *[Kleinberg]*
How can we think about these approaches to using unlabeled data in a PAC-style model?

**PAC-SSL Model [BB]**

- **Augment** the notion of a concept class $C$ with a notion of compatibility $\chi$ between a concept and the data distribution.
  - “learn $C$” becomes “learn $(C,\chi)$” (i.e. learn class $C$ under compatibility notion $\chi$)

- Express relationships that one hopes the target function and underlying distribution will possess.

- **Idea**: use unlabeled data & the belief that the target is compatible to reduce $C$ down to just {the highly compatible functions in $C$}.
  - Or, order the functions in $C$ by compatibility.
**PAC-SSL Model [BB]**

- Augment the notion of a concept class $C$ with a notion of compatibility $\chi$ between a concept and the data distribution.
  - “learn $C$” becomes “learn $(C, \chi)$” (i.e. learn class $C$ under compatibility notion $\chi$)

- To do this, need to be able to estimate compatibility of $h$ with $D$ from unlabeled data.

- Require that the degree of compatibility be something that can be estimated from a finite sample.

**PAC-SSL Model [BB]**

- Augment the notion of a concept class $C$ with a notion of compatibility $\chi$ between a concept and the data distribution.
  - “learn $C$” becomes “learn $(C, \chi)$” (i.e. learn class $C$ under compatibility notion $\chi$)

- Require $\chi$ to be an expectation over individual examples:
  - $\chi(h,D) = E_{x \sim D}[\chi(h, x)]$ = compatibility of $h$ with $D$, $\chi(h, x) \in [0,1]$,
  - $err_{unl}(h) = 1 - \chi(h, D)$ = incompatibility of $h$ with $D$ (unlabeled error rate of $h$)
Margins, Compatibility

- **Margins**: belief is that should exist a large margin separator.

- **Incompatibility of h and D** (unlabeled error rate of h): the probability mass within distance $\gamma$ of h.

- Can be written as an expectation over individual examples

  $\chi(h, D) = E_{x \sim D} [\chi(h, x)]$

  where:

  - $\chi(h, x) = 0$ if $\text{dist}(x, h) < \gamma$
  - $\chi(h, x) = 1$ if $\text{dist}(x, h) > \gamma$

  **err$_{unl}$** $(h) = \Pr_{x \sim D} [\text{dist}(x, h) < \gamma]$

Margins, Compatibility

- **Margins**: belief is that should exist a large margin separator.

- If do not want to commit to $\gamma$ in advance, define $\chi(h, x)$ to be a smooth function of $\text{dist}(x, h)$, e.g.:

  $\chi(h, x) = 1 - e^{\left[-\frac{\text{dist}(x, h)}{2\sigma^2}\right]}$

  **err$_{unl}$** $(h) = E_{x \sim D} \left[ e^{\left[-\frac{\text{dist}(x, h)}{2\sigma^2}\right]} \right]$

- **Illegal notion of compatibility**: the largest $\gamma$ s.t. D has probability mass exactly zero within distance $\gamma$ of h.
**Co-Training, Compatibility**

- **Co-training**: examples come as pairs \(<x_1, x_2>\) and the goal is to learn a pair of functions \(<h_1, h_2>\).
- **Hope** is that the two parts of the example are consistent.

- **Legal (and natural) notion of compatibility**:
  - the compatibility of \(<h_1, h_2>\) and \(D\):
    \[
    \Pr_{(x_1,x_2) \in D}[h_1(x_1) = h_2(x_2)]
    \]
  - can be written as an expectation over examples:
    \[
    \chi((h_1, h_2), (x_1, x_2)) = 1 \text{ if } h_1(x_1) = h_2(x_2)
    \]
    \[
    \chi((h_1, h_2), (x_1, x_2)) = 0 \text{ if } h_1(x_1) \neq h_2(x_2)
    \]

---

**Sample Complexity - Uniform convergence bounds**

**Finite Hypothesis Spaces, Doubly Realizable Case**

- Define \(C_{D,\chi}(\varepsilon) = \{ h \in C : err_{unl}(h) < \varepsilon \}\).

**Theorem**

If we see

\[
m_u \geq \frac{1}{\varepsilon} \left[ \ln |C| + \ln \frac{2}{\delta} \right]
\]

unlabeled examples and

\[
m_l \geq \frac{1}{\varepsilon} \left[ \ln |C_{D,\chi}(\varepsilon)| + \ln \frac{2}{\delta} \right]
\]

labeled examples, then with probability \( \geq 1 - \delta \), all \( h \in C \) with \( err(h) = 0 \) and \( err_{unl}(h) = 0 \) have \( err(h) \leq \varepsilon \).

- **Bound the # of labeled examples as a measure of the helpfulness of** \(D\) with respect to \(\chi\)
  - a helpful distribution is one in which \(C_{D,\chi}(\varepsilon)\) is small
**Example**

- Every variable is a positive indicator or negative indicator. No example has both kinds.
  - Algorithm: create graph on variables. Put an edge between two variables if any example has both of them.
  - Bad distribution: uniform over unit-vectors \( \{e_i\} \).
  - Good distribution:
    - Small number of connected components.
    - Both classes have good “expansion”.

**More Generally**

- Want algorithm that runs in poly time using samples poly in respective bounds.

- E.g., can think of:
  - \( \ln|C| \) as # bits to describe target without knowing \( D \),
  - \( \ln|C_{D,\epsilon}(\epsilon)| \) as number of bits to describe target knowing a good approx to \( D \), under assumption that target has low unlabeled error rate.

- Can get analogous sample-complexity bounds when target is not perfectly compatible.
**Infinite hypothesis spaces / VC-dimension**

**Infinite Hypothesis Spaces**
Assume $\chi(h, x)$ in $\{0,1\}$ and $\chi(C) = \{\chi_h : h \in C\}$ where $\chi_h(x) = \chi(h, x)$.

**Two issues:**
1. If we want uniform convergence of *unlabeled* error rates (all $h \in C$ have $|\bar{err}_{\text{unl}}(h) - err_{\text{unl}}(h)| \leq \epsilon$) then we need unlabeled sample size to be large as a function of VC-dimension of $\chi(C)$.
2. For “size” of highly-compatible set, the max number of ways of splitting $m$ points is not a good measure. Instead:

   \[ C[m, D]: \text{expected \# of splits of } m \text{ points from } D \text{ with concepts in } C. \]

---

**Theorem**

\[
m_u = O \left( \frac{VCdim(\chi(C))}{\epsilon^2} \log \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \log \frac{2}{\delta} \right)
\]

unlabeled examples and

\[
m_l > \frac{2}{\epsilon} \left[ \log(2s) + \log \frac{2}{\delta} \right]
\]

labeled examples, where

\[
s = C_{D,\chi}(t + 2\epsilon)[2m_l, D]
\]

are sufficient so that with probability at least $1 - \delta$, all $h \in C$ with $\bar{err}(h) = 0$ and $\bar{err}_{\text{unl}}(h) \leq t + \epsilon$ have $err(h) \leq \epsilon$, and furthermore all $h \in C$ have

\[
|err_{\text{unl}}(h) - \bar{err}_{\text{unl}}(h)| \leq \epsilon
\]

**Implication:** If $err_{\text{unl}}(c^*) \leq t$, then with probab. $\geq 1 - \delta$, the $h \in C$ that optimizes both $\bar{err}(h)$ and $\bar{err}_{\text{unl}}(h)$ has $err(h) \leq \epsilon$. 
\textbf{\(\epsilon\)-Cover-based bounds}

- For algorithms that behave in a specific way:
  - first use the unlabeled data to choose a representative set of compatible hypotheses
  - then use the labeled sample to choose among these

**Theorem**

If \( t \) is an upper bound for \( \text{err}_{\text{uni}}(c^*) \) and \( p \) is the size of a minimum \( \epsilon \) – cover for \( C_{D,X}(t + 4\epsilon) \), then using

\[
m_u = O\left(\frac{\text{VCdim}(\chi(C))}{\epsilon^2} \log \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \log \frac{2}{\delta}\right)
\]

unlabeled examples and

\[
m_l = O\left(\frac{1}{\epsilon} \ln \frac{p}{\delta}\right)
\]

labeled examples, we can with probab. \( \geq 1 - \delta \) identify a hypothesis which is 10\(\epsilon\) close to \( c^* \).

- Can result in much better bound than uniform convergence.

\[+\]
\[-\]

\textbf{\(\epsilon\)-Cover-based bounds}

- For algorithms that behave in a specific way:
  - first use the unlabeled data to choose a representative set of compatible hypotheses
  - then use the labeled sample to choose among these

\textit{E.g.}, in case of co-training linear separators with independence assumption:
  - \( \epsilon \)-cover of compatible set = \{0, 1, \( c^* \), \( \neg c^* \)\}

\textit{E.g.}, Transductive SVM when data is in two blobs.
Ways unlabeled data can help in this model

- If the target is highly compatible with D and have enough unlabeled data to estimate $\chi$ over all $h \in C$, then can reduce the search space (from $C$ down to just those $h \in C$ whose estimated unlabeled error rate is low).

- By providing an estimate of D, unlabeled data can allow a more refined distribution-specific notion of hypothesis space size (such as the size of the smallest $\varepsilon$-cover).

- If D is nice so that the set of compatible $h \in C$ has a small $\varepsilon$-cover and the elements of the cover are far apart, then can learn from even fewer labeled examples than the $1/\varepsilon$ needed just to verify a good hypothesis.

Some references