Advanced Data Science

Slides Adapted from Tom M. Mitchell
Co-training
[Blum & Mitchell ’98]

Different type of underlying regularity assumption:
Consistency or Agreement Between Parts
Co-training: Self-consistency

Agreement between two parts: co-training [Blum-Mitchell98].
- examples contain two sufficient sets of features, $x = h x_1, x_2$
- belief: the parts are consistent, i.e. $\exists c_1, c_2$ s.t. $c_1(x_1) = c_2(x_2) = c^*(x)$

For example, if we want to classify web pages: $x = h x_1, x_2$
as faculty member homepage or not
Iterative Co-Training

**Idea:** Use small labeled sample to learn initial rules.
- E.g., “my advisor” pointing to a page is a good indicator it is a faculty home page.
- E.g., “I am teaching” on a page is a good indicator it is a faculty home page.

**Idea:** Use unlabeled data to propagate learned information.
Iterative Co-Training

Idea: Use small labeled sample to learn initial rules.
  • E.g., “my advisor” pointing to a page is a good indicator it is a faculty home page.
  • E.g., “I am teaching” on a page is a good indicator it is a faculty home page.

Idea: Use unlabeled data to propagate learned information.

Look for unlabeled examples where one rule is confident and the other is not. Have it label the example for the other.

\[ h(x_1, x_2) \]

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

Works by using unlabeled data to propagate learned information.

- Have learning algos $A_1$, $A_2$ on each of the two views.
- Use labeled data to learn two initial hyp. $h_1$, $h_2$.

Repeat

- Look through unlabeled data to find examples where one of $h_i$ is confident but other is not.
- Have the confident $h_i$ label it for algorithm $A_{3-i}$. 
Original Application: Webpage classification

12 labeled examples, 1000 unlabeled

<table>
<thead>
<tr>
<th></th>
<th>Page-based</th>
<th>Hyperlink-based</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std. Supervised</td>
<td>12.9</td>
<td>12.4</td>
<td>11.1</td>
</tr>
<tr>
<td>Co-training</td>
<td>6.2</td>
<td>11.6</td>
<td>5.0</td>
</tr>
<tr>
<td>Just say neg</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
</tbody>
</table>

(same run)
Iterative Co-Training
A Simple Example: Learning Intervals

Use labeled data to learn $h_1^i$ and $h_2^i$

Use unlabeled data to bootstrap

Labeled examples
Unlabeled examples
Expansion, Examples: Learning Intervals

Consistency: zero probability mass in the regions

Non-expanding (non-helpful) distribution

Expanding distribution
Co-training: Theoretical Guarantees

What properties do we need for co-training to work well?

We need assumptions about:

1. the underlying data distribution
2. the learning algos on the two sides

[Blum & Mitchell, COLT ’98]

1. Independence given the label
2. Alg. for learning from random noise.

[Balcan, Blum, Yang, NIPS 2004]

1. Distributional expansion.
2. Alg. for learning from positive data only.
Co-training [BM’98]

Say that $h_1$ is a \textit{weakly-useful predictor} if

$$\Pr[h_1(x) = 1|c_1(x) = 1] > \Pr[h_1(x) = 1|c_1(x) = 0] + \gamma.$$ 

Has higher probability of saying positive on a true positive than it does on a true negative, by at least some gap $\gamma$.

Say we have enough labeled data to produce such a starting point.

\textbf{Theorem}: if $C$ is learnable from random classification noise, we can use a weakly-useful $h_1$ plus \textit{unlabeled} data to create a strong learner under independence given the label.
Co-training: Benefits in Principle

[BB’05]: Under independence given the label, any pair \( \langle h_1, h_2 \rangle \) with high agreement over unlabeled data must be close to:

- \( \langle c_1, c_2 \rangle, \langle \neg c_1, \neg c_2 \rangle, \langle \text{true, true} \rangle, \text{ or } \langle \text{false, false} \rangle \)
Co-training: Benefits in Principle

[BB’05]: Under independence given the label, any pair \( \langle h_1, h_2 \rangle \) with high agreement over unlabeled data must be close to:

- \( \langle c_1, c_2 \rangle \), \( \langle \neg c_1, \neg c_2 \rangle \), \( \langle \text{true, true} \rangle \), or \( \langle \text{false, false} \rangle \)

E.g.,

Because of independence, we will see a lot of disagreement...
Co-training/Multi-view SSL: Direct Optimization of Agreement

**Input:**

\[ S_l = \{ (x_{1l}, y_{1l}), \ldots, (x_{ml}, y_{ml}) \} \]

\[ S_u = \{ x_1, \ldots, x_m \} \]

\[ \arg\min_{h_1, h_2} \sum_{l=1}^{m_l} \sum_{i=1}^{m_{l_i}} l(h_1(x_i), y_i) + \sum_{l=1}^{m_u} \text{agreement}(h_1(x_i), h_2(x_i)) \]

- Each of them has small labeled error
- Regularizer to encourage agreement over unlabeled dat

E.g.,

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

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

\[
\begin{align*}
\arg\min_{h_1, h_2} & \sum_{l=1}^{2} \sum_{i=1}^{m_l} l(h_1(x_i), y_i) + C \sum_{i=1}^{m_u} \text{agreement}(h_1(x_i), h_2(x_i)) \\
\text{• } l(h(x_i), y_i) & \text{ loss function} \\
\text{  • E.g., square loss } & l(h(x_i), y_i) = (y_i - h(x_i))^2 \\
\text{  • E.g., 0/1 loss } & l(h(x_i), y_i) = 1_{y_i \neq h(x_i)}
\end{align*}
\]

E.g.,

Original Application: Webpage classification

12 labeled examples, 1000 unlabeled

<table>
<thead>
<tr>
<th></th>
<th>Page-based</th>
<th>Hyperlink-based</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std. Supervised</td>
<td>12.9</td>
<td>12.4</td>
<td>11.1</td>
</tr>
<tr>
<td>Co-training</td>
<td>6.2</td>
<td>11.6</td>
<td>5.0</td>
</tr>
<tr>
<td>Just say neg</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
</tbody>
</table>

(sample run)
Many Other Applications

E.g., [Levin-Viola-Freund03] identifying objects in images.
Two different kinds of preprocessing.

![Images of images and processed images]

E.g., [Collins&Singer99] named-entity extraction.
- “I arrived in London yesterday”
  ...

Central to NELL!!!
  ...

Similarity Based Regularity

[Blum&Chwala01], [ZhuGhahramaniLafferty03]
Graph-based Methods

- Assume we are given a pairwise similarity function and that very similar examples probably have the same label.

- If we have a lot of labeled data, this suggests a Nearest-Neighbor type of algorithm.

- If you have a lot of unlabeled data, perhaps can use them as “stepping stones”.

<table>
<thead>
<tr>
<th>E.g., handwritten digits [Zhu07]:</th>
<th>0 2</th>
<th>0 3 2 2 2 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>not similar</td>
<td>‘indirectly’ similar with stepping stones</td>
<td></td>
</tr>
</tbody>
</table>
Graph-based Methods

**Idea**: construct a graph with edges between very similar examples.

Unlabeled data can help "glue" the objects of the same class together.
Graph-based Methods

Idea: construct a graph with edges between very similar examples. Unlabeled data can help “glue” the objects of the same class together.

Often, transductive approach. (Given \( L + U \), output predictions on \( U \)). Are allowed to output any labeling of \( L \cup U \).

**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.

Several methods:
- Minimum/Multiway cut [Blum&Chawla01]
- Minimum “soft-cut” [ZhuGhahramaniLafferty'03]
- Spectral partitioning
- ...
What You Should Know

• Unlabeled data useful if we have beliefs not only about the form of the target, but also about its relationship with the underlying distribution.

• Different types of algorithms (based on different beliefs).
  - Transductive SVM [Joachims ‘99]
  - Co-training [Blum & Mitchell ‘98]
  - Graph-based methods [B&CO1], [ZGL03]
Additional Material on Graph Based Methods
Objective: Solve for labels on unlabeled points that minimize total weight of edges whose endpoints have different labels. (i.e., the total weight of bad edges)

• If just two labels, can be solved efficiently using max-flow min-cut algorithms
  - Create super-source $s$ connected by edges of weight $\infty$ to all + labeled pts.
  - Create super-sink $t$ connected by edges of weight $\infty$ to all – labeled pts.
  - Find minimum-weight $s$-$t$ cut
Objective: Solve for probability vector over labels $f_i$ on each unlabeled point $i$.

(labeled points get coordinate vectors in direction of their known label)

- Minimize $e = (i,j) \ w.e \ \|f_i - f_j\|^2$
  where $\|f_i - f_j\|$ is Euclidean distance.
- Can be done efficiently by solving a set of linear equations.
How to Create the Graph

• Empirically, the following works well:
  
  1. Compute distance between $i, j$
  
  2. For each $i$, connect to its kNN. $k$ very small but still connects the graph
  
  3. Optionally put weights on (only) those edges

$$\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

  4. Tune $\sigma$