Information Retrieval and Web Search

Cornelia Caragea

Computer Science

University of Illinois at Chicago

Credits for slides: Tom Mitchell, Avrim Blum

Semi-Supervised Learning

Semi-supervised Learning

- Supervised Learning models require labeled data
- Learning a reliable model usually requires plenty of labeled data
- Labeled Data: Expensive and Scarce
- Unlabeled Data: Abundant and Free/Cheap
  - E.g., webpage classification: easy to get unlabeled webpages
- **Semi-supervised Learning:** Devising ways of utilizing unlabeled data with labeled data to learn better models
  - Formally, given labeled training data $\mathcal{D}^l = \{\mathbf{x}_i, y_i\}_{i=1}^{L}$, and unlabeled data $\mathcal{D}^u = \{\mathbf{x}_i\}_{i=L+1}^{L+U}$ (usually $U \gg L$), the goal is to learn a classifier f better than using labeled data alone.

Why/How Might Unlabeled Data Help?

- At first consideration, one may think that nothing can be gained by having access to unlabeled data.
- However, they provide information about the joint probability distribution over words.
- Example: university webpage classification
  - Supposed that using only labeled data, documents containing "homework" belong to the "course" category.
  - If we estimate the classification of many unlabeled documents, we may find that "lecture" occurs frequently in unlabeled documents that are believed to belong to the "course" category.
  - The co-occurrence of "homework" and "lecture" over the large set of unlabeled data allows to construct a more accurate classifier that considers both "homework" and "lecture" as indicators of positive examples.
- **Assumption:** Examples from the same class follow a coherent distribution.

Using Expectation-Maximization for SSL

- Expectation-Maximization (EM) is a class of iterative algorithms for maximum likelihood or maximum a posteriori estimation in problems with incomplete data (Dempster, Laird, and Rubin, 1977)
- Unlabeled data are considered incomplete as they come without class labels.
- The EM algorithm:
  - First trains a classifier with only *labeled data* and uses the classifier to assign probabilistically-weighted class labels to each unlabeled example by calculating the expectation of the missing class labels.
  - It then trains a new classifier using all the documents and iterates.

Incorporating Unlabeled Data with EM

Applying EM to Naive Bayes.

- Inputs: Labeled data $\mathcal{D}^l = \{\mathbf{x}_i, y_i\}_{i=1}^{L}$, and unlabeled data $\mathcal{D}^u = \{\mathbf{x}_i\}_{i=L+1}^{L+U}$
- Train an initial naive Bayes classifier, $\hat{\theta}$, using just $\mathcal{D}^l$.
- Loop while classifier parameters improve, as measured by the change in the complete log probability of the labeled and unlabeled data and the prior:
  - (**E-step**) Use the current classifier, $\hat{\theta}$, to estimate component membership of each unlabeled example, $P(c_j|d_i;\hat{\theta})$.
  - (**M-step**) Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each example, $P(w_t|c_j;\hat{\theta})$ and $P(c_j|\hat{\theta})$
- **Output:** A classifier $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

Incorporating Unlabeled Data with EM

E-step:

$$
\begin{aligned}
\mathrm{P}(y_i = c_j | d_i; \hat{\theta}) &= \frac{\mathrm{P}(c_j|\hat{\theta})\mathrm{P}(d_i|c_j;\hat{\theta})}{\mathrm{P}(d_i|\hat{\theta})} \\
&= \frac{\mathrm{P}(c_j|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|} \mathrm{P}(c_r|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_r;\hat{\theta})}.
\end{aligned}
$$

M-step:

$$
\hat{\theta}_{w_t|c_j} \equiv \mathrm{P}(w_t|c_j;\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_t, d_i)\mathrm{P}(y_i = c_j|d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|\mathcal{D}|} N(w_s, d_i)\mathrm{P}(y_i = c_j|d_i)},
$$

$$
\hat{\theta}_{c_j} \equiv \mathrm{P}(c_j|\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(y_i = c_j|d_i)}{|\mathcal{C}| + |\mathcal{D}|}.
$$

EM with Unlabeled Data Increases Accuracy

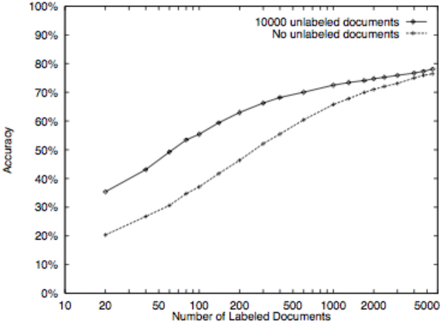


Figure 2. Classification accuracy on the **20 Newsgroups** data set, both with and without 10,000 unlabeled documents. With small amounts of training data, using EM yields more accurate classifiers. With large amounts of labeled training data, accurate parameter estimates can be obtained without the use of unlabeled data, and the two methods begin to converge.

EM with Unlabeled Data Increases Accuracy

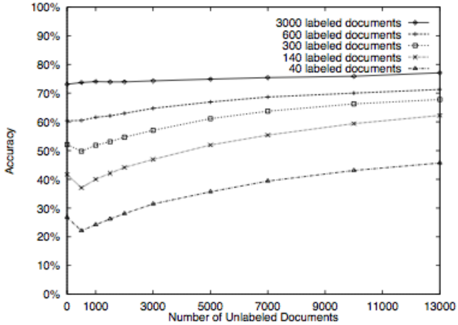


Figure 3. Classification accuracy while varying the number of unlabeled documents. The effect is shown on the 20 Newsgroups data set, with 5 different amounts of labeled documents, by varying the amount of unlabeled data on the horizontal axis. Having more unlabeled data helps. Note the dip in accuracy when a small amount of unlabeled data is added to a small amount of labeled data. We hypothesize that this is caused by extreme, almost 0 or 1, estimates of component membership, $P(c_j|d_i,\hat{\theta})$, for the unlabeled documents (as caused by naive Bayes' word independence assumption).

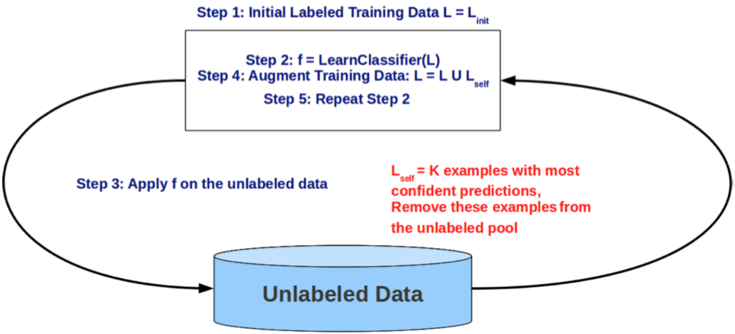
The Evolution of Naive Bayes over two EM iterations on WebKB data using 2500 unlabeled documents

Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^\star$	DD:DD
dist	DD:DD	due
identical	handout	$D^\star$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	DDam
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

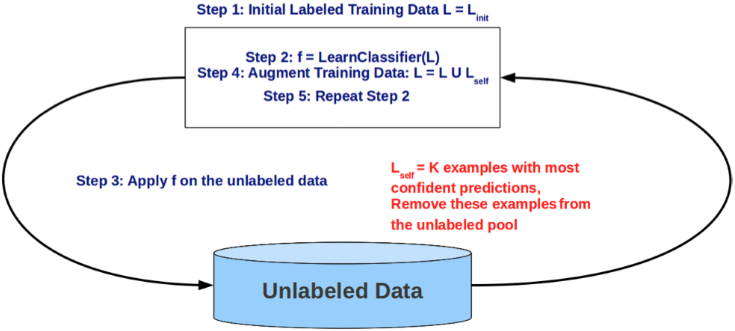
The Self-Training Approach to SSL

- **Given:** Small amount of initial labeled training data and large amount of unlabeled data
- **Idea:** Train, predict, re-train using your own (best) predictions, repeat



The Self-Training Approach to SSL

- **Given:** Small amount of initial labeled training data and large amount of unlabeled data
- **Idea:** Train, predict, re-train using your own (best) predictions, repeat



- Can be used with any supervised learner. Often works well in practice
- Caution: Prediction mistake can reinforce itself.

Co-Training Approach to SSL

- **Given:** Labeled data $\{\mathbf{x}_i, y_i\}_{i=1}^{L}$, unlabeled data $\{\mathbf{x}_i\}_{i=L+1}^{L+U}$
- Each example has 2 views: $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]$
- How do we get different views?
  - Naturally available (different types of features for the same object)
    - Webpages: view 1 from page text; view 2 from page URL
  - ... or by splitting the original features into two groups
- Assumption: Given sufficient data, each view is good enough to learn from
- Co-training: Utilize both views to learn better with fewer labeled examples
- **Idea:** Each view teaching (training) the other view
- Technical Condition: Views should be conditionally independent

Redundantly Predictive Features

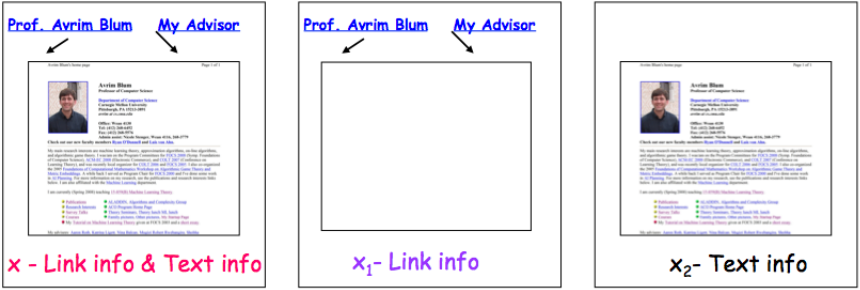
Assumption: Given sufficient data, either view is sufficient for learning

- There are f_1 and f_2 s.t. $f(x) = f_1(x) = f_2(x) = y$ for all (x, y) pairs.

Redundantly Predictive Features

Assumption: Given sufficient data, either view is sufficient for learning

- There are f_1 and f_2 s.t. $f(x) = f_1(x) = f_2(x) = y$ for all (x, y) pairs.



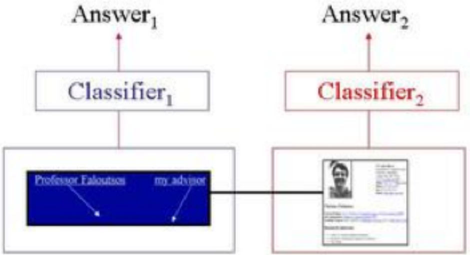
Co-Training

- Idea: Use small labeled sample to learn initial rules.
  - "my advisor" pointing to a page is a good indicator it is a faculty home page.
  - "I am teaching" on a page is a good indicator it is a faculty home page.

- Then look for unlabeled examples where one rule is confident and the other is not. Have it label the example for the other.
- Train 2 classifiers, one on each type of info. Use each to help train the other.
- Basic hope is that two views are consistent.

Co-Training

- Key idea: The classifiers C_1 and C_2 must:
  - Correctly classify labeled examples
  - Agree on classification of unlabeled.



Co-Training Algorithm #1

- Given: Labeled data L, unlabeled data U
- Loop:
  - Train f_1 (hyperlink classifier) using L
  - Train f_2 (page classifier) using L
  - Allow f_1 to label p positive, n negative examples from U
  - Allow f_2 to label p positive, n negative examples from U
  - Add these self-labeled examples to L.

Co-Training Algorithm #2

- Given: Labeled data L, unlabeled data U
- Create two labeled datasets L_1 and L_2 from L using views 1 and 2
- Learn classifiers f_1 from L_1 and f_2 from L_2
- Apply f_1 and f_2 on unlabeled data pool U to predict labels
    - Predictions are made only using their own set (view) of features
- Add k most confident predictions $(\mathbf{x}, f_1(\mathbf{x}))$ of f_1 to L_2
- Add k most confident predictions $(\mathbf{x}, f_2(\mathbf{x}))$ of f_2 to L_1
- Remove these examples from the unlabeled pool
- Re-train f_1 using L_1, f_2 using L_2
- Like self-training but two classifiers teaching each other
- Finally, use a voting or averaging to make predictions on the test data

Co-Training Results on WebKB

Training Naive Bayes classifiers on 12 labeled examples, 1000 unlabeled.

	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Table 2: Error rate in percent for classifying web pages as course home pages. The top row shows errors when training on only the labeled examples. Bottom row shows errors when co-training, using both labeled and unlabeled examples.