Partially Supervised Learning

- We have domains $\mathcal{X}$, $\mathcal{Y}$
- We have **labeled** examples (x_i, y_i) for $i = 1 \ldots n$
  (n is typically small)
- We have **unlabeled** examples (x_i) for $i = (n+1) \ldots (n+m)$
- Task is to learn a function $F : \mathcal{X} \rightarrow \mathcal{Y}$
- New questions:
  - Under what assumptions is unlabeled data "useful"?
  - Can we find NLP problems where these assumptions hold?
  - Which algorithms are suggested by the theory?

Named Entity Classification

- Classify entities as organizations, people or locations

Steptoe & Johnson	=	Organization
Mrs. Frank	=	Person
Honduras	=	Location

- Need to learn (weighted) rules such as

contains(Mrs.)	$\Rightarrow$	Person
full-string=Honduras	$\Rightarrow$	Location
context=company	$\Rightarrow$	Organization

An Approach Using Minimal Supervision

- Assume a small set of "seed" rules

contains(Incorporated)	$\Rightarrow$	Organization
full-string=Microsoft	$\Rightarrow$	Organization
full-string=I.B.M.	$\Rightarrow$	Organization
contains(Mr.)	$\Rightarrow$	Person
full-string=New_York	$\Rightarrow$	Location
full-string=California	$\Rightarrow$	Location
full-string=U.S.	$\Rightarrow$	Location

- Assume a large amount of unlabeled data

  .., says Mr. Cooper, a vice president of ...

- Methods gain leverage from redundancy:

  Either Spelling or Context alone is often sufficient to determine an entity's type

Cotraining

- We have domains $\mathcal{X}$, $\mathcal{Y}$
- We have **labeled** examples (x_i, y_i) for $i = 1 \ldots n$
- We have **unlabeled** examples (x_i) for $i = (n+1) \ldots (n+m)$
- We assume each example x_i splits into two views, x_{1i} and x_{2i}
- e.g., if x_i is a feature vector in $\mathbb{R}^{2d}$, then x_{1i} and x_{2i} are representations in $\mathbb{R}^d$.

The Data

- Approx 90,000 spelling/context pairs collected
- Two types of contexts identified by a parser

1. Appositives

   .., says Mr. Cooper, a vice president of ...

2. Prepositional Phrases

   Robert Haft , president of the Dart Group Corporation ...

Features: Two Views of Each Example

.., says Mr. Cooper, a vice president of ...

$\Downarrow$

Spelling Features	Contextual Features
Full-String = Mr. Cooper	appositive = president
Contains(Mr.)	
Contains(Cooper)	

Two Assumptions Behind Cotraining

Assumption 1: Either view is sufficient for learning

There are functions F_1 and F_2 such that

$$F(x) = F_1(x_1) = F_2(x_2) = y$$

for all (x, y) pairs

Examples of Problems with Two Natural Views

- Named entity classification (spelling vs. context)

- Web page classification [Blum and Mitchell, 1998]
  One view = words on the page, other view is pages linking to a page

- Word sense disambiguation: a random split of the text?

A Key Property: Redundancy

The ocean reflects the color of the sky, but even on cloudless days the color of the ocean is not a consistent blue. Phytoplankton, microscopic plant life that floats freely in the lighted surface waters, may alter the color of the water. When a great number of organisms are concentrated in an area, the plankton changes the color of the ocean surface. This is called a 'bloom.'

$\Downarrow$

w_{-1} = Phytoplankton	word-within-k = ocean
w_{+1} = life	word-within-k = reflects
w_{-2}, w_{-1} = (Phytoplankton,microscopic)	word-within-k = bloom
w_{-1}, w_{+1} = (microscopic,life)	word-within-k = color
w_{+1}, w_{+2} = (life,that)	$\ldots$

There are often many features which indicate the sense of the word

Two Assumptions Behind Cotraining

Assumption 2:

Some notion of independence between the two views

e.g., The **Conditional-independence-given-label** assumption:
If $D(x_1, x_2, y)$ is the distribution over examples, then

$$D(x_1, x_2, y) = D_0(y) D_1(x_1 \mid y) D_2(x_2 \mid y)$$

for some distributions D_0, D_1 and D_2

Why are these Assumptions Useful?

- Two examples/scenarios:
  - Rote learning, and a graph interpretation
  - Constraints on hypothesis spaces

Rote Learning, and a Graph Interpretation

- In a rote learner, functions F_1 and F_2 are look-up tables

Spelling	Category
Robert-Jordan	PERSON
Washington	LOCATION
Washington	LOCATION
Jamie-Gorelick	PERSON
Jerry-Jasinowski	PERSON
PacifiCorp	COMPANY
...	...

Context	Category
partner	PERSON
partner-at	COMPANY
law-in	LOCATION
firm-in	LOCATION
partner	PERSON
partner-of	COMPANY
...	...

- Note: this can be a very inefficient learning method
  (no chance to learn generalizations such as "any name containing *Mr.* is a person")

Rote Learning, and a Graph Interpretation

- Each node in the graph is a spelling or context
  A node for *Robert Jordan*, *Washington*, *law-in*, *partner* etc.

- Each (x_{1i}, x_{2i}) pair is an edge in the graph
  e.g., (Robert Jordan, partner)

- An edge between two nodes mean they have **the same label**
  (relies on assumption 1: each view is sufficient for classification)

- As quantity of unlabeled data increases, graph becomes more connected
  (relies on assumption 2: some independence between the two views)

Constraints on Hypothesis Spaces

- New case: n training examples (x_{1i}, x_{2i}, y_i) for $i = 1 \ldots n$, m unlabeled examples (x_{1i}, x_{2i}) for $i = (n+1) \ldots (n+m)$

- We assume a distribution $D(x_1, x_2, y)$ over training/test examples

- We have hypothesis spaces $\mathcal{H}_1$ and $\mathcal{H}_2$

- With labeled data alone, if n is number of training examples, then $\frac{\log |\mathcal{H}_1|}{2n}$ must be small

- With additional unlabeled data, we can consider the restricted hypothesis space

$$\mathcal{H}'_1 = \{F_1 : \quad F_1 \in \mathcal{H}_1, \exists F_2 \in \mathcal{H}_2 \text{ s.t. } F_1(x_{1i}) = F_2(x_{2i}) \text{ for } i = (n+1)\dots(n+m)\}$$

  i.e., we only consider functions F_1 which agree with at least one F_2 on all unlabeled examples

- Basic idea: we don't know the label for an unlabeled example, **but we do know that the two functions must agree on it**

- Now, we need $\frac{\log|\mathcal{H}'_1|}{2n}$ to be small
  if $|\mathcal{H}'_1| << |\mathcal{H}_1|$ then we need fewer training examples

Cotraining Summary

- $n + m$ training examples $x_i = (x_{1i}, x_{2i})$
- First n examples have labels y_i
- Learn functions F_1 and F_2 such that

$$F_1(x_{1i}) = F_2(x_{2i}) = y_i \qquad i = 1 \ldots n$$

$$F_1(x_{1i}) = F_2(x_{2i}) \qquad i = n + 1 \ldots n + m$$

A Linear Model

- How to build a classifier from spelling features alone?
  A linear model:
  - $\mathbf{GEN}(x_1)$ is possible labels $\{person, location, organization\}$
  - $\mathbf{\Phi}(x_1, y)$ is a set of features on spelling/label pairs, e.g.,

$$\mathbf{\Phi}_{100}(x_1, y) = \begin{cases} 1 & \text{if } x_1 \text{ contains } \textit{Mr.}\text{, and } y = \textit{person} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{\Phi}_{101}(x_1, y) = \begin{cases} 1 & \text{if } x_1 \text{ is } \textit{IBM}\text{, and } y = \textit{person} \\ 0 & \text{otherwise} \end{cases}$$

  - $\mathbf{W}$ is parameter vector, as usual choose

$$F_1(x_1, \mathbf{W}) = \arg \max_{y \in \mathbf{GEN}(x_1)} \mathbf{\Phi}(x_1, y) \cdot \mathbf{W}$$

  - $\Rightarrow$ each parameter in $\mathbf{W}$ gives a weight for a feature/label pair.
    e.g., $\mathbf{W}_{100} = 2.5$, $\mathbf{W}_{101} = -1.3$

A Boosting Approach to Supervised Learning

- Greedily minimize

$$L(\mathbf{W}) = \sum_i \sum_{y \neq y_i} e^{-\mathbf{m}(y_i, y, \mathbf{W})}$$

where

$$\mathbf{m}(y_i, y, \mathbf{W}) = \mathbf{\Phi}(x_i, y_i) \cdot \mathbf{W} - \mathbf{\Phi}(x_i, y) \cdot \mathbf{W}$$

- $L(\mathbf{W})$ is an upper bound on the number of ranking errors,

$$L(\mathbf{W}) \geq \sum_i \sum_{y \neq y_i} [[\mathbf{m}(y_i, y, \mathbf{W}) \leq 0]]$$

An Extension to the Cotraining Scenario

- Now build **two** linear models in parallel
  - $\mathbf{GEN}(x_1) = \mathbf{GEN}(x_2)$ is set of possible labels $\{person, location, organization\}$
  - $\mathbf{\Phi}^1(x_1, y)$ is a set of features on spelling/label pairs
  - $\mathbf{\Phi}^2(x_2, y)$ is a set of features on context/label pairs, e.g.,

$$\mathbf{\Phi}^2{}_{100}(x_2, y) = \begin{cases} 1 & \text{if } x_2 \text{ is } \textit{president} \text{ and } y = \textit{person} \\ 0 & \text{otherwise} \end{cases}$$

  - $\mathbf{W}^1$ and $\mathbf{W}^2$ are the two parameter vectors

$$F_1(x_1, \mathbf{W}^1) = \arg \max_{y \in \mathbf{GEN}(x_1)} \mathbf{\Phi}^1(x_1, y) \cdot \mathbf{W}^1$$

$$F_2(x_2, \mathbf{W}^2) = \arg \max_{y \in \mathbf{GEN}(x_2)} \mathbf{\Phi}^2(x_2, y) \cdot \mathbf{W}^2$$

An Extension to the Cotraining Scenario

- $n + m$ training examples $x_i = (x_{1i}, x_{2i})$
- First n examples have labels y_i
- Linear models define F_1 and F_2 as

$$F_1(x_1, \mathbf{W}^1) = \arg \max_{y \in \mathbf{GEN}(x_1)} \mathbf{\Phi}^1(x_1, y) \cdot \mathbf{W}^1$$

$$F_2(x_2, \mathbf{W}^2) = \arg \max_{y \in \mathbf{GEN}(x_2)} \mathbf{\Phi}^2(x_2, y) \cdot \mathbf{W}^2$$

- Three types of errors:

$$\begin{aligned} E_1 &= \sum_{i=1}^{n} [[F_1(x_{1i}, \mathbf{W}^1) \neq y_i]] \\ E_2 &= \sum_{i=1}^{n} [[F_2(x_{2i}, \mathbf{W}^2) \neq y_i]] \\ E_3 &= \sum_{i=n+1}^{m+1} [[F_1(x_{1i}, \mathbf{W}^1) \neq F_2(x_{2i}, \mathbf{W}^2)]] \end{aligned}$$

Objective Functions for Cotraining

- Define "pseudo labels"

$$z_{1i}(\mathbf{W}^1) = f_1(x_{1i}, \mathbf{W}^1) \qquad i = (n+1)\dots(n+m)$$

$$z_{2i}(\mathbf{W}^2) = f_2(x_{2i}, \mathbf{W}^2) \qquad i = (n+1)\dots(n+m)$$

e.g., z_{1i} is output of first classifier on the i'th example

$$\begin{aligned}
L(\mathbf{W}^1, \mathbf{W}^2) \quad = \quad & \sum_{i=1}^{n} \sum_{y \neq y_i} e^{\mathbf{\Phi}^1(x_{1i}, y)\cdot\mathbf{W}^1 - \mathbf{\Phi}^1(x_{1i}, y_i)\cdot\mathbf{W}^1} \\
& + \sum_{i=1}^{n} \sum_{y \neq y_i} e^{\mathbf{\Phi}^2(x_{2i}, y)\cdot\mathbf{W}^2 - \mathbf{\Phi}^2(x_{2i}, y_i)\cdot\mathbf{W}^2} \\
& + \sum_{i=n+1}^{n+m} \sum_{y \neq z_{2i}} e^{\mathbf{\Phi}^1(x_{1i}, y)\cdot\mathbf{W}^1 - \mathbf{\Phi}^1(x_{1i}, z_{2i})\cdot\mathbf{W}^1} \\
& + \sum_{i=n+1}^{n+m} \sum_{y \neq z_{1i}} e^{\mathbf{\Phi}^2(x_{2i}, y)\cdot\mathbf{W}^2 - \mathbf{\Phi}^2(x_{2i}, z_{2i})\cdot\mathbf{W}^2}
\end{aligned}$$

More Intuition

- Need to minimize $L(\mathbf{W}^1, \mathbf{W}^2)$, do this by greedily minimizing w.r.t. first $\mathbf{W}^1$, then $\mathbf{W}^2$
- Algorithm boils down to:

1. Start with labeled data alone
2. Induce a contextual feature for each class (person/location/organization) from the current set of labelled data
3. Label unlabeled examples using contextual rules
4. Induce a spelling feature for each class (person/location/organization) from the current set of labelled data
5. Label unlabeled examples using spelling rules
6. Return to step 2

Optimization Method

1. Set pseudo labels z_{2i}
2. Update $\mathbf{W}^1$ to minimize

$$\sum_{i=1}^{n} \sum_{y \neq y_i} e^{\mathbf{\Phi}^1(x_{1i}, y) \cdot \mathbf{W}^1 - \mathbf{\Phi}^1(x_{1i}, y_i) \cdot \mathbf{W}^1}$$

$$+ \sum_{i=n+1}^{n+m} \sum_{y \neq z_{2i}} e^{\mathbf{\Phi}^1(x_{1i}, y) \cdot \mathbf{W}^1 - \mathbf{\Phi}^1(x_{1i}, z_{2i}) \cdot \mathbf{W}^1}$$

(for each class choose a spelling feature, weight)

3. Set pseudo labels z_{1i}

4. Update $\mathbf{W}^2$ to minimize

$$\sum_{i=1}^{n} \sum_{y \neq y_i} e^{\boldsymbol{\Phi}^2(x_{2i}, y) \cdot \mathbf{W}^2 - \boldsymbol{\Phi}^2(x_{2i}, y_i) \cdot \mathbf{W}^2}$$

$$+ \sum_{i=n+1}^{n+m} \sum_{y \neq z_{1i}} e^{\boldsymbol{\Phi}^2(x_{2i}, y) \cdot \mathbf{W}^2 - \boldsymbol{\Phi}^2(x_{2i}, z_{2i}) \cdot \mathbf{W}^2}$$

(for each class choose a contextual feature, weight)

5. Return to step 1

An Example Trace

1. Use seeds to label 8593 examples
   (4160 companies, 2788 people, 1645 locations)

2. Pick a contextual feature for each class:

COMPANY:	preposition=unit of	2.386	274/2
PERSON:	appositive=president	1.593	120/6
LOCATION:	preposition=Company of	1.673	46/1

3. Set pseudo labels using seeds + contextual features
   (5319 companies, 6811 people, 1961 locations)

4. Pick a spelling feature for each class

COMPANY:	Contains(Corporation)	2.475	495/10
PERSON:	Contains(.)	2.482	4229/106
LOCATION:	fullstring=America	2.311	91/0

5. Set pseudo labels using seeds + spelling features
   (7180 companies, 8161 people, 1911 locations)

6. Continue ...

Evaluation

- 88,962 $(spelling, context)$ pairs extracted as training data

- 7 seed rules used

contains(Incorporated)	$\Rightarrow$	Organization
full-string=Microsoft	$\Rightarrow$	Organization
full-string=I.B.M.	$\Rightarrow$	Organization
contains(Mr.)	$\Rightarrow$	Person
full-string=New_York	$\Rightarrow$	Location
full-string=California	$\Rightarrow$	Location
full-string=U.S.	$\Rightarrow$	Location

- 1,000 examples picked at random, and labelled by hand to give a test set.

- Around 9% of examples were “noise”, not falling into any of the three categories
- Two measures given: one excluding all noise items, the other counting noise items as errors

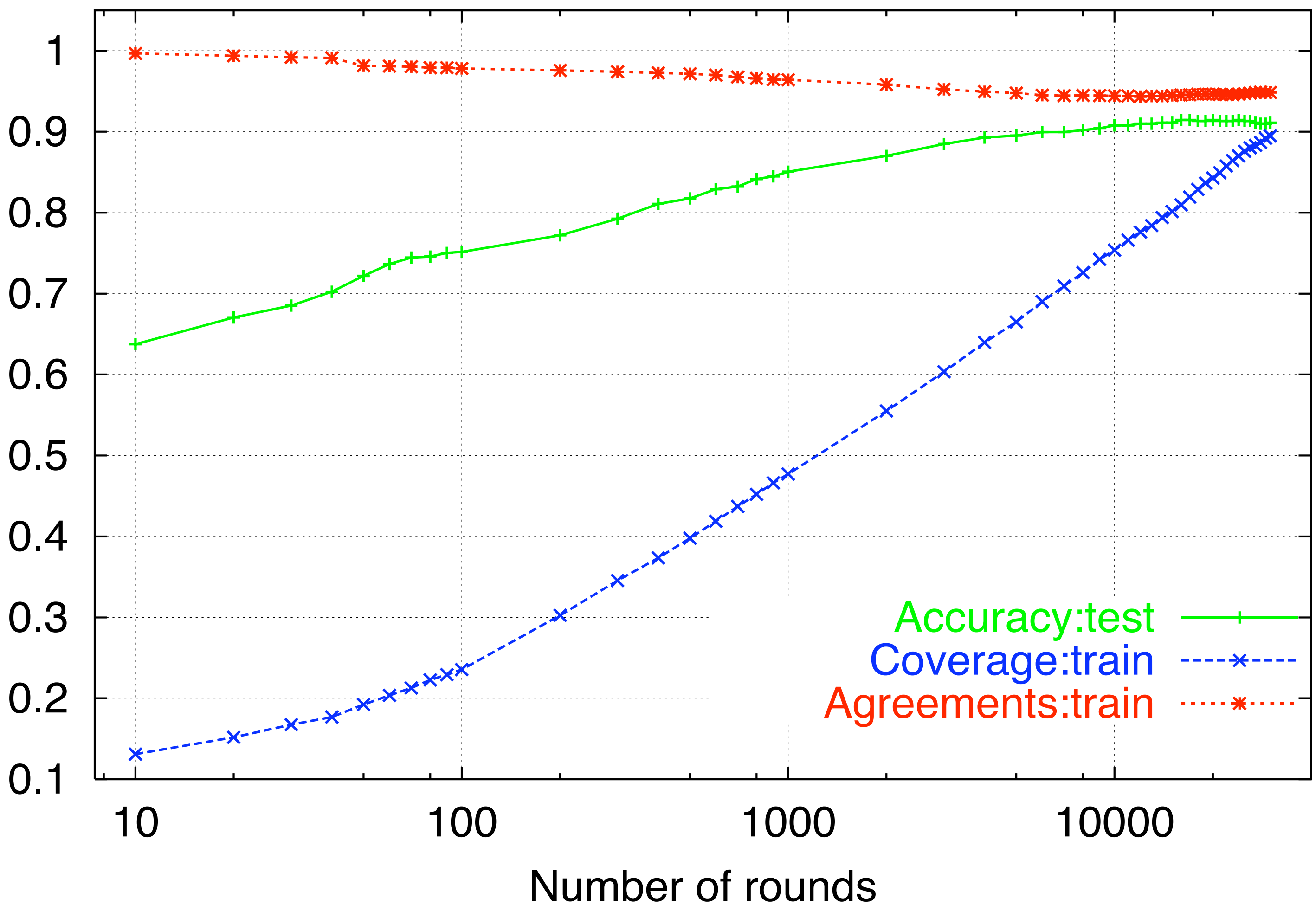
Other Methods

- EM approach
- Decision list (Yarowsky 95)
- Decision list 2 (modification of Yarowsky 95)
- DL-Cotrain:
  decision list alternating between two feature types

Results

Learning Algorithm	Accuracy (Clean)	Accuracy (Noise)
Baseline	45.8%	41.8%
EM	83.1%	75.8%
Decision List	81.3%	74.1%
Decision List 2	91.2%	83.2%
DL-CoTrain	91.3%	83.3%
CoBoost	91.1%	83.1%

Learning Curves for Coboosting



Summary

- Appears to be a complex task: many features/rules required
- With unlabeled data, supervision is reduced to 7 “seed” rules
- Key is **redundancy** in the data
- Cotraining suggests training two classifiers that “**agree**” as much as possible on unlabeled examples
- **CoBoost** algorithm builds two additive models in parallel, with an objective function that bounds the rate of agreement