Learning from Labeled and Unlabeled Data

Tom Mitchell

Statistical Approaches to Learning and Discovery, 10-702 and 15-802

March 31, 2003

When can Unlabeled Data help supervised learning?

Important question! In many cases, unlabeled data is plentiful, labeled data expensive

- Medical outcomes (x=<patient,treatment>, y=outcome)
- Text classification (x=document, y=relevance)
- User modeling (x=user actions, y=user intent)
- …

When can Unlabeled Data help supervised learning?

Consider setting:

- Set *X* of instances drawn from unknown *P(X)*
- *f: X→Y* target function (or, P(Y|X))
- Set *H* of possible hypotheses for *f*

Given:

- iid labeled examples $L = \{\langle x_1, y_1\rangle \ldots \langle x_m, y_m\rangle\}$
- iid unlabeld examples $U = \{x_{m+1}, \ldots x_{m+n}\}$

Determine:

$$\hat{f} \leftarrow \arg\min_{h \in H} \Pr_{x \in P(X)} [h(x) \neq f(x)]$$

Four Ways to Use Unlabeled Data for Supervised Learning

1. Use to reweight labeled examples
2. Use to help EM learn class-specific generative models
3. If problem has redundantly sufficient features, use CoTraining
4. Use to detect/preempt overfitting

1. Use U to reweight labeled examples

Can use $U \rightarrow \hat{P}(X)$ to alter optimization problem

- Wish to find

$$\hat{f} \leftarrow \operatorname*{argmin}_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) P(x)$$

- Often approximate as

$$\hat{f} \leftarrow \operatorname*{argmin}_{h \in H} \frac{1}{|L|} \sum_{\langle x,y \rangle \in L} \delta(h(x) \neq y)$$

$$\hat{f} \leftarrow \operatorname*{argmin}_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) \frac{n(x,L)}{|L|}$$

- Can use U for improved approximation:

$$\hat{f} \leftarrow \operatorname*{argmin}_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) \frac{n(x,L) + n(x,U)}{|L| + |U|}$$

- **Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.
- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents, $\mathcal{D}^l$, only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta P(\mathcal{D}|\theta)P(\theta)$ (see Equations 5 and 6).
- Loop while classifier parameters improve, as measured by the change in $l_c(\theta|\mathcal{D}; \mathbf{z})$ (the complete log probability of the labeled and unlabeled data, and the prior) (see Equation 10):
  - **(E-step)** Use the current classifier, $\hat{\theta}$, to estimate component membership of each unlabeled document, *i.e.*, the probability that each mixture component (and class) generated each document, $P(c_j|d_i;\hat{\theta})$ (see Equation 7).
  - **(M-step)** Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg\max_\theta P(\mathcal{D}|\theta)P(\theta)$ (see Equations 5 and 6).
- **Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

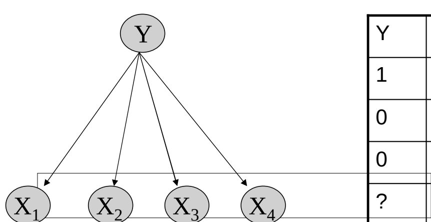
Table 1. The basic EM algorithm described in Section 5.1.

From [Nigam et al., 2000]

---

2. Use U with EM and Assumed Generative Model

Learn P(Y|X)



Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	1	1	0
?	0	0	0	1
?	0	1	0	1

E Step:

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

M Step:

w_t is t-th word in vocabulary

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

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

Elaboration 1: Downweight the influence of unlabeled examples by factor λ

$$l_c(\theta|\mathcal{D};\mathbf{z}) = \log(\mathrm{P}(\theta)) + \sum_{d_i \in \mathcal{D}^l} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right)$$
$$+ \lambda \left(\sum_{d_i \in \mathcal{D}^u} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right) \right).$$

Chosen by cross validation

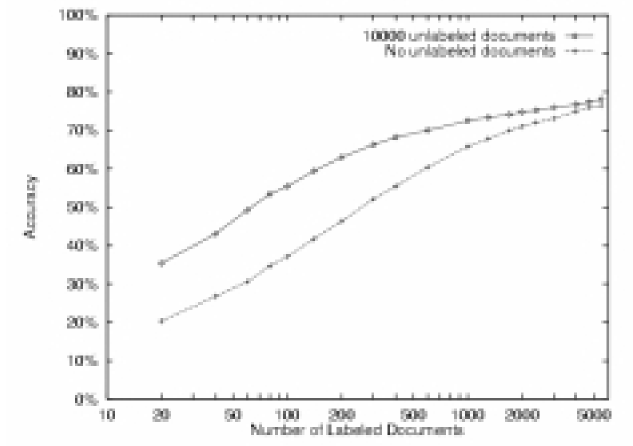
New M step:

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

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

$$\Lambda(i) = \begin{cases} \lambda & \text{if } d_i \in \mathcal{D}^u \\ 1 & \text{if } d_i \in \mathcal{D}^l. \end{cases}$$

20 Newsgroups



Experimental Evaluation

- Newsgroup postings
  - 20 newsgroups, 1000/group
- Web page classification
  - student, faculty, course, project
  - 4199 web pages
- Reuters newswire articles
  - 12,902 articles
  - 90 topics categories

20 Newsgroups

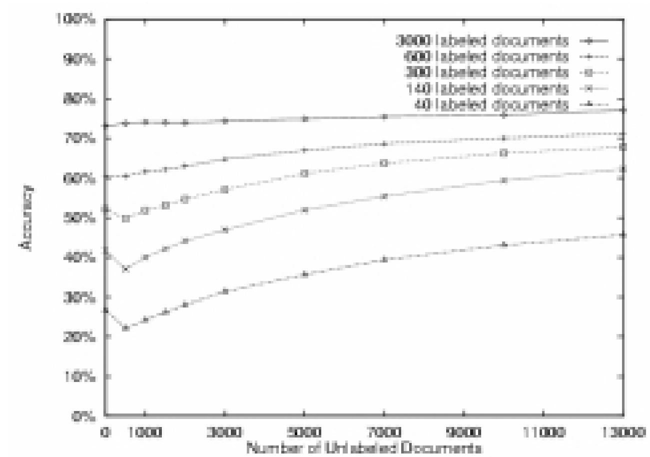


Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	D^*	DD:DD
dist	DD:DD	due
identical	handout	D^*
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

Using one labeled example per class

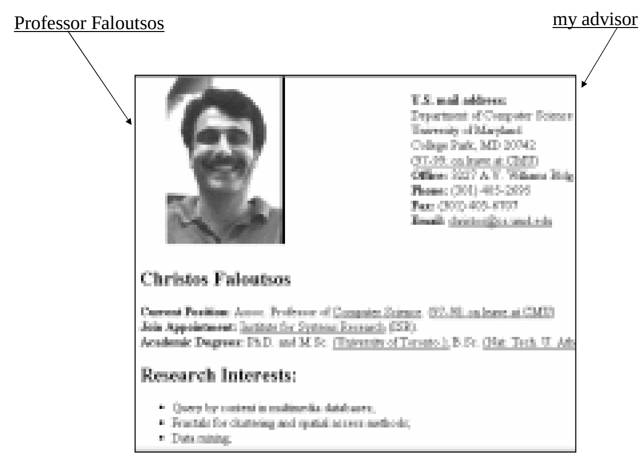
3. If Problem Setting Provides Redundantly Sufficient Features, use CoTraining

$$learn \quad f : X \rightarrow Y$$
$$where \quad X = X_1 \times X_2$$
$$where \quad x \quad drawn \quad from \quad unknown \quad distribution$$
$$and \quad \exists g_1, g_2 \quad (\forall x) g_1(x_1) = g_2(x_2) = f(x)$$

2. Use U with EM and Assumed Generative Model

- Can't really get something for nothing…
- But unlabeled data useful to degree that assumed form for P(X,Y) is correct
- E.g., in text classification, useful despite obvious error in assumed form of P(X,Y)

Redundantly Sufficient Features



CoTraining Algorithm #1

[Blum&Mitchell, 1998]

Given: labeled data L,

unlabeled data U

Loop:

Train g1 (hyperlink classifier) using L

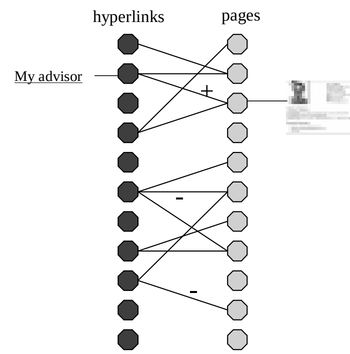
Train g2 (page classifier) using L

Allow g1 to label *p* positive, *n* negative examps from U

Allow g2 to label *p* positive, *n* negative examps from U

Add these self-labeled examples to L

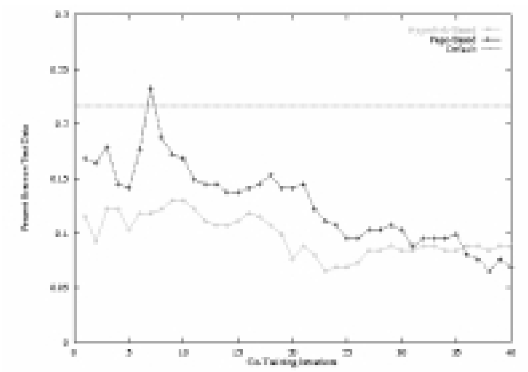
Co-Training Rote Learner



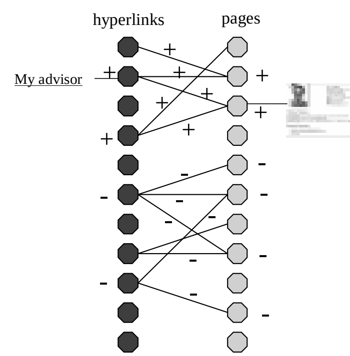
CoTraining: Experimental Results

- begin with 12 labeled web pages (academic course)
- provide 1,000 additional unlabeled web pages
- average error: learning from labeled data 11.1%;
- average error: cotraining 5.0%

Typical run:



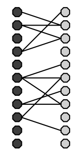
Co-Training Rote Learner



Expected Rote CoTraining error given *m* examples

$$\begin{aligned} &CoTraining \quad setting: \\ &learn \quad f: X \rightarrow Y \\ &where \quad X = X_1 \times X_2 \\ &where \quad x \quad drawn \quad from \quad unknown \quad distribution \\ &and \quad \exists g_1, g_2 \quad (\forall x) g_1(x_1) = g_2(x_2) = f(x) \end{aligned}$$

$$E[error] = \sum_j P(x \in g_j)(1 - P(x \in g_j))^m$$



Where g_j is the jth connected component of graph

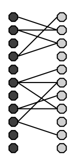
CoTraining Setting

$$\begin{aligned} &learn \quad f: X \rightarrow Y \\ &where \quad X = X_1 \times X_2 \\ &where \quad x \quad drawn \quad from \quad unknown \quad distribution \\ &and \quad \exists g_1, g_2 \quad (\forall x) g_1(x_1) = g_2(x_2) = f(x) \end{aligned}$$

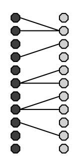
- If
  - x1, x2 conditionally independent given y
  - f is PAC learnable from noisy *labeled* data
- Then
  - f is PAC learnable from weak initial classifier plus *unlabeled* data

How many *unlabeled* examples suffice?

Want to assure that connected components in the underlying distribution, G_D, are connected components in the observed sample, G_S



G_D



G_S

O(log(N)/α) examples assure that with high probability, G_S has same connected components as G_D [Karger, 94]

N is size of G_D, α is min cut over all connected components of G_D

PAC Generalization Bounds on CoTraining

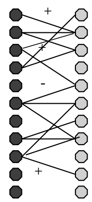
[Dasgupta et al., NIPS 2001]

Theorem 1 *With probability at least $1-\delta$ over the choice of the sample S, we have that for all h_1 and h_2, if $\gamma_i(h_1, h_2, \delta) > 0$ for $1 \leq i \leq k$ then (a) f is a permutation and (b) for all $1 \leq i \leq k$,*

$$P(h_1 \neq i \mid f(y) = i, h_1 \neq \bot) \leq \frac{\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \bot) + \epsilon_i(h_1, h_2, \delta)}{\gamma_i(h_1, h_2, \delta)}.$$

The theorem states, in essence, that if the sample size is large, and h_1 and h_2 largely agree on the unlabeled data, then $\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \bot)$ is a good estimate of the error rate $P(h_1 \neq i \mid f(y) = i, h_1 \neq \bot)$.

What if CoTraining Assumption Not Perfectly Satisfied?



- Idea: Want classifiers that produce a *maximally consistent* labeling of the data
- If learning is an optimization problem, what function should we optimize?

What Function Approximators?

$$\hat{g}_1(x) = \frac{1}{1+e^{\sum_j w_{j,1} x_j}} \qquad \hat{g}_2(x) = \frac{1}{1+e^{\sum_j w_{j,2} x_j}}$$

- Same fn form as Naïve Bayes, Max Entropy
- Use gradient descent to simultaneously learn g1 and g2, directly minimizing E = E1 + E2 + E3 + E4
- No word independence assumption, use both labeled and unlabeled data

What Objective Function?

$$E = E1 + E2 + c_3 E3 + c_4 E4$$

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

$$E3 = \sum_{x\in U} (\hat{g}_1(x_1) - \hat{g}_2(x_2))^2$$

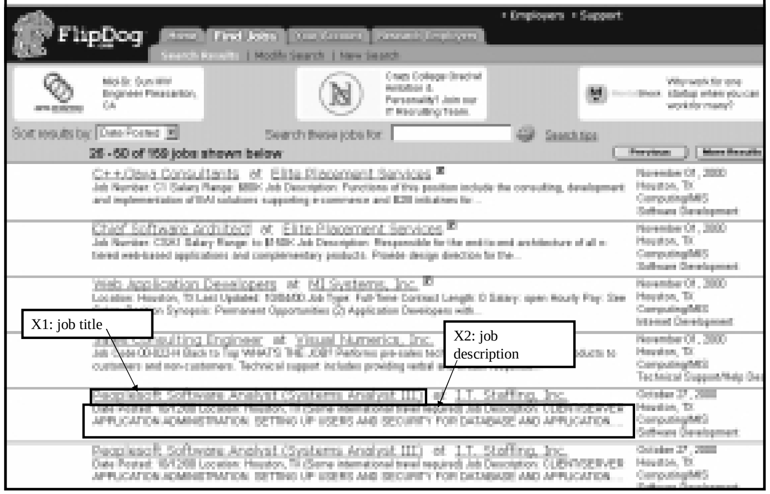
$$E4 = \left(\left(\frac{1}{|L|} \sum_{<x,y>\in L} y \right) - \left(\frac{1}{|L|+|U|} \sum_{x\in L\cup U} \frac{\hat{g}_1(x_1) + \hat{g}_2(x_2)}{2} \right) \right)^2$$

Error on labeled examples

Disagreement over unlabeled

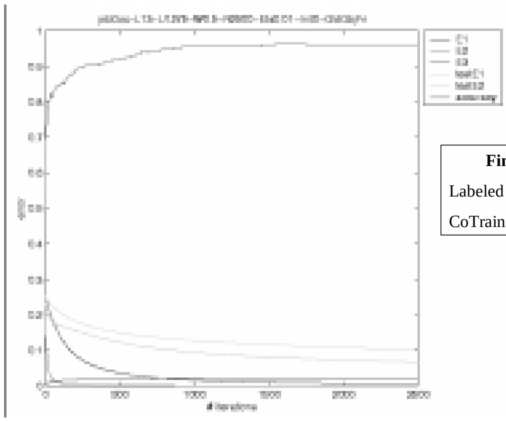
Misfit to estimated class priors

Classifying Jobs for FlipDog



Gradient CoTraining

Classifying FlipDog job descriptions: SysAdmin vs. WebProgrammer



Final Accuracy

Labeled data alone: 86%

CoTraining: 96%

CoTraining Summary

- Unlabeled data improves supervised learning when example features are redundantly sufficient
  - Family of algorithms that train multiple classifiers
- Theoretical results
  - Expected error for rote learning
  - If X1,X2 conditionally indep given Y
    - PAC learnable from weak initial classifier plus unlabeled data
    - error bounds in terms of disagreement between g1(x1) and g2(x2)
- Many real-world problems of this type
  - Semantic lexicon generation [Riloff, Jones 99], [Collins, Singer 99]
  - Web page classification [Blum, Mitchell 98]
  - Word sense disambiguation [Yarowsky 95]
  - Speech recognition [de Sa, Ballard 98]

Gradient CoTraining

Classifying Upper Case sequences as Person Names

Error Rates

	25 labeled 5000 unlabeled	*2300 labeled 5000 unlabeled*
Using labeled data only	.24	.13
Cotraining	.15 *	.11 *
Cotraining without fitting class priors (E4)	.27 *	

* sensitive to weights of error terms E3 and E4

4. Use U to Detect/Preempt Overfitting

Define *metric* over $H \cup \{f\}$

$$d(h_1, h_2) \equiv \int \delta(h_1(x) \neq h_2(x)) p(x) dx$$

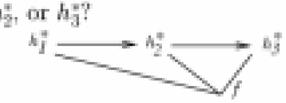
$$\hat{d}(h_1, f) = \frac{1}{|L|} \sum_{x_i \in L} \delta(h_1(x_i) \neq y_i)$$

$$\hat{d}(h_1, h_2) = \frac{1}{|U|} \sum_{x \in U} \delta(h_1(x) \neq h_2(x))$$

Organize H into complexity classes, sorted by $P(h)$

Let h_i^* be hypothesis with lowest $\hat{d}(h, f)$ in H_i

Prefer h_1^*, h_2^*, or h_3^*?



- Definition of distance metric
  - Non-negative *d(f,g)¸0;*
  - symmetric *d(f,g)=d(g,f)*;
  - triangle inequality *d(f,g) · d(f,h)+d(h,g)*

- Classification with zero-one loss:

$$d(h_1, h_2) \equiv \int \delta(h_1(x) \neq h_2(x)) p(x) dx$$

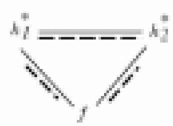
- Regression with squared loss:

$$d(h_1, h_2) \equiv \sqrt{\int (h_1(x) - h_2(x))^2 p(x) dx}$$

Procedure TRI

- Given hypothesis sequence $h_0, h_1, \ldots$
- Choose the last hypothesis h_ℓ in the sequence that satisfies the triangle inequality $d(h_k, h_\ell) \leq d(h_k, \widehat{P_{Y|X}}) + d(h_\ell, \widehat{P_{Y|X}})$ with every preceding hypothesis h_k, $0 \leq k < \ell$. (Note that the inter-hypothesis distances $d(h_k, h_\ell)$ are measured on the unlabeled training data.)

Idea: Use U to Avoid Overfitting



Note:

- $\hat{d}(h_i^*, f)$ optimistically biased (too short)
- $\hat{d}(h_i^*, h_j^*)$ unbiased
- Distances must obey triangle inequality!

$$d(h_1, h_2) \leq d(h_1, f) + d(f, h_2)$$

→ Heuristic:

- Continue training until $\hat{d}(h_i, h_{i+1})$ fails to satisfy triangle inequality

Experimental Evaluation of TRI

[Schuurmans & Southey, MLJ 2002]

• Use it to select degree of polynomial for regression

• Compare to alternatives such as cross validation, structural risk minimization, …

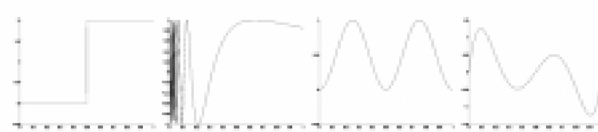


Figure 5: Target functions used in the polynomial curve fitting experiments (in order): step($x \geq 0.5$), $\sin(1/x)$, $\sin^2(2\pi x)$, and a fifth degree polynomial.

Generated y values contain zero mean Gaussian noise

Y=f(x)+ε

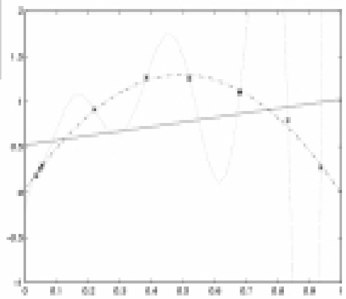


Figure 4: An example of minimum squared error polynomials of degrees 1, 2, and 9 for a set of 10 training points. The large degree polynomial demonstrates erratic behavior off the training set.

$t=20$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	2.04	1.03	1.00	1.00	1.06	1.00	1.01	1.58	1.02
50	3.11	1.37	1.33	1.34	1.94	1.35	1.61	18.2	1.32
75	3.87	2.23	2.30	2.13	10.0	2.75	4.14	1.2e3	1.83
95	5.11	9.45	8.84	8.26	5.0e3	11.8	82.9	1.8e5	3.94
100	8.92	105	526	105	2.0e7	2.1e3	2.7e5	2.4e7	6.30

$t=30$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.50	1.00	1.00	1.00	1.00	1.00	1.00	1.02	1.01
50	3.51	1.16	1.03	1.05	1.11	1.02	1.08	1.45	1.27
75	4.15	1.64	1.45	1.48	2.02	1.39	1.88	6.44	1.60
95	5.51	5.21	5.06	4.21	26.4	5.01	19.9	295	3.02
100	9.75	124	1.4e3	20.0	9.1e3	28.4	9.4e3	1.0e4	8.35

Table 4: Fitting $f(x)=\sin^2(2\pi x)$ with $P_x=U(0,1)$ and $\sigma=0.05$. Tables give distribution of approximation ratios achieved at training sample size $t=20$ and $t=30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.

Approximation ratio:

$$\frac{\text{true error of selected hypothesis}}{\text{true error of best hypothesis considered}}$$

Results using 200 unlabeled, t labeled

Cross validation (Ten-fold) → CVT

Structural risk minimization → SRM

Worst performance in top .50 of trials → row 50

$t=20$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.00	1.06	1.14	7.54	5.47	15.2	22.2	25.8	1.02
50	1.06	1.17	1.39	224	118	394	585	590	1.12
75	1.17	1.42	3.62	5.8e3	3.9e3	9.8e3	1.2e4	1.2e4	1.24
95	1.44	6.75	56.1	6.1e5	3.7e5	7.8e5	9.2e5	8.2e5	1.54
100	2.41	1.1e4	2.2e4	1.5e8	6.5e7	1.5e8	1.5e8	8.2e7	3.02

$t=30$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.00	1.08	1.17	4.69	1.51	5.41	5.45	2.72	1.06
50	1.08	1.17	1.54	34.8	9.19	39.6	40.8	19.1	1.14
75	1.19	1.37	9.68	258	91.3	266	266	150	1.25
95	1.45	6.11	419	4.7e3	2.7e3	4.8e3	5.1e3	4.0e3	1.51
100	2.18	643	1.6e7	1.6e7	1.6e7	1.6e7	1.6e7	1.6e7	2.10

Table 1: Fitting $f(x)=\text{step}(x \geq 0.5)$ with $P_x=U(0,1)$ and $\sigma=0.05$. Tables give distribution of approximation ratios achieved at training sample size $t=20$ and $t=30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.

Bound on Error of TRI Relative to Best Hypothesis Considered

Proposition 1 *Let h_m be the optimal hypothesis in the sequence $h_0, h_1, \ldots$ (that is, $h_m = \arg\min_{h_k} d(h_k, P_{Y|X})$) and let h_ℓ be the hypothesis selected by TRI. If (i) $m \leq \ell$ and (ii) $d(h_m, \widehat{P_{Y|X}}) \leq d(h_m, P_{Y|X})$ then*

$$d(h_\ell, P_{Y|X}) \leq 3d(h_m, P_{Y|X}) \quad (6)$$

Extension to TRI: Adjust for expected bias of training data estimates

[Schuurmans & Southey, MLJ 2002]

Procedure ADJ

- Given hypothesis sequence $h_0, h_1, \ldots$
- For each hypothesis h_ℓ in the sequence
  - multiply its estimated distance to the target $\widehat{d(h_\ell, P_{Y|X})}$ by the worst ratio of unlabeled and labeled distance to some predecessor h_k to obtain an adjusted distance estimate $\widehat{\widehat{d(h_\ell, P_{Y|X})}} = \widehat{d(h_\ell, P_{Y|X})} \frac{d(h_k, h_\ell)}{\widehat{\widehat{d(h_k, h_\ell)}}}$.
- Choose the hypothesis h_n with the smallest adjusted distance $\widehat{\widehat{d(h_n, P_{Y|X})}}$.

Experimental results: averaged over multiple target functions, outperforms TRI

Further Reading

- EM approach: K.Nigam, et al., 2000. "Text Classification from Labeled and Unlabeled Documents using EM", *Machine Learning,* 39, pp.103—134.
- CoTraining: A. Blum and T. Mitchell, 1998. “Combining Labeled and Unlabeled Data with Co-Training,” *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98).*
- S. Dasgupta, et al., “PAC Generalization Bounds for Co-training”, *NIPS 2001*
- Model selection: D. Schuurmans and F. Southey, 2002. “Metric-Based methods for Adaptive Model Selection and Regularizaiton,” Machine Learning, 48, 51—84.

Summary

Several ways to use unlabeled data in supervised learning

1. Use to reweight labeled examples
2. Use to help EM learn class-specific generative models
3. If problem has redundantly sufficient features, use CoTraining
4. Use to detect/preempt overfitting

Ongoing research area