Learning from Labeled and Unlabeled Data

Optional reading:
- see reading list on final slide

Machine Learning 10-601

April 1, 2009

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

When can Unlabeled Data improve supervised learning?

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

- Medical outcomes (x=<symptoms,treatment>, y=outcome)
- Text classification (x=document, y=relevance)
- Customer modeling (x=user actions, y=user intent)
- Sensor interpretation (x=<video,audio>, y=who's there)

When can Unlabeled Data help supervised learning?

Problem setting (the PAC learning setting):

- Set *X* of instances drawn from unknown distribution *P(X)*
- Wish to learn target function *f: X→ Y* (or, P(Y|X))
- Given a set *H* of possible hypotheses for *f*

Given:

- i.i.d. labeled examples $L = \{\langle x_1, y_1\rangle \ldots \langle x_m, y_m\rangle\}$
- i.i.d. unlabeled examples $U = \{x_{m+1}, \ldots x_{m+n}\}$

Wish to find hypothesis with lowest true error:

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

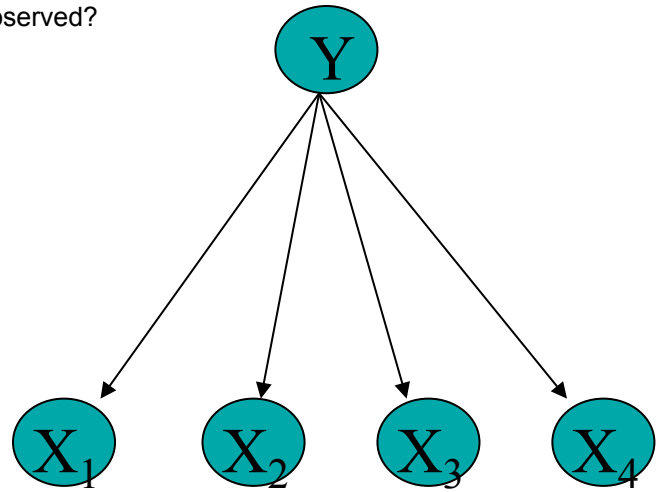
Idea 1: Use Labeled and Unlabeled Data to Train Bayes Net for P(X,Y)

Idea 1: Use Labeled and Unlabeled Data to Train Bayes Net for P(X,Y), then infer P(Y|X)

What CPDs are needed?

How do we estimate them from fully observed data?

How do we estimate them from partly observed?



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

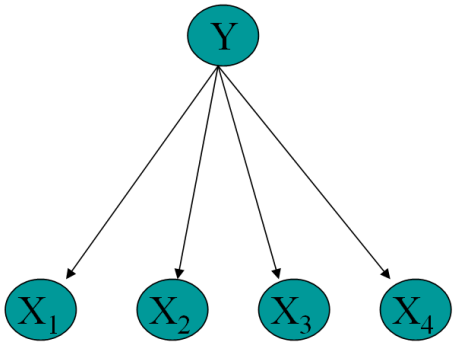
Supervised: Naïve Bayes Learner

Train:

For each class y_j of documents

1. Estimate $P(Y=y_j)$

2. For each word w_i estimate $P(X=w_i \mid Y=y_j)$



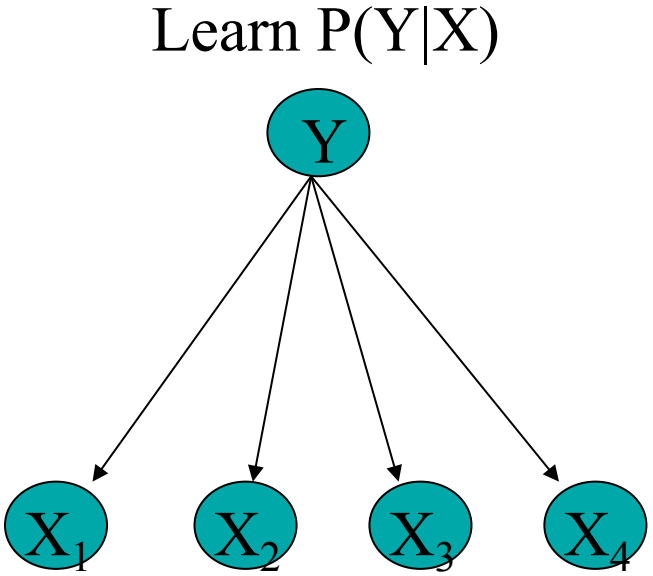
Classify (doc):

Assign *doc* to most probable* class

$$\hat{P}(y_j|doc) \leftarrow \frac{\hat{P}(y_j)\prod_i \hat{P}(w_i|y_j)}{\sum_k \hat{P}(y_k)\prod_i \hat{P}(w_i|y_k)}$$

* assuming words w_i are conditionally independent, given class

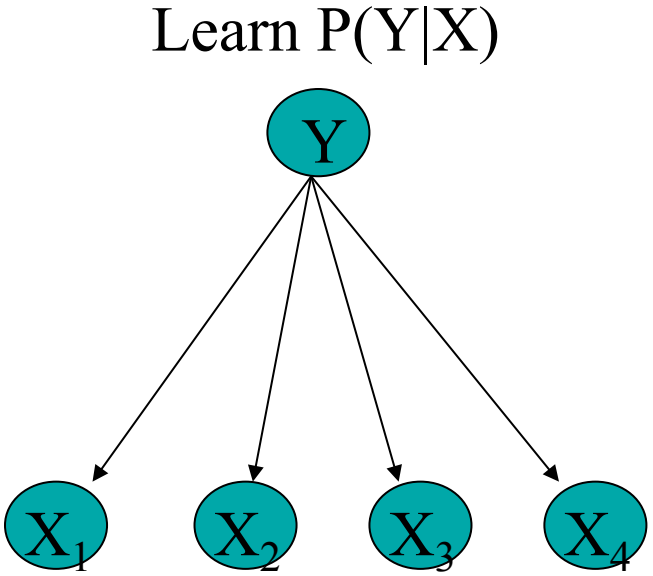
What if we have labels for only *some* documents?



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

What if we have labels for only *some* documents?

[Nigam et al., 2000]



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

EM: Repeat until convergence

1. Use probabilistic labels to train classifier h
2. Apply h to assign probabilistic labels to unlabeled data

[Nigam et al., 2000]

E Step:

$$\mathrm{P}(y_i = c_j|d_i;\hat{\theta}) = \frac{\mathrm{P}(c_j|\hat{\theta})\mathrm{P}(d_i|c_j;\hat{\theta})}{\mathrm{P}(d_i|\hat{\theta})}$$

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

M Step:

w_t is t-th word in vocabulary

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

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

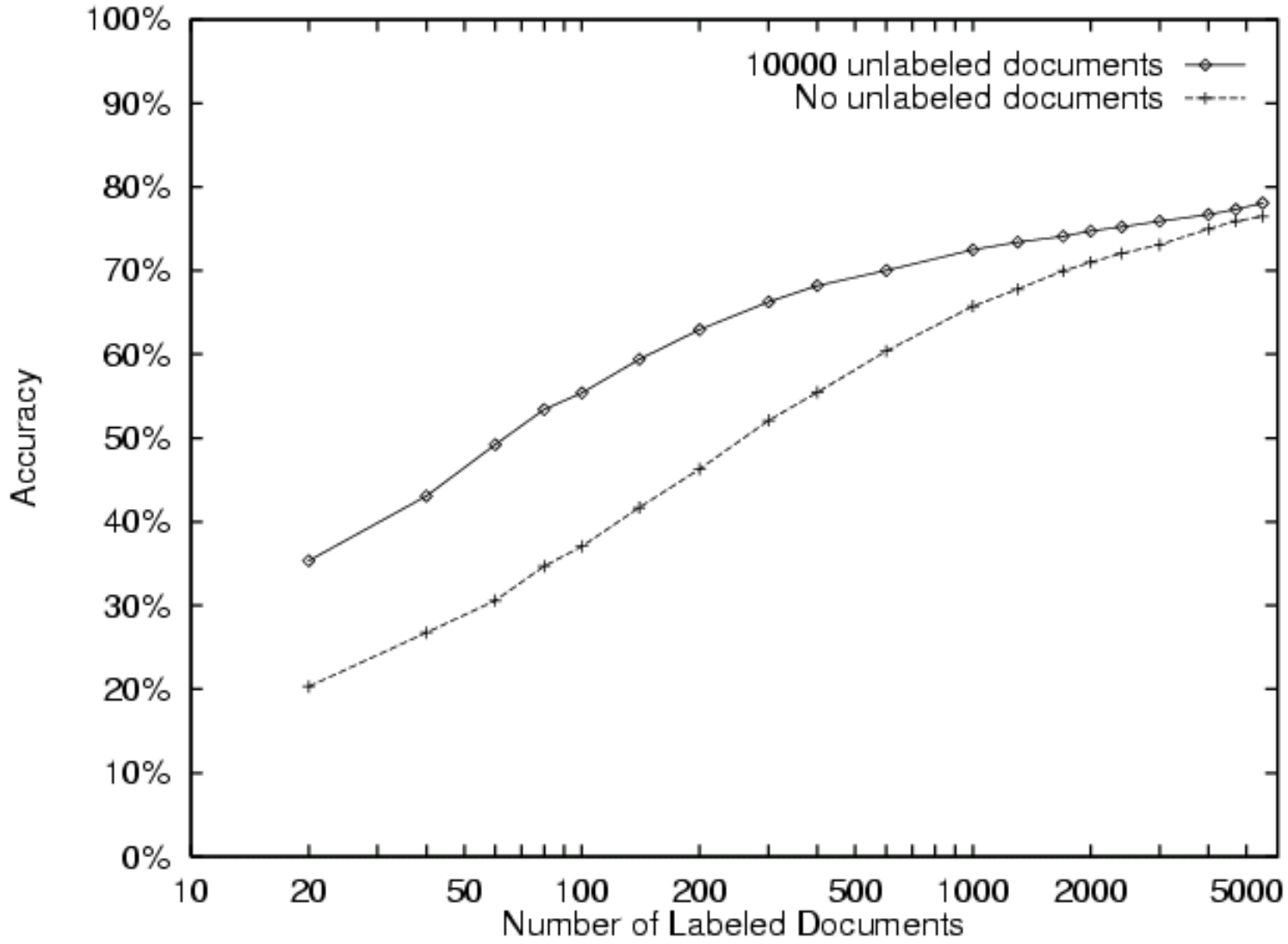
Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol *D* indicates an arbitrary digit.

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

Using one labeled example per class

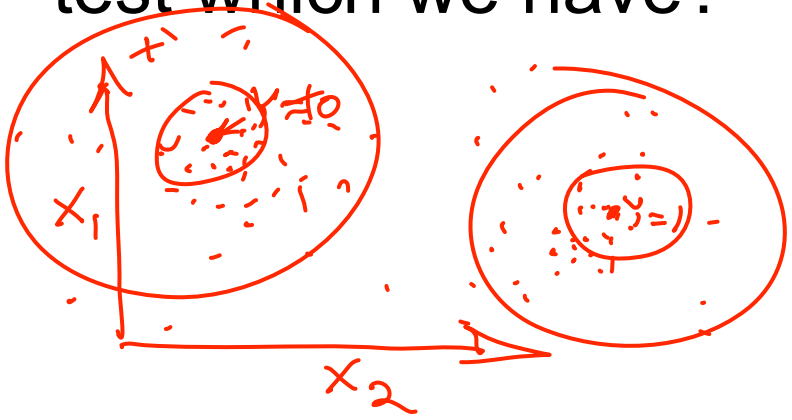
Words sorted by P(w|course) / P(w| : course)

20 Newsgroups

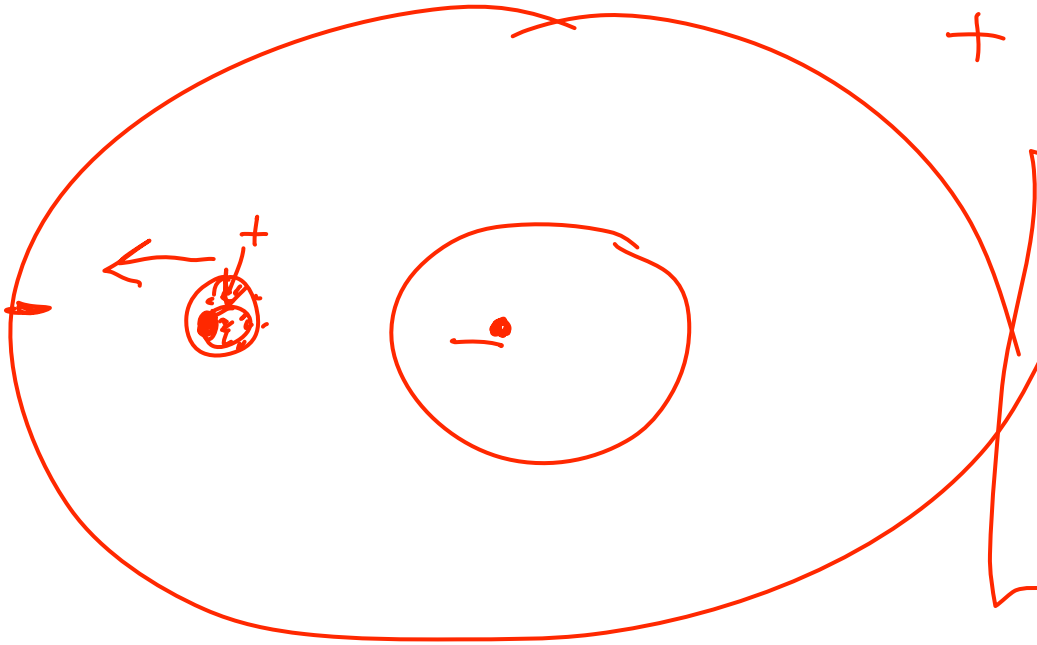


Why/When will this work?

- What's best case? Worst case? How can we test which we have?



Same data pts but target function has very scattered + vs − labels



assumptions that relate P(x) to P(y|x)

Summary : Semisupervised Learning with EM and Naïve Bayes Model

- If all data is labeled, corresponds to supervised training of Naïve Bayes classifier
- If all data unlabeled, corresponds to unsupervied, mixture-of-multinomial clustering
- If both labeled and unlabeled data, then unlabeled data helps if the Bayes net modeling assumptions are correct (e.g., P(X) is a mixture of class-conditional multinomials with conditionally independent X_i)
- Of course we could use Bayes net models other than Naïve Bayes

Idea 2: Use U to reweight labeled examples

- Most learning algorithms *minimize errors over labeled examples*
- But we really want to *minimize error over future examples* drawn from the same underlying distribution (ie., *true error* of hypothesis)
- If we know the underlying distribution P(X), we could weight each labeled training example <x,y> by its probability according to P(X=x)
- Unlabeled data allows us to estimate P(X)

Idea 2: Use U to reweight labeled examples L

Use $U \rightarrow \hat{P}(X)$ to alter the loss function

if its argument is true, then 1, else 0

- Wish to minimize true error:

$$\hat{f} \leftarrow \arg\min_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) P(x)$$

- Usually approximate this by training error:

$$\hat{f} \leftarrow \arg\min_{h \in H} \frac{1}{L} \sum_{\langle x,y \rangle \in L} \delta(h(x) \neq y)$$

Which equals:

$$\hat{f} \leftarrow \arg\min_{h \in H} \sum_{x \in X} \delta(h(x) \neq y) \left[\frac{n(x, L)}{|L|} \right]$$

n(x,L) = number of times x occurs in L

- We can produce a better approximation by incorporating U:

$$\hat{f} \leftarrow \arg\min_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) \left[\frac{n(x, L) + n(x, U)}{|L| + |U|} \delta(n(x, L) > 0) \right]$$

Reweighting Labeled Examples

- Wish to find

$$\hat{f} \leftarrow \arg\min_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) \left[\delta(n(x, L) > 0) \frac{n(x, L) + n(x, U)}{|L| + |U|} \right]$$

- Already have algorithm (e.g., decision tree learner) to find

$$\hat{f} \leftarrow \arg\min_{h \in H} \frac{1}{L} \sum_{\langle x, y \rangle \in L} \delta(h(x) \neq y)$$

- Just reweight examples in L, and have algorithm minimize

$$\hat{f} \leftarrow \arg\min_{h \in H} \frac{1}{L} \sum_{\langle x, y \rangle \in L} \delta(h(x) \neq y) \frac{n(x, L) + n(x, U)}{|L| + |U|}$$

- Or if X is continuous, use L+U to estimate p(X), and minimize

$$\hat{f} \leftarrow \arg\min_{h \in H} \frac{1}{L} \sum_{\langle x, y \rangle \in L} \delta(h(x) \neq y) \ \hat{p}(x)$$

Reweighting Labeled Examples: Summary

- Simple, very general idea
- But I haven't seen this discussed or attempted anywhere in the literature…
- Why not?

3. Use U to Detect/Preempt Overfitting

- Overfitting is a problem for many learning algorithms (e.g., decision trees, neural networks)

- The symptom of overfitting: complex hypothesis h2 performs better on training data than simpler hypothesis h1, but worse on test data

- Unlabeled data can help detect overfitting, by comparing predictions of h1 and h2 over the unlabeled examples
  - Key insight: The rate at which h1 and h2 disagree on U should be bounded by the rates at which they each disagree with L, unless overfitting is occurring

4. Use U to Detect/Preempt Overfitting

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

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

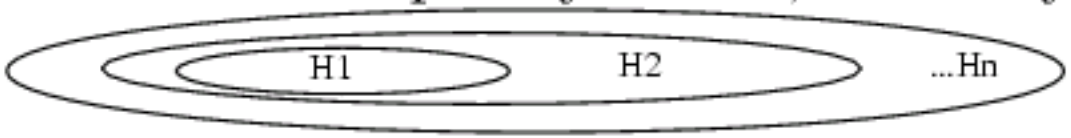
true error

estimates →

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

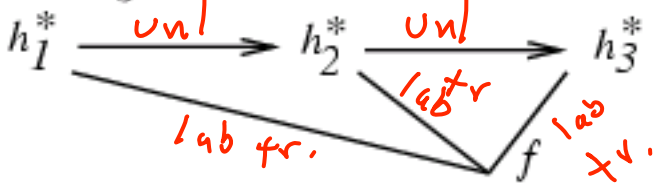
$$\hat{d}(h_1, h_2) = \frac{1}{|U|} \sum_{x \in U} \delta(h_1(x) \neq h_2(x))$$

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)* $\leq$ *d(f,h)+d(h,g)*

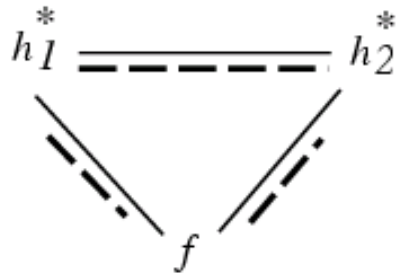
- Classification with zero-one loss:

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

- Regression with squared loss:

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

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!

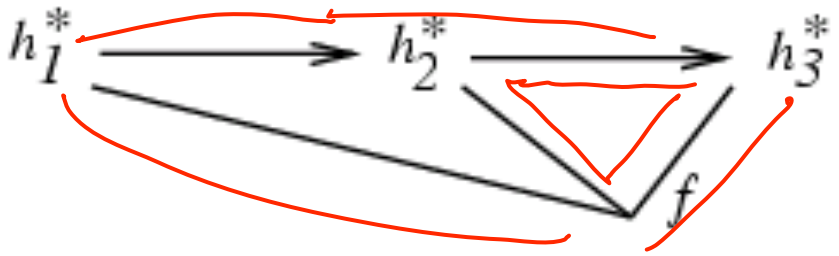
$$d(h_1, h_2) \leq d(h_1, f) + d(f, h_2)$$

$\rightarrow$ Heuristic:

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

Procedure TRI

- Given hypothesis sequence $h_0, h_1, ...$
- Choose the last hypothesis h_ℓ in the sequence that satisfies the triangle inequality $d(h_k, h_\ell) \leq d(h_k, \widehat{\mathrm{P}_{Y|X}}) + d(h_\ell, \widehat{\mathrm{P}_{Y|X}})$ 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 *un*labeled training data.)



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, …

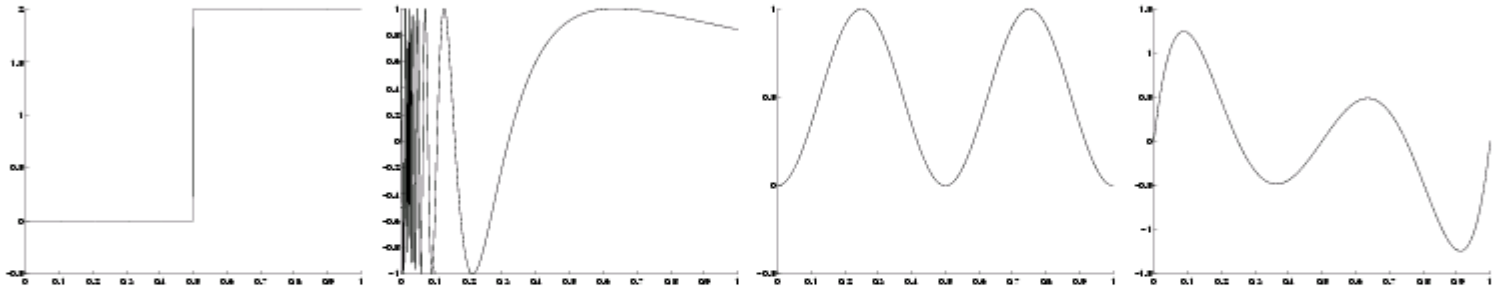


Figure 5: Target functions used in the polynomial curve fitting experiments (in order): $\text{step}(x \geq 0.5)$, $\sin(1/x)$, $\sin^2(2\pi x)$, and a fifth degree polynomial.

Generated y values contain zero mean Gaussian noise ε

Y=f(x)+ε

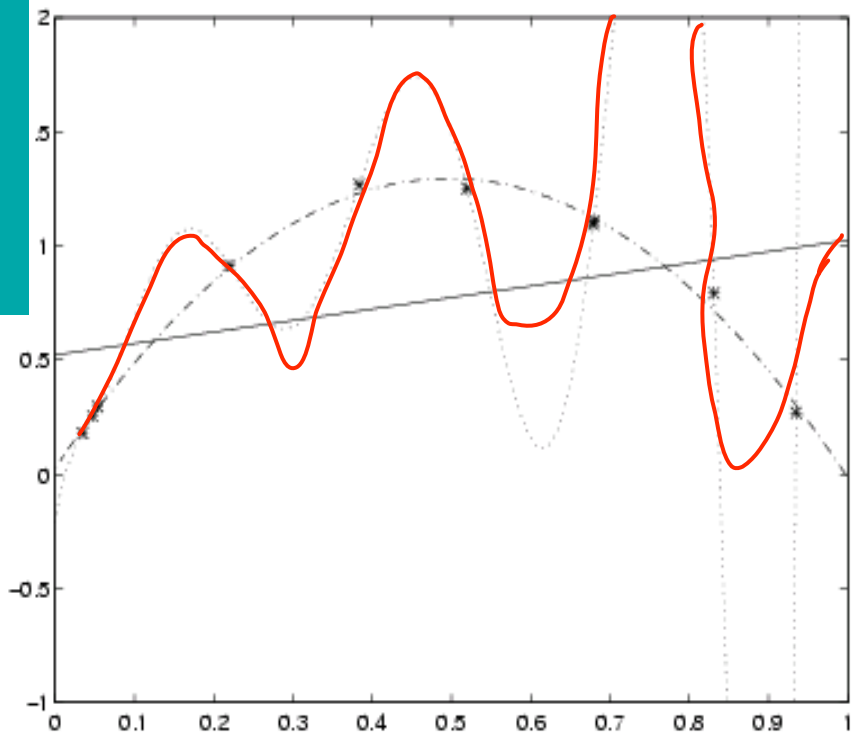


Figure 4: An example of minimum squared error polynomials of degrees 1, 2, and 9 for a set of 10 training points. The large degree polynomial demonstrates erratic behavior off the training set.

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)

Structural risk minimization

Worst performance in top .50 of trials

$t = 20$	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	159	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.

$t = 20$	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 $\mathrm{P}_X = U(0, 1)$ 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, $h_m = \arg\min_{h_k} d(h_k, \mathrm{P}_{Y|X})$) and let h_ℓ be the hypothesis selected by TRI. If (i) $m \leq \ell$ and (ii) $d(h_m, \widehat{\mathrm{P}_{Y|X}}) \leq d(h_m, \mathrm{P}_{Y|X})$ then*

$$d(h_\ell, \mathrm{P}_{Y|X}) \quad \leq \quad 3d(h_m, \mathrm{P}_{Y|X}) \tag{6}$$

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

[Schuurmans & Southey, MLJ 2002]

Procedure ADJ

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

Experimental results: averaged over multiple target functions, outperforms TRI

What you should know

1. Unlabeled can help EM learn Bayes nets for P(X,Y)
   - If we assume the Bayes net structure is correct

2. Using unlabeled data to reweight labeled examples gives better approximation to true error
   - If we assume examples drawn from stationary P(X)

3. Use unlabeled data to detect/preempt overfitting
   - If we assume priors over H that correctly order hypotheses

Further Reading

- Semi-Supervised Learning, O. Chapelle, B. Sholkopf, and A. Zien (eds.), MIT Press, 2006. (excellent book)

- EM for Naïve Bayes classifiers: K.Nigam, et al., 2000. "Text Classification from Labeled and Unlabeled Documents using EM", *Machine Learning,* 39, pp.103—134.

- Model selection: D. Schuurmans and F. Southey, 2002. “Metric-Based methods for Adaptive Model Selection and Regularization,” Machine Learning, 48, 51—84.