Today:  - Semi supervised learning
Next time: - Markov models
Topics

Experiment Selection

Modeling

Supervised Learning (Classification)

Unsupervised Learning (Clustering)

Data Analysis (Estimation and Regression)

Today
When can Unlabeled Data improve 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)

• Customer modeling (x=user actions, y=user intent)

• …
When can Unlabeled Data help supervised learning?

Consider setting:
- Set $X$ of instances drawn from unknown distribution $P(X)$
- Wish to learn target function $f: X \rightarrow Y$ (or, $P(Y|X)$)
- Given a 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 unlabeled examples $U = \{x_{m+1}, \ldots x_{m+n}\}$

Determine:
$$\hat{f} \leftarrow \arg \min_{h \in H} \frac{\Pr_{x \in P(X)} [h(x) \neq f(x)]}{\Pr_{x \in P(U)} [h(x) \neq f(x)]}$$
Four Ways to Use Unlabeled Data for Supervised Learning

1. Use to re-weight 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

- Most machine learning algorithms (neural nets, decision trees, Support Vector Machines) attempt to minimize errors over labeled examples.
- But we really want to minimize error over future examples drawn from the same underlying distribution.
- If we know the underlying distribution, we should weight each training example by its probability according to this distribution.
- Unlabeled data allows us to estimate this distribution more accurately, and to reweight our labeled examples accordingly.
1. Use $U$ to reweight labeled examples

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

- Wish to find

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

- Often approximate as

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

- Can use $U$ for improved approximation:

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

1 if hypothesis $h$ disagrees with true function $f$, else 0
2. If assume form of $P(X|Y)$, use Clustering-Style EM

- First, consider completely unsupervised clustering, where we assume data $X$ is generated by a mixture of probability distributions, one for each cluster
  - There are widely used algorithms for finding clusters (e.g., Gaussian mixtures) using EM
- Some classifier learning algorithms such as Gaussian classifiers also assume the data $X$ is generated by a mixture of distributions, one for each class $Y$
- Supervised learning: estimate $P(X|Y)$ from labeled data
- Opportunity: estimate $P(X|Y)$ from labeled and unlabeled data, using EM as in clustering
Bag of Words Text Classification

Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.

aardvark 0
about 2
all 2
Africa 1
apple 0
anxious 0
...
gas 1
...
oil 1
...
Zaire 0
Baseline: Naïve Bayes Learner

**Train:**

For each class $c_j$ of documents

1. Estimate $P(c_j)$

2. For each word $w_i$ estimate $P(w_i | c_j)$

**Classify (doc):**

Assign doc to most probable* class

$$\arg \max_{j} P(c_j) \prod_{w_i \in \text{doc}} P(w_i | c_j)$$

* assuming words are conditionally independent, given class
<table>
<thead>
<tr>
<th>Faculty</th>
<th>Students</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>associate</td>
<td>resume</td>
<td>homework</td>
</tr>
<tr>
<td>chair</td>
<td>advisor</td>
<td>syllabus</td>
</tr>
<tr>
<td>member</td>
<td>student</td>
<td>assignments</td>
</tr>
<tr>
<td>ph</td>
<td>working</td>
<td>exam</td>
</tr>
<tr>
<td>director</td>
<td>stuff</td>
<td>grading</td>
</tr>
<tr>
<td>fax</td>
<td>links</td>
<td>midterm</td>
</tr>
<tr>
<td>journal</td>
<td>homepage</td>
<td>pm</td>
</tr>
<tr>
<td>recent</td>
<td>interests</td>
<td>instructor</td>
</tr>
<tr>
<td>received</td>
<td>personal</td>
<td>due</td>
</tr>
<tr>
<td>award</td>
<td>favorite</td>
<td>final</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Departments</th>
<th>Research Projects</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>departmental</td>
<td>investigators</td>
<td>type</td>
</tr>
<tr>
<td>colloquia</td>
<td>group</td>
<td>jan</td>
</tr>
<tr>
<td>apartment</td>
<td>members</td>
<td>enter</td>
</tr>
<tr>
<td>seminars</td>
<td>researchers</td>
<td>random</td>
</tr>
<tr>
<td>schedules</td>
<td>laboratory</td>
<td>program</td>
</tr>
<tr>
<td>webmaster</td>
<td>develop</td>
<td>net</td>
</tr>
<tr>
<td>events</td>
<td>related</td>
<td>time</td>
</tr>
<tr>
<td>facilities</td>
<td>accessed</td>
<td>format</td>
</tr>
<tr>
<td>people</td>
<td>affiliated</td>
<td>access</td>
</tr>
<tr>
<td>postgraduate</td>
<td>project</td>
<td>begin</td>
</tr>
</tbody>
</table>
Expectation Maximization (EM) Algorithm

• Use labeled data $L$ to learn initial classifier $h$

Loop:
• E Step:
  – Assign probabilistic labels to $U$, based on $h$
• M Step:
  – Retrain classifier $h$ using both $L$ and assigned labels to $U$

Under certain conditions, guaranteed to converge to locally maximum likelihood $h$
2. Use U with EM and Assumed Generative Model

Learn $P(Y|X)$

<table>
<thead>
<tr>
<th>Y</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
E Step:

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

\(w_t\) is \(t\)-th word in vocabulary

M Step:

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

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

<table>
<thead>
<tr>
<th>Iteration 0</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>intelligence</td>
<td>$DD$</td>
<td>$D$</td>
</tr>
<tr>
<td>artificial</td>
<td>$D$</td>
<td></td>
</tr>
<tr>
<td>understanding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$DDw$</td>
<td>$D$</td>
<td></td>
</tr>
<tr>
<td>dist</td>
<td>lecture</td>
<td></td>
</tr>
<tr>
<td>identical</td>
<td>cc</td>
<td></td>
</tr>
<tr>
<td>rus</td>
<td>$D^*$</td>
<td></td>
</tr>
<tr>
<td>arrange</td>
<td>handout</td>
<td></td>
</tr>
<tr>
<td>games</td>
<td>due</td>
<td></td>
</tr>
<tr>
<td>dartmouth</td>
<td>problem</td>
<td></td>
</tr>
<tr>
<td>natural</td>
<td>set</td>
<td></td>
</tr>
<tr>
<td>cognitive</td>
<td>tay</td>
<td></td>
</tr>
<tr>
<td>logic</td>
<td>$DDam$</td>
<td></td>
</tr>
<tr>
<td>proving</td>
<td>yurttas</td>
<td></td>
</tr>
<tr>
<td>prolog</td>
<td>homework</td>
<td></td>
</tr>
<tr>
<td>knowledge</td>
<td>homework</td>
<td></td>
</tr>
<tr>
<td>human</td>
<td>set</td>
<td></td>
</tr>
<tr>
<td>representation</td>
<td>hw</td>
<td></td>
</tr>
<tr>
<td>field</td>
<td>exam</td>
<td></td>
</tr>
<tr>
<td></td>
<td>solution</td>
<td></td>
</tr>
<tr>
<td></td>
<td>assaf</td>
<td></td>
</tr>
</tbody>
</table>

Using one labeled example per class
Experimental Evaluation

- Newsgroup postings
  - 20 newsgroups, 1000/group
Experimental Evaluation

Newsgroup postings
– 20 newsgroups,
  1000/group
3. If Problem Setting Provides Redundantly Sufficient Features, use CoTraining

- In some settings, available data features are so redundant that we can train two classifiers using different features

- In this case, the two classifiers should at least agree on the classification for each unlabeled example

- Therefore, we can use the unlabeled data to constrain training of both classifiers, forcing them to agree
3. If Problem Setting Provides Redundantly Sufficient Features, use CoTraining

\[
\text{learn } f : X \rightarrow Y \\
\text{where } X = X_1 \times X_2 \\
\text{where } x \text{ drawn from unknown distribution} \\
\text{and } \exists g_1, g_2 \quad (\forall x) g_1(x_1) = g_2(x_2) = f(x)
\]
Redundantly Sufficient Features

Professor Faloutsos

U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto); B.Sc. (National Technical University of Athens)

Research Interests:

- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;
Given: labeled data $L$, unlabeled data $U$

Loop:

Train $g_1$ (hyperlink classifier) using $L$
Train $g_2$ (page classifier) using $L$
Allow $g_1$ to label $p$ positive, $n$ negative examps from $U$
Allow $g_2$ to label $p$ positive, $n$ negative examps from $U$
Add the intersection of the self-labeled examples to $L$
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%
Co-Training Rote Learner

My advisor

hyperlinks

+ 

- 

pages
Co-Training Rote Learner
### Classifying Jobs for FlipDog

**X1: job title**

<table>
<thead>
<tr>
<th>Job Title</th>
<th>Company</th>
<th>Location</th>
<th>Date Posted</th>
<th>Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++/Java Consultants</td>
<td>Elite Placement Services</td>
<td>November 01, 2000</td>
<td>Houston, TX</td>
<td>Computing/MIS Software Development</td>
</tr>
<tr>
<td>Chief Software Architect</td>
<td>Elite Placement Services</td>
<td>November 01, 2000</td>
<td>Houston, TX</td>
<td>Computing/MIS Software Development</td>
</tr>
<tr>
<td>Web Application Developers</td>
<td>MI Systems, Inc.</td>
<td>November 01, 2000</td>
<td>Houston, TX</td>
<td>Computing/MIS Internet Development</td>
</tr>
<tr>
<td>Sales Consulting Engineer</td>
<td>Visual Numerics, Inc.</td>
<td>November 01, 2000</td>
<td>Houston, TX</td>
<td>Computing/MIS Technical Support/Help Desk</td>
</tr>
<tr>
<td>Peoplesoft Software Analyst</td>
<td>I.T. Staffing, Inc.</td>
<td>October 27, 2000</td>
<td>Houston, TX</td>
<td>Computing/MIS Software Development</td>
</tr>
<tr>
<td>Peoplesoft Software Analyst</td>
<td>I.T. Staffing, Inc.</td>
<td>October 27, 2000</td>
<td>Houston, TX</td>
<td>Computing/MIS Software Development</td>
</tr>
</tbody>
</table>

**X2: job description**

- Functions of this position include the consulting, development and implementation of EAI solutions supporting e-commerce and B2B initiatives for...
- Responsible for the end-to-end architecture of all r-tiered web-based applications and complementary products. Provide design direction for the...
- Permanent Opportunities (2) Application Developers with...
- Performs pre-sales technical demonstrations, product demonstrations and product training to customers and non-customers. Technical support includes providing verbal and written response...
- CLIENT/SERVER APPLICATION ADMINISTRATION. SETTING UP USERS AND SECURITY FOR DATABASE AND APPLICATION.
Gradient CoTraining
Classifying FlipDog job descriptions: SysAdmin vs. WebProgrammer

Final Accuracy
Labeled data alone: 86%
CoTraining: 96%
4. 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 $h_2$ performs better on training data than simpler hypothesis $h_1$, but worse on test data

- Unlabeled data can help detect overfitting, by comparing predictions of $h_1$ and $h_2$ over the unlabeled examples
  - The rate at which $h_1$ and $h_2$ disagree on $U$ should be the same as the rate on $L$, unless overfitting is occurring
Defining a distance metric

• Definition of distance metric
  – Non-negative \( d(f,g) \geq 0 \);
  – 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:

\[
d(h_1, h_2) \equiv \sqrt{\int (h_1(x) - h_2(x))^2p(x)dx}
\]
Using the distance metric

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

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

Prefer $h_1^*$, $h_2^*$, or $h_3^*$?
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
Procedure TRI

- Given hypothesis sequence $h_0, h_1, \ldots$
- Choose the last hypothesis $h_\ell$ in the sequence that satisfies the triangle inequality $d(h_k, h_\ell) \leq d(h_k, P_{Y|x}) + d(h_\ell, P_{Y|x})$ 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.)
Generated y values contain zero mean Gaussian noise $\epsilon$.

$Y = f(x) + \epsilon$

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.
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.
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

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

<table>
<thead>
<tr>
<th>$t = 30$</th>
<th>TRI</th>
<th>CVT</th>
<th>SRM</th>
<th>RIC</th>
<th>GCV</th>
<th>BIC</th>
<th>AIC</th>
<th>FPE</th>
<th>ADJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>1.00</td>
<td>1.08</td>
<td>1.17</td>
<td>4.69</td>
<td>1.51</td>
<td>5.41</td>
<td>5.45</td>
<td>2.72</td>
<td>1.06</td>
</tr>
<tr>
<td>50</td>
<td>1.08</td>
<td>1.17</td>
<td>1.54</td>
<td>34.8</td>
<td>9.19</td>
<td>39.6</td>
<td>40.8</td>
<td>19.1</td>
<td>1.14</td>
</tr>
<tr>
<td>75</td>
<td>1.19</td>
<td>1.37</td>
<td>9.68</td>
<td>258</td>
<td>91.3</td>
<td>266</td>
<td>266</td>
<td>159</td>
<td>1.25</td>
</tr>
<tr>
<td>95</td>
<td>1.45</td>
<td>6.11</td>
<td>419</td>
<td>4.7e3</td>
<td>2.7e3</td>
<td>4.8e3</td>
<td>5.1e3</td>
<td>4.0e3</td>
<td>1.51</td>
</tr>
<tr>
<td>100</td>
<td>2.18</td>
<td>643</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>2.10</td>
</tr>
</tbody>
</table>

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.
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
Further Reading

- S. Dasgupta, et al., “PAC Generalization Bounds for Co-training”, *NIPS 2001*
Acknowledgment

These slides are based in on slides from Tom Mitchell.