The problem of subcellular location analysis

Cell Type (Order 10^2)
Condition (Order 10^2)
Protein (Order 10^4)

Plus: Time scale from subsecond to years

Automated Science
(Active Learning)

Experimental Data
Automated Interpretation
Other Data
What new data is needed the most?
Modeling
Efficient Acquisition and Learning of Fluorescence Microscope Data Models

1. Use all the input from the microscope to model the data set
2. Choose acquisition requests that allow us to construct an accurate model in the shortest amount of time

Charles Jackson & Jelena Kovacevic

Active Learning

Slides from Irina Rish and Barbara Engelhardt

Problem Setup

- Unlabeled data available but labels are expensive
- I would like to choose which data to label
  - to maximize the “value” of that data to my problem
  - to minimize the “cost” of labeling
Toy Example: threshold function

Unlabeled data: labels are all 0 then all 1 (left to right)
Classifier is threshold function:
\[ h(x) = 1 \text{ if } x > w \text{ (0 otherwise)} \]
Goal: find transition between 0 and 1 labels in minimum steps
Naive method: choose points to label at random on line
Better method: binary search for transition between 0 and 1

Example: Sequencing genomes
- What genome should be sequenced next?
- Criteria for selection?
- Optimal species to detect functional elements across genomes
- Breadth of species encompassing biological phenomena of interest
- (Not the same as the most diverged set of species)
- Marsupials should be sequenced next

Example: collaborative filtering
- Users rate only a few movies usually; ratings “expensive”
- Which movies do you show users to best extrapolate movie preferences?
- Also known as questionnaire design
  - Baseline questionnaires:
    - Random: m movies randomly
    - Most Popular Movies: m most frequently rated movies
  - Most popular movies is not better than random design!
  - Popular movies rated highly by all users; do not discriminate tastes

[McAuliffe et al., 2004]
[Yu et al. 2006]
Entropy Function

- A measure of information in a random event $X$ with possible outcomes $\{x_1, \ldots, x_n\}$
  
  \[ H(X) = -\sum p(x_i) \log p(x_i) \]

- Comments on entropy function:
  - Entropy of an event is zero when the outcome is known
  - Entropy is maximal when all outcomes are equally likely

- The average minimum yes/no questions to answer some question (connection to binary search)

Loss Functions

- A function $L$ that maps an event to a real number, representing cost or regret associated with event

- E.g., in regression problems, $L(y, \theta f(x))$ maps to reals

- Examples:
  - Quadratic (least squares) loss
  - Linear (absolute value) loss
  - 0-1 (binary) loss
  - Exponential

Risk Function

- Risk is also known as expected loss

- The (frequentist) risk function is explicitly expected loss

\[ R(\theta, X) = \sum L(\theta, x) p(x | \theta) \]

- Bayes risk is defined as posterior expected loss:

\[ R(\Theta, X) = \sum \theta L(\theta, x) p(\theta | x) \]

- Trade-off: Bayes risk performs well when $p(\theta | x)$ is accurate

- "Gain" here is choosing $x$ to minimize expected loss
What is Active Learning?

- Unlabeled data are readily available; labels are expensive
- Want to use adaptive decisions to choose which labels to acquire for a given dataset
- Goal is accurate classifier with minimal cost

Active learning warning

- Choice of data is only as good as the model itself
- Assume a linear model, then two data points are sufficient
- What happens when data are not linear?

Active Learning

- Active learner is able to query world and receive a response before outputting a classifier
- Learner selects queries (but cannot impact response)
- Two general methods:
  - Select “most uncertain” data given model and parameters
  - Select “most informative” data to optimize expected gain
- Given model $M$ with parameters $\theta$ and loss function $L$
- Query $q$ with response $x$ updates the model posterior $\theta'$
  $$L(\theta', X) = E_L(\theta')$$
Active Learning Approaches

- Membership queries
- Uncertainty Sampling
- Query by committee

Membership queries

Earliest model of active learning in theory work [Angluin 1992]

- $X =$ space of possible inputs, like $[0,1]^n$
- $H =$ class of hypotheses

Target concept $h^* \in H$ to be identified exactly.
You can ask for the label of any point in $X$: no unlabeled data.

- $H_0 = H$
- For $t = 1, 2, \ldots$
  - pick a point $x \in X$ and query its label $h^*(x)$
  - let $H_t =$ all hypotheses in $H_{t-1}$ consistent with $(x, h^*(x))$

What is the minimum number of “membership queries” needed to reduce $H$ to just $[h^*]$?

Membership queries: example

- $X = [0,1]^n$
- $H =$ AND-of-positive-literals, like $x_1 \land x_2 \land x_{10}$
- $S = \{}$ (set of AND positions)
  - For $i = 1$ to $n$:
    - ask for the label of $(1, \ldots, 1, 0, 1, \ldots, 1)$ [0 at position $i$]
    - if negative: $S = S \cup \{i\}$

Total: $n$ queries

General idea: synthesize highly informative points.
Each query cuts the version space -- the set of consistent hypotheses -- in half.
Problem

Many results in this framework, even for complicated hypothesis classes.
[Baum and Lang, 1991] tried fitting a neural net to handwritten characters. Synthetic instances created were incomprehensible to humans!

[Lewis and Gale, 1992] tried training text classifiers. “an artificial text created by a learning algorithm is unlikely to be a legitimate natural language expression, and probably would be uninterpretable by a human teacher.”

Uncertainty Sampling

[Lewis & Gale, 1994]

• Query the event that the current classifier is most uncertain about

If uncertainty is measured in Euclidean distance:

• Used trivially in SVMs, graphical models, etc.

A Sequential Algorithm for Training Text Classifiers

David B. Lewis (dlewis@research.att.com) and William A. Gale (wag@research.att.com)

AT&T Bell Laboratories, Murray Hill, NJ 07974, USA


Abstract

The ability to cheaply train text classifiers is critical to their use in information retrieval, content analysis, natural language processing, and other tasks involving text which is partly or fully textually unstructured. This paper presents a novel approach for training text classifiers which combines on-line learning techniques and novel text classification techniques. The method, called sequential uncertainty sampling, is described and tested on a number of text classification tasks.
Score Function

\[
\text{score}_{\text{uncert}}(S_i) = \text{uncertainty}(P(S_i \mid O_i)) = H(S_i) = \sum P(S_i = i) \log P(S_i = i)
\]

Uncertainty Sampling Example

<table>
<thead>
<tr>
<th></th>
<th>Sex</th>
<th>Age</th>
<th>Test A</th>
<th>Test B</th>
<th>Test C</th>
<th>S_i</th>
<th>P(S_i)</th>
<th>H(S_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>20-30</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>0.02</td>
<td>0.043</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>20-30</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>0.01</td>
<td>0.024</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>30-40</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>?</td>
<td>0.05</td>
<td>0.086</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>60+</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>FALSE</td>
<td>0.12</td>
<td><strong>0.159</strong></td>
</tr>
<tr>
<td>5</td>
<td>M</td>
<td>10-20</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>0.01</td>
<td>0.024</td>
</tr>
<tr>
<td>6</td>
<td>M</td>
<td>20-30</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td><strong>0.96</strong></td>
<td>0.073</td>
</tr>
</tbody>
</table>

Uncertainty Sampling Example

<table>
<thead>
<tr>
<th></th>
<th>Sex</th>
<th>Age</th>
<th>Test A</th>
<th>Test B</th>
<th>Test C</th>
<th>S_i</th>
<th>P(S_i)</th>
<th>H(S_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>20-30</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>0.01</td>
<td>0.024</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>20-30</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>0.02</td>
<td>0.043</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>30-40</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>?</td>
<td>0.04</td>
<td>0.073</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>60+</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>FALSE</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>M</td>
<td>10-20</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>TRUE</td>
<td><strong>0.06</strong></td>
<td><strong>0.112</strong></td>
</tr>
<tr>
<td>6</td>
<td>M</td>
<td>20-30</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td><strong>0.97</strong></td>
<td>0.059</td>
</tr>
</tbody>
</table>
Uncertainty Sampling

GOOD: couldn’t be easier
GOOD: often performs pretty well

BAD: $H(S_t)$ measures information gain about the samples, not the model

Sensitive to noisy samples

---

If our objective is to reduce the prediction error, then

"the expected information gain of an unlabeled sample is NOT a sufficient criterion for constructing good queries"
Ooh, now we're going to learn something for sure!
One of them is definitely wrong.
The Original QBC Algorithm

As each example arrives...

1. Choose a committee, $C$, (usually of size 2) randomly from Version Space

2. Have each member of $C$ classify it

3. If the committee disagrees, select it.