Machine Learning & Datamining

CSE 454

Project Part 1 Feedback

• Serialization
  Java Supplied vs. Manual

"... to balance speed of indexing with speed of lookup"

Inverted Files for Multiple Documents

LEXICON

<table>
<thead>
<tr>
<th>WORD</th>
<th>IDOCID</th>
<th>OCCUR</th>
<th>POS 1</th>
<th>POS 2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
<td>20</td>
<td>1</td>
<td>10</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>word 1</td>
<td>15</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>word 2</td>
<td>4</td>
<td>1</td>
<td>30</td>
<td>100</td>
<td>500</td>
</tr>
</tbody>
</table>

OCCURRENCE INDEX

<table>
<thead>
<tr>
<th>IDOCID</th>
<th>OCCUR</th>
<th>POS 1</th>
<th>POS 2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1</td>
<td>10</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>30</td>
<td>100</td>
<td>500</td>
</tr>
</tbody>
</table>

• One method. Alta Vista uses alternative

Course Overview

Datamining

- Information Retrieval
- Precision vs Recall
- Inverted Indices

P2P

Security

Web Services

Semantic Web

Advt

New Stuff

Case Studies: Nutch, Google, AltaVista

Systems Foundation: Networking, Synchronization & Monitors
Tentative Schedule

11/1 Machine learning & datamining
   Text categorization & evaluation methods
11/8 Information extraction
11/15 continued
   Clustering & Focused crawling
11/22 Cryptography
   Security
11/29 Outbreak
   Guest Lecture
12/6 P2P & Advertising
12/8 Semantic Web

Why Machine Learning

- Flood of data
  WalMart - 25 Terabytes
  WWW - 1,000 Terabytes
- Speed of computer vs. %#@! of programming
  Highly complex systems (telephone switching systems)
  Productivity = 1 line code @ day @ programmer
- Desire for customization
  A browser that browses by itself?
- Hallmark of Intelligence
  How do children learn language?

Applications of ML

- Credit card fraud
- Product placement / consumer behavior
- Recommender systems
- Speech recognition

Most mature & successful area of AI

Examples of Learning

- Baby touches stove, gets burned, ... next time...
- Medical student is shown cases of people with disease X, learns which symptoms...
- How many groups of dots?

What is Machine Learning??

A program is said to learn from experience E with respect to task T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

- Task T:
  Playing checkers
- Performance Measure P:
  Percent of games won against opponents
- Experience E:
  Playing practice games against itself

Defining a Learning Problem
Issues

- What feedback (experience) is available?
- How should these features be represented?
- What kind of knowledge is being increased?
- How is that knowledge represented?
- What prior information is available?
- What is the right learning algorithm?
- How avoid overfitting?

Choosing the Training Experience

- Credit assignment problem:
  - Direct training examples:
    - E.g. individual checker boards + correct move for each
    - Supervised learning
  - Indirect training examples:
    - E.g. complete sequence of moves and final result
    - Reinforcement learning
- Which examples:
  - Random, teacher chooses, learner chooses

Choosing the Target Function

- What type of knowledge will be learned?
- How will the knowledge be used by the performance program?
- E.g. checkers program
  - Assume it knows legal moves
  - Needs to choose best move
  - So learn function: F: Boards -> Moves
    - hard to learn
  - Alternative: F: Boards -> R

The Ideal Evaluation Function

- V(b) = 100 if b is a final, won board
- V(b) = -100 if b is a final, lost board
- V(b) = 0 if b is a final, drawn board
- Otherwise, if b is not final
  - V(b) = V(s) where s is best, reachable final board

Nonoperational…

Want operational approximation of V: \( \hat{V} \)

How Represent Target Function

- \( x_1 \) = number of black pieces on the board
- \( x_2 \) = number of red pieces on the board
- \( x_3 \) = number of black kings on the board
- \( x_4 \) = number of red kings on the board
- \( x_5 \) = num of black pieces threatened by red
- \( x_6 \) = num of red pieces threatened by black

\[ \hat{V}(b) = a + bx_1 + cx_2 + dx_3 + ex_4 + fx_5 + gx_6 \]

Now just need to learn 7 numbers!

Example: Checkers

- Task T:
  - Playing checkers
- Performance Measure P:
  - Percent of games won against opponents
- Experience E:
  - Playing practice games against itself
- Target Function
  - \( V: \text{board} \rightarrow R \)
  - Representation of approx. of target function

\[ \hat{V}(b) = a + bx_1 + cx_2 + dx_3 + ex_4 + fx_5 + gx_6 \]
Target Function

- Profound Formulation: Can express any type of inductive learning as approximating a function.
- E.g., Checkers: $V: \text{boards} \rightarrow \text{evaluation}$
- E.g., Handwriting recognition: $V: \text{image} \rightarrow \text{word}$
- E.g., Mushrooms: $V: \text{mushroom-attributes} \rightarrow \{E, P\}$

More Examples

- Collaborative Filtering:
  Eg, when you look at book B in Amazon
  It says "Buy B and also book C together & save!"

- Automatic Steering

More Examples

- Given: Training examples $(x_i, f(x_i))$ for some unknown function $f$.
- Find: A good approximation to $f$.

Example Applications

- Credit risk assessment: $x$: Financial information; $f(x)$: Credit risk
- Disease diagnosis: $x$: Symptoms and medical history; $f(x)$: Disease
- Voice recognition: $x$: Audio signal; $f(x)$: Speaker
- Face recognition: $x$: Image of a face; $f(x)$: Person

Supervised Learning

- Inductive learning or "Prediction": Given examples of a function $(X, F(X))$ Predict function $F(X)$ for new examples $X$
  - Classification
    $F(X) = \text{Discrete}$
  - Regression
    $F(X) = \text{Continuous}$
  - Probability estimation
    $F(X) = \text{Probability}(X)$

Why is Learning Possible?

Experience alone never justifies any conclusion about any unseen instance.

Learning occurs when PREJUDICE meets DATA!

Bias

- The nice word for prejudice is "bias".
- What kind of hypotheses will you consider?
  What is allowable range of functions you use when approximating?
- What kind of hypotheses do you prefer?
Some Typical Bias

The world is simple

Occam's razor

"It is needless to do more when less will suffice"

- William of Occam,
died 1349 of the Black plague

MDL - Minimum description length

Concepts can be approximated by
- conjunctions of predicates
- by linear functions
- by short decision trees

A Learning Problem

### Hypothesis Spaces

- **Complete Ignorance:** There are $2^n = 65536$ possible boolean functions over four input features. We can't check which one is correct until we've seen every possible input-output pair. After 7 examples, we still have 32 possibilities.

### Terminology

- **Training examples:** An example of the form $(x, f(x))$.
- **Target function (target concept):** The true function $f$.
- **Hypothesis:** A proposed function $h$ believed to be similar to $f$.
- **Concept:** A boolean function. Examples for which $f(x) = 1$ are called positive examples or positive instances of the concept. Examples for which $f(x) = 0$ are called negative examples or negative instances.
- **Classifier:** A classifier-valued function. The possible values $f(x) \in \{1, \ldots, K\}$ are called the classes or class labels.
- **Hypothesis Space:** The space of all hypotheses that can, in principle, be output by a learning algorithm.
- **Version Space:** The space of all hypotheses in the hypothesis space that have not yet been ruled out by a training example.

### Two Strategies for ML

- **Restriction bias:** Use prior knowledge to specify a restricted hypothesis space.
  - Version space algorithm over conjunctions.
- **Preference bias:** Use a broad hypothesis space, but impose an ordering on the hypotheses.
  - Decision trees.
Key Issues

- What are good hypothesis spaces?
  What space has been useful in practical applications and why?
- What algorithms can work with these spaces?
  Are there general design principles for machine learning algorithms?
- How can we optimize accuracy on future data points?
  This is sometimes called the “problem of overfitting.”
- How can we have confidence in the results?
  How much training data is required to find accurate hypotheses? (the statistical question)
- Are some learning problems computationally intractable?
  (the computational question)
- How can we formulate application problems as machine learning problems? (the engineering question)

A Framework for ML

- Search Procedures:
  Directional Search: search for the hypothesis directly.
  Local Search: start with an initial hypothesis, make small improvements until a local optimum.
  Constructive Search: start with an empty hypothesis, gradually add structure to it until good optimum.
- Timing:
  Online: Analyze each training example and construct an explicit hypothesis.
  Lazy: Store the training data and wait until a test data point is presented, then construct an ad hoc hypothesis to classify that one data point.
- On-line vs. Batch (for inductive algorithms)
  Online: Analyze each training example as it is presented.
  Batch: Collect training examples, analyze them, output an hypothesis.