Chapter 1: Introduction

CS 536: Machine Learning
Littman (Wu, TA)

Outline

• Why Machine Learning?
• What is a well-defined learning problem?
• An example: Learning to play checkers
• What questions should we ask about Machine Learning?
Why Machine Learning

- Recent progress in algorithms and theory
- Growing flood of online data
- Computational power is available
- Budding industry

Three Niches for ML

- Data mining: using historical data to improve decisions
  - medical records  medical knowledge
- Software applications we can't program by hand
  - autonomous driving
  - speech recognition
- Self customizing programs
  - Newsreader that learns user interests
Typical Datamining Task

Given:
- 9714 patient records, each describing a pregnancy and birth
- Each patient record contains 215 features

Learn to predict:
- Classes of future patients at high risk for Emergency Cesarean Section

Patient Data

Patient103_{time=1} → Patient103_{time=2} → Patient103_{time=n}

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient103_{time=1}</td>
<td>Age: 23</td>
<td>FirstPregnancy: no</td>
<td>Anemia: no</td>
<td>Diabetes: YES</td>
<td>PreviousPrematureBirth: no</td>
<td>Ultrasound: abnormal</td>
<td>Elective C-Section: no</td>
</tr>
<tr>
<td>Patient103_{time=2}</td>
<td>Age: 23</td>
<td>FirstPregnancy: no</td>
<td>Anemia: no</td>
<td>Diabetes: no</td>
<td>PreviousPrematureBirth: no</td>
<td>Ultrasound: ?</td>
<td>Elective C-Section: no</td>
</tr>
<tr>
<td>Patient103_{time=n}</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
**Datamining Result**

One of 18 learned rules:

If No previous vaginal delivery, and Abnormal 2nd Trimester Ultrasound, and Malpresentation at admission

Then Probability of Emergency C-Section is 0.6

• Over training data: 26/41 = .63,
• Over test data: 12/20 = .60

---

**Credit Risk Analysis Data**

\[\text{Cust103}_{time=1} \rightarrow \text{Cust103}_{time=2} \rightarrow \text{Cust103}_{time=n}\]

- \(\text{Cust103}_{time=1}\):
  - Years of credit: 9
  - Loan balance: $2,400
  - Income: $52k
  - Own House: Yes
  - OtherDeliquentAccts: 2
  - Max billing cycles late: 3
  - Profitable customer?: ?

- \(\text{Cust103}_{time=2}\):
  - Years of credit: 9
  - Loan balance: $3,250
  - Income: ?
  - Own House: Yes
  - OtherDeliquentAccts: 2
  - Max billing cycles late: 4
  - Profitable customer?: ?

- \(\text{Cust103}_{time=n}\):
  - Years of credit: 9
  - Loan balance: $4,500
  - Income: ?
  - Own House: Yes
  - OtherDeliquentAccts: 3
  - Max billing cycles late: 6
  - Profitable customer?: No

...
Datamining Result

Rules learned from synthesized data:

If Other-Delinquent-Accounts > 2, and 
    Number-Delinquent-Billing-Cycles > 1
Then Profitable-Customer? = No
    [Deny Credit Card application]

If Other-Delinquent-Accounts = 0, and 
    (Income > $30k) OR (Years-of-Credit > 3)
Then Profitable-Customer? = Yes
    [Accept Credit Card application]

Customer Purchase Behavior

\[ \text{Cust103}_{time=1} \rightarrow \text{Cust103}_{time=2} \rightarrow \text{Cust103}_{time=n} \]

<table>
<thead>
<tr>
<th></th>
<th>\text{Cust103}_{time=1}</th>
<th>\rightarrow</th>
<th>\text{Cust103}_{time=2}</th>
<th>\rightarrow</th>
<th>\text{Cust103}_{time=n}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>M</td>
<td></td>
<td>M</td>
<td></td>
<td>M</td>
</tr>
<tr>
<td>Age</td>
<td>53</td>
<td></td>
<td>53</td>
<td></td>
<td>53</td>
</tr>
<tr>
<td>Income</td>
<td>$50k</td>
<td></td>
<td>$50k</td>
<td></td>
<td>$50k</td>
</tr>
<tr>
<td>Own House</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>MS Products</td>
<td>Word</td>
<td></td>
<td>Word</td>
<td></td>
<td>Word</td>
</tr>
<tr>
<td>Computer</td>
<td>386 PC</td>
<td></td>
<td>Pentium</td>
<td></td>
<td>Pentium</td>
</tr>
<tr>
<td>Purchase Excel?</td>
<td>?</td>
<td></td>
<td>?</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Customer Retention

\[\text{Cust103}_{\text{time}=1} \rightarrow \text{Cust103}_{\text{time}=2} \rightarrow \text{Cust103}_{\text{time}=n}\]

- **Sex:** M
- **Age:** 53
- **Income:** $50k
- **Own House:** Yes
- **Checking:** $5k
- **Savings:** $15k
- **Current-customer?**:
  - Yes

- **Sex:** M
- **Age:** 53
- **Income:** $50k
- **Own House:** Yes
- **Checking:** $20k
- **Savings:** $0k
- **Current-customer?**:
  - Yes

- **Sex:** M
- **Age:** 53
- **Income:** $50k
- **Own House:** Yes
- **Checking:** $0k
- **Savings:** $0k
- **Current-customer?**:
  - No

Process Optimization

\[\text{Product72}_{\text{time}=1} \rightarrow \text{Product72}_{\text{time}=2} \rightarrow \text{Product72}_{\text{time}=n}\]

- **Stage:** mix
- **Mixing-speed:** 60rpm
- **Viscosity:** 1.3
- **Fat content:** 15%
- **Density:** 2.6
- **Spectral peak:** 2800
- **Product underweight?**:
  - ?

- **Stage:** cook
- **Temperature:** 325
- **Viscosity:** 3.2
- **Fat content:** 12%
- **Density:** 1.1
- **Spectral peak:** 3200
- **Product underweight?**:
  - ?

- **Stage:** cool
- **Fan-speed:** medium
- **Viscosity:** 1.3
- **Fat content:** 12%
- **Density:** 1.2
- **Spectral peak:** 3100
- **Product underweight?**:
  - Yes
Too Difficult to Program

Problems too difficult to program by hand
ALVINGN [Pomerleau] drives 70mph on highways

Software Customizes to User

http://www.wisewire.com
April 30, 1998
Lycos Acquires WiseWire
By internetnews.com Staff
Lycos, Inc. today announced the acquisition of targeted-content provider WiseWire Corp. for around $39.75 million in Lycos shares.
Under the acquisition, the three-year old, Pittsburgh-based WiseWire's automated directory will be incorporated into Lycos’ search page. Lycos said it will be the first online service using the technology which is based on user input and an intelligent agent product.
Where Is This Headed?

Today: tip of the iceberg
• First-generation algorithms: neural nets, decision trees, regression ...
• Applied to well-formatted database
• Budding industry

Looking to Tomorrow

Opportunity for tomorrow: enormous impact
• Learn across full mixed-media data
• Learn across multiple internal databases, plus the web and newsfeeds
• Learn by active experimentation
• Learn decisions rather than predictions
• Cumulative, lifelong learning
• Programming languages with learning embedded?
Relevant Disciplines

• Artificial intelligence
• Bayesian methods
• Computational complexity theory
• Control theory
• Information theory
• Philosophy
• Psychology and neurobiology
• Statistics
• …

What is the Learning Problem?

Learning = Improving with experience at some task
• Improve over task $T$,
• with respect to performance measure $P$,
• based on experience $E$.
E.g., Learn to play checkers
• $T$: Play checkers
• $P$: % of games won in world tournament
• $E$: opportunity to play against self
Learning Curves

- $T$: parsing Korean
- $P$: “crossing brackets”
- $E$: training sentences

From “Rapid Parser Development: A Machine Learning Approach for Korea” by Ulf Hermjakob

crossings brackets per sentence

Other Kinds of “Learning”?

“Tamagotchi is a tiny pet from cyberspace who needs your love to survive and grow. If you take good care of your Tamagotchi pet, it will slowly grow bigger, healthier, and more beautiful every day. But if you neglect your little cyber creature, your Tamagotchi may grow up to be mean or ugly. How old will your Tamagotchi be when it returns to its home planet? What kind of virtual caretaker will you be?” - Bandai

http://www.mimitchi.com/html/q1.htm
Learning to Play Checkers

- $T$: Play checkers
- $P$: Percent of games won in world tournament
- What experience?
- What exactly should be learned?
- How shall it be represented?
- What specific algorithm to learn it?

Type of Training Experience

- Direct or indirect?
- Teacher or not?

A problem: is training experience representative of performance goal?
Choose the Target Function

• ChooseMove: Board ⊢ Move ??
• V: Board ⊢ ??
• ...

Possible Definition for V

• if \( b \) is a final board state that is won, then \( V(b) = 100 \)
• if \( b \) is a final board state that is lost, then \( V(b) = -100 \)
• if \( b \) is a final board state that is drawn, then \( V(b) = 0 \)
• if \( b \) is a not a final state in the game, then \( V(b) = V(b') \), where \( b' \) is the best final board state that can be achieved starting from \( b \) and playing optimally until the end of the game.

This gives correct values, but is not operational.
Target Function Representation

- collection of rules?
- neural network?
- polynomial function of board features?
- ...

A Linear Representation

\[ V(b) = \sum \frac{w}{w_{0} + \frac{w_{1}}{bp(b)} + \frac{w_{2}}{rp(b)} + \frac{w_{3}}{bk(b)}} \]
\[ + \frac{w_{4}}{rk(b)} + \frac{w_{5}}{bt(b)} + \frac{w_{6}}{rt(b)} \]
\[ = w \cdot f(b) \]

- \( bp(b) \): number of black pieces on board \( b \)
- \( rp(b) \): number of red pieces on \( b \)
- \( bk(b) \): number of black kings on \( b \)
- \( rk(b) \): number of red kings on \( b \)
- \( bt(b) \): number of red pieces threatened by black (i.e., which can be taken on black's next turn)
- \( rt(b) \): number of black pieces threatened by red
Obtaining Training Examples

- \( V(b) \): the true target function
- \( \hat{V}(b) \): the learned function
- \( V_{\text{train}}(b) \): the training value

One rule for estimating training values:

- \( V_{\text{train}}(b) \geq \hat{V}(\text{Successor}(b)) \)

What’s a Successor?

From the textbook:

- \( \text{Successor}(b) \)

Alternative:

- \( \text{Successor}(b) \)
Choose Weight Tuning Rule

LMS Weight update rule:
Do repeatedly:
• Select a training example $b$ at random
  1. Compute $\text{error}(b)$:
     \[
     \text{error}(b) = V_{\text{train}}(b) - \hat{V}(b)
     \]
  2. For each board feature $f_i(b)$, update weight $w_i$:
     \[
     w_i \leftarrow w_i + c \cdot f_i(b) \cdot \text{error}(b)
     \]
     $c$ is some small constant, say 0.1, to moderate the rate of learning.

Design Choices
Some Issues in ML

- What algorithms can approximate functions well (and when)?
- How does number of training examples influence accuracy?
- How does complexity of hypothesis representation impact it?
- How does noisy data influence accuracy?
- What are the theoretical limits of learnability?
- How can prior knowledge of learner help?
- What clues can we get from biological learning systems?
- How can systems alter their own representations?