1 Machine Learning – Introduction

- 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
Why Machine Learning?

▶ Recent progress in algorithms and theory

▶ Growing flood of online data

▶ Computational power is available

▶ Budding industry

▶ Niches for machine learning:
  ▶ 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

▶ Data:

<table>
<thead>
<tr>
<th>Time</th>
<th>Patient103</th>
<th>Age</th>
<th>FirstPregnancy</th>
<th>Anemia</th>
<th>Diabetes</th>
<th>PreviousPrematureBirth</th>
<th>Ultrasound</th>
<th>Elective C−Section</th>
<th>Emergency C−Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Patient103</td>
<td>23</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>?</td>
<td>no</td>
<td>?</td>
</tr>
<tr>
<td>2</td>
<td>Patient103</td>
<td>23</td>
<td>no</td>
<td>no</td>
<td>YES</td>
<td>no</td>
<td>abnormal</td>
<td>no</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

▶ 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
Datamining Result

Data:

Patient103 time=1 → Patient103 time=2 → ... → Patient103 time=n

Age: 23
FirstPregnancy: no
Anemia: no
Diabetes: no
PreviousPrematureBirth: no
Ultrasound: ?
Elective C−Section: ?
Emergency C−Section: ?
...
Age: 23
FirstPregnancy: no
Anemia: no
Diabetes: YES
PreviousPrematureBirth: no
Ultrasound: abnormal
Elective C−Section: no
Emergency C−Section: ?
...
Age: 23
FirstPregnancy: no
Anemia: no
Diabetes: no
PreviousPrematureBirth: no
Ultrasound: ?
Elective C−Section: no
Emergency C−Section: Yes
...

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:

- Customer103: (time=t0)
  - Years of credit: 9
  - Loan balance: $2,400
  - Income: $52k
  - Own House: Yes
  - Other delinquent accts: 2
  - Max billing cycles late: 3
  - Profitable customer?: ?
- Customer103: (time=t1)
  - Years of credit: 9
  - Loan balance: $3,250
  - Income: ?
  - Own House: Yes
  - Other delinquent accts: 2
  - Max billing cycles late: 4
  - Profitable customer?: ?
- Customer103: (time=tn)
  - Years of credit: 9
  - Loan balance: $4,500
  - Income: ?
  - Own House: Yes
  - Other delinquent accts: 3
  - Max billing cycles late: 6
  - Profitable customer?: No

► 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]
Other Prediction Problems

Customer purchase behavior:

Customer103:  
\begin{align*}
\text{Sex}: & M \\
\text{Age}: & 53 \\
\text{Income}: & $50k \\
\text{Own House}: & Yes \\
\text{MS Products}: & \text{Word} \\
\text{Computer}: & 386 \text{ PC} \\
\text{Purchase Excel?:} & ? \\
\end{align*}

...  

Customer103:  
\begin{align*}
\text{Sex}: & M \\
\text{Age}: & 53 \\
\text{Income}: & $50k \\
\text{Own House}: & Yes \\
\text{MS Products}: & \text{Word} \\
\text{Computer}: & \text{Pentium} \\
\text{Purchase Excel?:} & ? \\
\end{align*}

...  

Customer103:  
\begin{align*}
\text{Sex}: & M \\
\text{Age}: & 53 \\
\text{Income}: & $50k \\
\text{Own House}: & Yes \\
\text{MS Products}: & \text{Word} \\
\text{Computer}: & \text{Pentium} \\
\text{Purchase Excel?:} & \text{Yes} \\
\end{align*}

Process optimization:

Product72:  
\begin{align*}
\text{Stage}: & \text{mix} \\
\text{Mixing−speed}: & 60\text{rpm} \\
\text{Viscosity}: & 1.3 \\
\text{Fat content}: & 15\% \\
\text{Density}: & 2.8 \\
\text{Spectral peak}: & 2800 \\
\text{Product underweight?:} & ?? \\
\end{align*}

...  

Product72:  
\begin{align*}
\text{Stage}: & \text{cook} \\
\text{Temperature}: & 325 \\
\text{Viscosity}: & 3.2 \\
\text{Fat content}: & 12\% \\
\text{Density}: & 1.1 \\
\text{Spectral peak}: & 3200 \\
\text{Product underweight?:} & ?? \\
\end{align*}

...  

Product72:  
\begin{align*}
\text{Stage}: & \text{cool} \\
\text{Fan−speed}: & \text{medium} \\
\text{Viscosity}: & 1.3 \\
\text{Fat content}: & 12\% \\
\text{Density}: & 1.2 \\
\text{Spectral peak}: & 3100 \\
\text{Product underweight?:} & \text{Yes} \\
\end{align*}
Problems Too Difficult to Program by Hand

- ALVINN [Pomerleau] drives 70 mph on highways
Software that Customizes to User
Where Is this Headed?

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

► 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.

E.g., Learning to drive:

- $T$: driving on public four-lane highway using vision sensors,
- $P$: average distance travelled before an error,
- $E$: a sequence of images and steering commands recorded while observing a human driver.
Learning to Play Checkers

- $T$: Play checkers

- $P$: Percent of games won in world tournament
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?
- The problem of credit assignment.
- Teacher or not?
Type of Training Experience

Direct or indirect?

The problem of credit assignment.

Teacher or not?

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

- ChooseMove : Board → Move
- V : Board → \( \mathbb{R} \)
- ...

1 Machine Learning – Introduction 14
Possible Definition for Target Function $V$

- 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.

Ultimate goal: Find an operational description of the ideal target function $V$.

But we can often only acquire some approximation $\hat{V}$.
Choose Representation for Target Function

▶ collection of rules?

▶ neural network?

▶ polynomial function of board features?

▶ ...
A Representation for Learned Function

\[ w_0 + w_1 \cdot bp(b) + w_2 \cdot rp(b) + w_3 \cdot bk(b) + w_4 \cdot rk(b) + w_5 \cdot bt(b) + w_6 \cdot rt(b) \]

- \(bp(b)\): 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
- $V_{train}(b)$: the training value
Obtaining Training Examples

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

One rule for estimating training values:

- $V_{train}(b) \leftarrow \hat{V}(Successor(b))$
Choose Weight Tuning Rule

- LMS Weight update rule:

- Do repeatedly:
  - Select a training example \( b \) at random
  - Compute \( \text{error}(b) \):
    \[
    \text{error}(b) = V_{\text{train}}(b) - \hat{V}(b)
    \]
  - For each board feature \( f_i \), update weight \( w_i \):
    \[
    w_i \leftarrow w_i + c \cdot f_i \cdot \text{error}(b)
    \]

- \( c \) is some small constant, say 0.1, to moderate the rate of learning
Design Choices

1. Determine Target Function
   - Board move
   - Board value

2. Determine Type of Training Experience
   - Games against self
   - Games against experts
   - Table of correct moves

3. Determine Representation of Learned Function
   - Polynomial
   - Linear function of six features
   - Artificial neural network

4. Determine Learning Algorithm
   - Gradient descent
   - Linear programming

Completed Design
Some Issues in Machine Learning

- 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?