Lecture 0 of 42

Machine Learning / Advanced Topics in AI
Course Organization and Survey

Friday, 18 January 2008

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course pages: http://snipurl.com/1y5qc / http://snipurl.com/1y5ih
Course web site: http://www.kddresearch.org/Courses/Spring-2008/CIS732
Instructor home page: http://www.cis.ksu.edu/~bhsu

Reading for Next Class:
Syllabus and Course Intro
Handout: Chapters 1-2, Mitchell

Course Administration

- Course Pages (KSOL): http://snipurl.com/1y5qc / http://snipurl.com/1y5ih
- Class Web Page: www.kddresearch.org/Courses/Spring-2008/CIS732
- Instructional E-Mail Addresses (for Topics in AI, substitute 830 for 732)
  - CIS732TA-L@listserv.ksu.edu (always use this to reach instructor and TA)
  - CIS732-L@listserv.ksu.edu (this goes to everyone)
- Instructor: William Hsu, Nichols 213
  - Office phone: +1 785 532 7905; home phone: +1 785 539 7180
  - IM: AIM/MSN/YIM hsuwh/rizanabsith, ICQ 28651394/191317559, Google banazir
  - Office hours: after class Mon/Wed/Fri; other times by appointment
- Graduate Teaching Assistant: Jing Xia
  - Office location: Nichols 213a
  - Office hours: to be announced on class web board
- Grading Policy
  - Midterm: 15% (in-class, open-book); final (take-home): 20%
  - Machine problems, problem sets (6 of 8): 30%; term project: 20%
  - Paper reviews (10 of 12): 10%; class participation: 5% (HW, Q&A)
Class Resources

- Web Page (Required)
  - Lecture notes (MS PowerPoint 97-2003, PDF)
  - Homeworks (MS PowerPoint 97-2003, PDF)
  - Exam and homework solutions (MS PowerPoint 97-2003, PDF)
  - Class announcements (students’ responsibility) and grade postings
- Course Notes at Copy Center (Required)
- Course Web Group
  - Mirror of announcements from class web page
  - Discussions (instructor and other students)
  - Dated research announcements (seminars, conferences, calls for papers)
- Mailing List (Automatic)
  - [CIS732-L@listserv.ksu.edu](mailto:CIS732-L@listserv.ksu.edu)
  - Sign-up sheet
  - Reminders, related research announcements

Course Overview

- Learning Algorithms and Models
  - Models: decision trees, winnow, artificial neural networks, naïve Bayes, genetic algorithms (GAs) and genetic programming (GP), instance-based learning (nearest-neighbor), inductive logic programming (ILP)
  - Algorithms: for decision trees (ID3/C4.5/J48), ANNs (backprop), etc.
  - Methodologies: supervised, unsupervised, reinforcement; knowledge-guided
- Theory of Learning
  - Computational learning theory (COLT): complexity, limitations of learning
  - Probably Approximately Correct (PAC) learning
  - Probabilistic, statistical, information theoretic results
- Multistrategy Learning: Combining Techniques, Knowledge Sources
- Data: Time Series, Very Large Databases (VLDB), Text Corpora
- Applications
  - Performance element: classification, decision support, planning, control
  - Database mining and knowledge discovery in databases (KDD)
  - Computer inference: learning to reason
Why Machine Learning?

- **New Computational Capability**
  - Database mining: converting records into knowledge
  - Self-customizing programs: learning news filters, adaptive monitors
  - Learning to act: robot planning, control optimization, decision support
  - Applications that are hard to program: automated driving, speech recognition

- **Better Understanding of Human Learning and Teaching**
  - Cognitive science: theories of knowledge acquisition (e.g., through practice)
  - Performance elements: reasoning (inference) and recommender systems

- **Time is Right**
  - Recent progress in algorithms and theory
  - Rapidly growing volume of online data from various sources
  - Available computational power
  - Growth, interest in learning-based industries (e.g., data mining/KDD)

---

Rule and Decision Tree Learning

- **Example: Rule Acquisition from Historical Data**

  Data

  - Patient 103 (time = 1): Age 23, First-Pregnancy: no, Anemia: no, Diabetes: no, Previous-Premature-Birth: no, Ultrasound: unknown, Elective C-Section: unknown, Emergency-C-Section: unknown
  - Patient 103 (time = 2): Age 23, First-Pregnancy: no, Anemia: no, Diabetes: yes, Previous-Premature-Birth: no, Ultrasound: unknown, Elective C-Section: no, Emergency-C-Section: unknown
  - Patient 103 (time = n): Age 23, First-Pregnancy: no, Anemia: no, Diabetes: yes, Previous-Premature-Birth: no, Ultrasound: abnormal, Elective C-Section: no, Emergency-C-Section: YES

- **Learned Rule**

  IF no previous vaginal delivery, AND abnormal 2nd trimester ultrasound, AND malpresentation at admission, AND no elective C-Section
  
  THEN probability of emergency C-Section is 0.6

  - Training set: 26/41 = 0.634
  - Test set: 12/20 = 0.600
Neural Network Learning

- **Autonomous Learning Vehicle In a Neural Net (ALVINN):** Pomerleau et al.
  - Drives 70mph on highways

© 1999, 2001 Carnegie Mellon University

Relevant Disciplines

- Artificial Intelligence
- Bayesian Methods
- Cognitive Science
- Computational Complexity Theory
- Control Theory
- Information Theory
- Neuroscience
- Philosophy
- Psychology
- Statistics

- PAC Formalism
- Mistake Bounds
- Optimization
- Learning Predictors
- Meta-Learning
- Entropy Measures
- MDL Approaches
- Optimal Codes
- Bayes’s Theorem
- Missing Data Estimators
- Language Learning
- Learning to Reason
- Bias/Variance Formalism
- Confidence Intervals
- Hypothesis Testing
- Power Law of Practice
- Heuristic Learning
- Occam’s Razor
- Inductive Generalization
- ANN Models
- Modular Learning
Specifying A Learning Problem

- Learning = Improving with Experience at Some Task
  - Improve over task \( T \),
  - with respect to performance measure \( P \),
  - based on experience \( E \).

- Example: Learning to Play Checkers
  - \( T \): play games of checkers
  - \( P \): percent of games won in tournament play
  - \( E \): opportunity to play against self

- Refining the Problem Specification: Issues
  - What experience?
  - What exactly should be learned?
  - How shall it be represented?
  - What specific algorithm to learn it?

- Defining the Problem Milieu
  - Performance element: How shall results of learning be applied?
  - How shall performance element be evaluated? Learning system?

Example: Learning to Play Checkers

- Type of Training Experience
  - Direct or indirect?
  - Teacher or not?
  - Knowledge about the game (e.g., openings/endgames)?

- Problem: Is Training Experience Representative (of Performance Goal)?

- Software Design
  - Assumptions of the learning system: legal move generator exists
  - Software requirements: generator, evaluator(s), parametric target function

- Choosing a Target Function
  - \( \text{ChooseMove}: \text{Board} \rightarrow \text{Move} \) – action selection function, or policy
  - \( V: \text{Board} \rightarrow \mathbb{R} \) – evaluation function for game tree search (minimax / \( \alpha-\beta \))
  - Ideal target \( V \); approximated target \( \hat{V} \)
  - Goal of learning process: operational description (approximation) of \( V \)

- \textbf{Chinook}: Checkers Solved by Game Tree Search (July 2007)

A Target Function for Learning to Play Checkers

- Possible Definition
  - If \( b \) is final board state that is won, then \( V(b) = +100 \) (or MAXINT)
  - If \( b \) is final board state that is lost, then \( V(b) = -100 \) (or MAXINT)
  - If \( b \) is final board state that is drawn, then \( V(b) = 0 \)
  - If \( b \) is not final board state in the game, then \( V(b) = V(b') \) where \( b' \) is best final board state that can be achieved starting from \( b \) and playing optimally until end
  - Correct values, but not operational

- Choosing a Representation for the Target Function
  - Collection of rules?
  - Neural network?
  - Polynomial function (e.g., linear, quadratic combination) of board features?
  - Other?

- A Representation for Learned Function
  \[
  V(b) = w_p + w_{b/p}(b) + w_{r/p}(b) + w_{b/k}(b) + w_{r/k}(b) + w_{b/t}(b) + w_{r/t}(b)
  \]
  \( b/p_r \) = number of black/red pieces; \( b/k_r \) = number of black/red kings
  \( b/t_r \) = number of black/red pieces threatened (can be taken on next turn)

A Training Procedure for Learning to Play Checkers

- Obtaining Training Examples
  - \( V(b) \) target function
  - \( \hat{V}(b) \) learned function
  - \( V_{\text{train}}(b) \) training value ("signal")

- Rule For Estimating Training Values
  \[
  V_{\text{train}}(b) \leftarrow \hat{V}(\text{Successor}(b))
  \]

- Rule for Training (Weight Tuning)
  - Least Mean Square (LMS) weight update rule
  - \( \text{REPEAT} \)
    - Select training example \( b \) at random
    - Compute the \( \text{error}(b) \) for this training example
      \[
      \text{error}(b) = V_{\text{train}}(b) - \hat{V}(b)
      \]
    - For each board feature \( f_i \), update weight \( w_i \) as follows
      \[
      w_i \leftarrow w_i + c \cdot f_i \cdot \text{error}(b)
      \]
      where \( c \) is small, constant factor to adjust learning rate
Design Choices for Learning to Play Checkers

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

**Determine Target Function**
- Board → move
- Board → value

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

**Determine Learning Algorithm**
- Gradient descent
- Linear programming

**Completed Design**

Some Issues in Machine Learning

- What Algorithms Can Approximate Functions Well? When?
- How Do Learning System Design Factors Influence Accuracy?
  - Number of training examples
  - Complexity of hypothesis representation
- How Do Learning Problem Characteristics Influence Accuracy?
  - Noisy data
  - Multiple data sources
- What Are 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 Representation?
Interesting Applications

Database Mining

Clustering (Cartia ThemeScapes) - http://snurl.com/1y5l1

Reasoning (Inference, Decision Support)

6000 news stories from the WWW in 1997

Planning, Control

DC-ARM - http://www.stanford.edu/~dwilkins/members.htm

What to Learn?

- **Classification Functions**
  - Learning hidden functions: estimating (“fitting”) parameters
  - Concept learning (e.g., chair, face, game)
  - Diagnosis, prognosis: risk assessment, medical monitoring, security, ERP

- **Models**
  - Map (for navigation)
  - Distribution (query answering, aka QA)
  - Language model (e.g., automaton/grammar)

- **Skills**
  - Playing games
  - Planning
  - Reasoning (acquiring representation to use in reasoning)

- **Cluster Definitions for Pattern Recognition**
  - Shapes of objects
  - Functional or taxonomic definition

- **Many Problems Can Be Reduced to Classification**
How to Learn It?

- **Supervised**
  - What is learned? Classification function; other models
  - Inputs and outputs? Learning: examples \( \langle x, f(x) \rangle \rightarrow \text{approximation} \hat{f}(x) \)
  - How is it learned? Presentation of examples to learner (by teacher)

- **Unsupervised**
  - Cluster definition, or vector quantization function (codebook)
  - Learning: observations \( x \times \text{distance metric} \ d(x_1, x_2) \rightarrow \text{discrete codebook} \ f(x) \)
  - Formation, segmentation, labeling of clusters based on observations, metric

- **Reinforcement**
  - Control policy (function from states of the world to actions)
  - Learning: state/reward sequence \( \langle s_i, r_i \rangle \; : \; 1 \leq i \leq n \) → policy \( p : s \rightarrow a \)
  - (Delayed) feedback of reward values to agent based on actions
  - Model updated based on reward, (partially) observable state

Supervised Inductive Learning:
Classification and Regression

- Given: Training Examples \( \langle x, f(x) \rangle \) of Some Unknown Function \( f \)
- Find: A Good Approximation to \( f \)
- Examples (besides Concept Learning)
  - Disease diagnosis
    - \( x \) = properties of patient (medical history, symptoms, lab tests)
    - \( f \) = disease (or recommended therapy)
  - Risk assessment
    - \( x \) = properties of consumer, policyholder (demographics, accident history)
    - \( f \) = risk level (expected cost)
  - Automatic steering
    - \( x \) = bitmap picture of road surface in front of vehicle
    - \( f \) = degrees to turn the steering wheel
  - Part-of-speech tagging
  - Computer security: fraud/intrusion detection, attack graphs
  - Information extraction: clusters of documents
  - Social networks and weblogs: predicting links, sentiment analysis
  - Multisensor integration and prediction
Learning and Types [1]:
A Generic Supervised Learning Problem

\[ y = f(x_1, x_2, x_3, x_4) \]

<table>
<thead>
<tr>
<th>Example</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(x_3)</th>
<th>(x_4)</th>
<th>(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- Input: \(x_i\), desired output: \(y\), “target” function: \(f: (t_1 \times t_2 \times t_3 \times t_4) \rightarrow t\)
- Learning function: Vector \((t_1 \times t_2 \times t_3 \times t_4 \times t) \rightarrow (t_1 \times t_2 \times t_3 \times t_4) \rightarrow t\)

Example:

<table>
<thead>
<tr>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(x_3)</th>
<th>(x_4)</th>
<th>(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
</tr>
</tbody>
</table>

Learning and Types [2]:
Unrestricted Hypothesis Space

- \(|A \rightarrow B| = |B|^{|A|}\) – proof?
- \(|H^4 \rightarrow H| = |\{0,1\} \times \{0,1\} \times \{0,1\} \times \{0,1\} \rightarrow \{0,1\}| = 2^4 = 65536\) functions
- Complete Ignorance: Is Learning Possible?
  - Need to see every possible input/output pair
  - After 7 examples, still have \(2^7 = 512\) possibilities (out of 65536) for \(f\)
### Training Examples for Concept \textit{EnjoySport}

- **Specification for Examples**
  - Similar to a data type definition
  - 6 attributes: Sky, Temp, Humidity, Wind, Water, Forecast
  - Nominal-valued (symbolic) attributes - enumerative data type
- **Binary (Boolean-Valued or \(H\)-Valued) Concept**
- **Supervised Learning Problem:** \textit{Describe the General Concept}

<table>
<thead>
<tr>
<th>Example</th>
<th>Sky</th>
<th>AirTemp</th>
<th>Humidity</th>
<th>Wind</th>
<th>Water</th>
<th>Forecast</th>
<th>EnjoySport</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Sunny</td>
<td>Warm</td>
<td>Normal</td>
<td>Strong</td>
<td>Warm</td>
<td>Same</td>
<td>Yes</td>
</tr>
<tr>
<td>1</td>
<td>Sunny</td>
<td>Warm</td>
<td>High</td>
<td>Strong</td>
<td>Warm</td>
<td>Same</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Rainy</td>
<td>Cold</td>
<td>High</td>
<td>Strong</td>
<td>Warm</td>
<td>Change</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Sunny</td>
<td>Warm</td>
<td>High</td>
<td>Strong</td>
<td>Cool</td>
<td>Change</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\[
\text{Example Hypothesis for } \text{EnjoySport} = \text{Sky} \land \text{AirTemp} \land \text{Humidity} \land \text{Wind} \land \text{Water} \land \text{Forecast} \\
\text{``Sunny ? ? Strong ? Same''}
\]

- Is this consistent with the training examples?
- What are some hypotheses that are consistent with the examples?
Summary

- **Reading:** Chapters 1-2, Mitchell
- **Suggested Exercises:** 2.2, 2.3, 2.4, 2.6
- **Taxonomy of Learning Systems**
- **Week 1:** Overview, Learning from Examples
  - (Supervised) concept learning framework
  - Simple approach: assumes no noise; illustrates key concepts
- **Week 2:** Hypothesis Learning and Inductive Bias
  - Sources: Mitchell (1997) – online notes and handout
  - Wednesday: inductive learning, version space
  - Friday: candidate elimination algorithm, active learning, inductive bias
  - Background concepts: partially-ordered set (poset) formalism
- **Week 3:** Decision Trees, Intro Computational Learning Theory (COLT)
  - First paper review due Wed 30 Jan 2008

Terminology

- **Learning:** Improving at Task given Performance Measure, Experience
- **Performance Element:** Part of System that Applies Result of Learning
- **Types of Learning**
  - **Supervised:** with “teacher” (often, classification from labeled examples)
  - **Unsupervised:** from data, using similarity measure (unlabeled instances)
  - **Reinforcement:** “by doing”, with reward/punishment signal
- **Supervised Learning: Target Functions**
  - **Target function** – function $c$ or $f$ to be learned
  - **Target** – desired value $y$ to be predicted (sometimes “target function”)
  - **Example / labeled instance** – tuples of the form $<x, f(x)>$
  - **Classification function, classifier** – nominal-valued $f$ (enumerated return type)
- **Clustering:** Application of Unsupervised Learning
- **Concepts and Hypotheses**
  - **Concept** – function $c$ from observations to TRUE or FALSE (membership)
  - **Class label** – output of classification function
  - **Hypothesis** – proposed function $h$ believed to be similar to $c$ (or $f$)