Lecture 16 of 42

Intro to Genetic Algorithms (continued) and Bayesian Preliminaries

Wednesday, 21 February 2007

William H. Hsu
Department of Computing and Information Sciences, KSU
http://www.kddresearch.org

Readings:
Sections 6.1-6.5, Mitchell

Lecture Outline

• Read Sections 6.1-6.5, Mitchell
• Overview of Bayesian Learning
  – Framework: using probabilistic criteria to generate hypotheses of all kinds
  – Probability: foundations
• Bayes’s Theorem
  – Definition of conditional (posterior) probability
  – Ramifications of Bayes’s Theorem
    • Answering probabilistic queries
    • MAP hypotheses
• Generating Maximum A Posteriori (MAP) Hypotheses
• Generating Maximum Likelihood Hypotheses
• Next Week: Sections 6.6-6.13, Mitchell; Roth; Pearl and Verma
  – More Bayesian learning: MDL, BOC, Gibbs, Simple (Naïve) Bayes
  – Learning over text
Simple Genetic Algorithm (SGA)

- Algorithm Simple-Genetic-Algorithm (Fitness, Fitness-Threshold, p, r, m)
  
  \[
  P \leftarrow p \text{ random hypotheses} \quad // \text{initialize population}
  \]
  
  \[
  \text{FOR each } h \text{ in } P \text{ DO } f[h] \leftarrow \text{Fitness}(h) \quad // \text{evaluate Fitness: hypothesis } \rightarrow \mathbb{R}
  \]
  
  \[
  \text{WHILE } (\text{Max}(f) < \text{Fitness-Threshold}) \text{ DO}
  \]
  
  1. Select: Probabilistically select \((1 - r)p\) members of \(P\) to add to \(P_s\)
  
  \[
  P(h) = \frac{f[h]}{\sum_i f[h_i]}
  \]
  
  2. Crossover:
  
  - Probabilistically select \((r \cdot p)/2\) pairs of hypotheses from \(P\)
  
  \[
  P_s \leftarrow \text{Crossover}(<h_{i1}, h_{i2}>) \quad // P_{s}[t+1] = P_s[t] + <\text{offspring}_1, \text{offspring}_2>
  \]
  
  3. Mutate: Invert a randomly selected bit in \(m \cdot p\) random members of \(P_s\)
  
  4. Update: \(P \leftarrow P_s\)
  
  5. Evaluate: FOR each \(h\) in \(P\) DO \(f[h] \leftarrow \text{Fitness}(h)\)
  
  RETURN the hypothesis \(h\) in \(P\) that has maximum fitness \(f[h]\)

GA-Based Inductive Learning (GABIL)

- GABIL System [Dejong et al, 1993]
  - Given: concept learning problem and examples
  - Learn: disjunctive set of propositional rules
  - Goal: results competitive with those for current decision tree learning algorithms (e.g., C4.5)

- Fitness Function: \(\text{Fitness}(h) = (\text{Correct}(h))^2\)

- Representation
  - Rules: IF \(a_1 = T \land a_2 = F\) THEN \(c = T\); IF \(a_2 = T\) THEN \(c = F\)
  
  Bit string encoding: \(a_1[10] \cdot a_2[01] \cdot c[1] \cdot a_1[11] \cdot a_2[10] \cdot c[0] = 100111100\)

- Genetic Operators
  - Want variable-length rule sets
  - Want only well-formed bit string hypotheses
Crossover: Variable-Length Bit Strings

- Basic Representation
  - Start with
    \[
    h_1 = (0, 01, 1, 11, 0, 0)
    \]
    \[
    h_2 = (0, 1, 0, 10, 01, 0)
    \]
  - Idea: allow crossover to produce variable-length offspring

- Procedure
  - 1. Choose crossover points for \(h_1\), e.g., after bits 1, 8
  - 2. Now restrict crossover points in \(h_2\) to those that produce bitstrings with well-defined semantics, e.g., \(<1, 3>, <1, 8>, <6, 8>\)

- Example
  - Suppose we choose \(<1, 3>\)
  - Result
    \[
    h_3 = (1, 10, 0)
    \]
    \[
    h_4 = (0, 01, 11, 10, 01, 0)
    \]

GABIL Extensions

- New Genetic Operators
  - Applied probabilistically
    - 1. Add Alternative: generalize constraint on \(a_i\) by changing a 0 to a 1
    - 2. Drop Condition: generalize constraint on \(a_i\) by changing every 0 to a 1

- New Field
  - Add fields to bit string to decide whether to allow the above operators
  - So now the learning strategy also evolves!
    - aka genetic wrapper
GABIL Results

- **Classification Accuracy**
  - Compared to symbolic rule/tree learning methods
    - C4.5 [Quinlan, 1993]
    - ID5R
    - AQ14 [Michalski, 1986]
  - Performance of GABIL comparable
    - Average performance on a set of 12 synthetic problems: 92.1% test accuracy
    - Symbolic learning methods ranged from 91.2% to 96.6%

- **Effect of Generalization Operators**
  - Result above is for GABIL without AA and DC
  - Average test set accuracy on 12 synthetic problems with AA and DC: 95.2%

Building Blocks (Schemas)

- **Problem**
  - How to characterize evolution of population in GA?
- **Goal**
  - Identify basic building block of GAs
  - Describe family of individuals
- **Definition: Schema**
  - String containing 0, 1, * ("don't care")
  - Typical schema: 10**0*
  - Instances of above schema: 101101, 100000, ...

- **Solution Approach**
  - Characterize population by number of instances representing each possible schema
  - \( m(s, t) \) = number of instances of schema \( s \) in population at time \( t \)
Selection and Building Blocks

- **Restricted Case: Selection Only**
  - \( f(t) \) = average fitness of population at time \( t \)
  - \( m(s, t) \) = number of instances of schema \( s \) in population at time \( t \)
  - \( \hat{u}(s, t) \) = average fitness of instances of schema \( s \) at time \( t \)

- **Quantities of Interest**
  - Probability of selecting \( h \) in one selection step
    \[ P(h) = \frac{f(h)}{\sum_{h_i} f(h_i)} \]
  - Probability of selecting an instance of \( s \) in one selection step
    \[ P(h \in s) = \frac{f(h)}{\sum_{h_i \in s} f(h_i)} \]
  - Expected number of instances of \( s \) after \( n \) selections
    \[ E[m(s, t + 1)] = \hat{u}(s, t) \cdot m(s, t) \]

Schema Theorem

- **Theorem**
  \[ E[m(s, t + 1)] \geq \frac{\hat{u}(s, t)}{f(t)} \cdot m(s, t) \cdot \left(1 - p_c \cdot \frac{d}{l-1}\right) \cdot \left(1 - p_m\right)^{d(s)} \]
  - \( m(s, t) \) = number of instances of schema \( s \) in population at time \( t \)
  - \( f(t) \) = average fitness of population at time \( t \)
  - \( \hat{u}(s, t) \) = average fitness of instances of schema \( s \) at time \( t \)
  - \( p_c \) = probability of single point crossover operator
  - \( p_m \) = probability of mutation operator
  - \( l \) = length of individual bit strings
  - \( d(s) \) = number of defined (non "*" or "0") bits in \( s \)
  - \( d(s) \) = distance between rightmost, leftmost defined bits in \( s \)

- **Intuitive Meaning**
  - “The expected number of instances of a schema in the population tends toward its relative fitness”
  - A fundamental theorem of GA analysis and design
Bayesian Learning

- Framework: Interpretations of Probability [Cheeseman, 1985]
  - Bayesian subjectivist view
    - A measure of an agent's belief in a proposition
    - Proposition denoted by random variable (sample space: range)
    - e.g., \( Pr(\text{Outlook} = \text{Sunny}) = 0.8 \)
  - Frequentist view: probability is the frequency of observations of an event
  - Logicist view: probability is inferential evidence in favor of a proposition

- Typical Applications
  - HCI: learning natural language; intelligent displays; decision support
  - Approaches: prediction; sensor and data fusion (e.g., bioinformatics)

- Prediction: Examples
  - Measure relevant parameters: temperature, barometric pressure, wind speed
  - Make statement of the form \( Pr(\text{Tomorrow's-Weather} = \text{Rain}) = 0.5 \)
  - College admissions: \( Pr(\text{Acceptance}) = p \)
    - Plain beliefs: unconditional acceptance (\( p = 1 \)) or categorical rejection (\( p = 0 \))
    - Conditional beliefs: depends on reviewer (use probabilistic model)

Two Roles for Bayesian Methods

- Practical Learning Algorithms
  - Naïve Bayes (aka simple Bayes)
  - Bayesian belief network (BBN) structure learning and parameter estimation
  - Combining prior knowledge (prior probabilities) with observed data
    - A way to incorporate background knowledge (BK), aka domain knowledge
    - Requires prior probabilities (e.g., annotated rules)

- Useful Conceptual Framework
  - Provides "gold standard" for evaluating other learning algorithms
    - Bayes Optimal Classifier (BOC)
    - Stochastic Bayesian learning: Markov chain Monte Carlo (MCMC)
  - Additional insight into Occam's Razor (MDL)
Probabilistic Concepts versus Probabilistic Learning

- Two Distinct Notions: Probabilistic Concepts, Probabilistic Learning

- Probabilistic Concepts
  - Learned concept is a function, \( c: X \rightarrow [0, 1] \)
  - \( c(x) \), the target value, denotes the probability that the label 1 (i.e., True) is assigned to \( x \)
  - Previous learning theory is applicable (with some extensions)

- Probabilistic (i.e., Bayesian) Learning
  - Use of a probabilistic criterion in selecting a hypothesis \( h \)
    - e.g., “most likely” \( h \) given observed data \( D \): MAP hypothesis
    - e.g., \( h \) for which \( D \) is “most likely”: max likelihood (ML) hypothesis
    - May or may not be stochastic (i.e., search process might still be deterministic)
  - NB: \( h \) can be deterministic (e.g., a Boolean function) or probabilistic

Probability: Basic Definitions and Axioms

- Sample Space (\( \Omega \)): Range of a Random Variable \( X \)
- Probability Measure \( \Pr(\cdot) \)
  - \( \Omega \) denotes a range of “events”; \( X: \Omega \)
  - Probability \( \Pr \), or \( P \), is a measure over \( \Omega \)
  - In a general sense, \( \Pr(X = x \in \Omega) \) is a measure of belief in \( X = x \)
    - \( \Pr(X = x) = 0 \) or \( \Pr(X = x) = 1 \): plain (aka categorical) beliefs (can’t be revised)
    - All other beliefs are subject to revision

Kolmogorov Axioms

1. \( \forall x \in \Omega : 0 \leq \Pr(X = x) \leq 1 \)
2. \( \Pr(\Omega) = \sum_{x \in \Omega} \Pr(X = x) = 1 \)
3. \( \forall X_1, X_2, \ldots, i \neq j \Rightarrow X_i \land X_j = \emptyset \)
\[
\Pr\left(\bigcup_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \Pr(X_i)
\]

- Joint Probability: \( \Pr(X_1 \land X_2) = \Pr(X_1) \land \Pr(X_2) \)
- Independence: \( \Pr(X \land X_2) = \Pr(X_1) \land \Pr(X_2) \land \Pr(X_3) \land \Pr(X_4) \land \Pr(X_5) \)
Bayes’s Theorem

- **Theorem**
  \[ P(h | D) = \frac{P(D | h) P(h)}{P(D)} = \frac{P(h \land D)}{P(D)} \]

- **\( P(h) \)** = Prior Probability of Hypothesis \( h \)
  - Measures initial beliefs (BK) before any information is obtained (hence prior)

- **\( P(D) \)** = Prior Probability of Training Data \( D \)
  - Measures probability of obtaining sample \( D \) (i.e., expresses \( D \))

- **\( P(h | D) \)** = Probability of \( h \) Given \( D \)
  - \( | \) denotes conditioning - hence \( P(h | D) \) is a conditional (aka posterior) probability

- **\( P(D | h) \)** = Probability of \( D \) Given \( h \)
  - Measures probability of observing \( D \) given that \( h \) is correct (“generative” model)

- **\( P(h \land D) \)** = Joint Probability of \( h \) and \( D \)
  - Measures probability of observing \( D \) and of \( h \) being correct

Choosing Hypotheses

- **Bayes’s Theorem**
  \[ P(h | D) = \frac{P(D | h) P(h)}{P(D)} = \frac{P(h \land D)}{P(D)} \]

- **MAP Hypothesis**
  - Generally want most probable hypothesis given the training data
  - Define: \( \arg \max_{x \in \Omega} f(x) \) = the value of \( x \) in the sample space \( \Omega \) with the highest \( f(x) \)
  - Maximum a posteriori hypothesis, \( h_{MAP} \)
    \[ h_{MAP} = \arg \max_{h} P(h | D) \]
    \[ = \arg \max_{h} \frac{P(D | h) P(h)}{P(D)} \]
    \[ = \arg \max_{h} P(D | h) P(h) \]

- **ML Hypothesis**
  - Assume that \( p(h_j) = p(h_i) \) for all pairs \( i, j \) (uniform priors, i.e., \( P_p \sim \text{Uniform} \))
  - Can further simplify and choose the maximum likelihood hypothesis, \( h_{ML} \)
    \[ h_{ML} = \arg \max_{h_j \in H} P(D | h_j) \]
Bayes’s Theorem: Query Answering (QA)

- **Answering User Queries**
  - Suppose we want to perform intelligent inferences over a database DB
    - Scenario 1: DB contains records (instances), some "labeled" with answers
    - Scenario 2: DB contains probabilities (annotations) over propositions
  - QA: an application of probabilistic inference

- **QA Using Prior and Conditional Probabilities: Example**
  - Query: *Does patient have cancer or not?*
  - Suppose: patient takes a lab test and result comes back positive
    - Correct + result in only 98% of the cases in which disease is actually present
    - Correct - result in only 97% of the cases in which disease is not present
    - Only 0.008 of the entire population has this cancer
  - \( \alpha \equiv P(\text{false negative for } H_0 \equiv \text{Cancer}) = 0.02 \) (NB: for 1-point sample)
  - \( \beta \equiv P(\text{false positive for } H_0 \equiv \text{Cancer}) = 0.03 \) (NB: for 1-point sample)
  - \( P(\text{Cancer}) = 0.008 \) \( P(+ \mid \text{Cancer}) = 0.98 \) \( P(+) \neg \text{Cancer} ) = 0.03 \)
  - \( P(\neg \text{Cancer}) = 0.992 \) \( P(\neg \mid \neg \text{Cancer} ) = 0.02 \) \( P(\neg \mid \text{Cancer} ) = 0.97 \)
  - \( P(+ \mid H_0) P(H_0) = 0.0078 \) \( P(+ \mid H_1) P(H_1) = 0.0298 \Rightarrow h_{MAP} = H_1 = \neg \text{Cancer} \)

Basic Formulas for Probabilities

- **Product Rule** (Alternative Statement of Bayes’s Theorem)
  \[ P(A \mid B) = \frac{P(A \land B)}{P(B)} \]
  - Proof: requires axiomatic set theory, as does Bayes’s Theorem

- **Sum Rule**
  \[ P(A \lor B) = P(A) + P(B) - P(A \land B) \]
  - Sketch of proof (immediate from axiomatic set theory)
    - Draw a Venn diagram of two sets denoting events A and B
    - Let \( A \lor B \) denote the event corresponding to \( A \lor B \)

- **Theorem of Total Probability**
  - Suppose events \( A_1, A_2, \ldots, A_n \) are mutually exclusive and exhaustive
    - Mutually exclusive: \( i \neq j \Rightarrow A_i \land A_j = \emptyset \)
    - Exhaustive: \( \sum P(A_i) = 1 \)
  - Then \( P(B) = \sum P(B \mid A_i) P(A_i) \)
  - Proof: follows from product rule and 3rd Kolmogorov axiom
### MAP and ML Hypotheses: A Pattern Recognition Framework

- **Pattern Recognition Framework**
  - Automated speech recognition (ASR), automated image recognition
  - Diagnosis

- **Forward Problem: One Step in ML Estimation**
  - Given: model $h$, observations (data) $D$
  - Estimate: $P(D | h)$, the "probability that the model generated the data"

- **Backward Problem: Pattern Recognition / Prediction Step**
  - Given: model $h$, observations $D$
  - Maximize: $P(h(X) = x | h, D)$ for a new $X$ (i.e., find best $x$)

- **Forward-Backward (Learning) Problem**
  - Given: model space $H$, data $D$
  - Find: $h \in H$ such that $P(h | D)$ is maximized (i.e., MAP hypothesis)

- **More Info**
  - Emphasis on a particular $H$ (the space of hidden Markov models)

---

### Bayesian Learning Example: Unbiased Coin [1]

- **Coin Flip**
  - Sample space: $\Omega = \{Head, Tail\}$
  - Scenario: given coin is either fair or has a 60% bias in favor of Head
    - $h_1$: fair coin: $P(\text{Head}) = 0.5$
    - $h_2$: 60% bias towards Head: $P(\text{Head}) = 0.6$
  - Objective: to decide between default (null) and alternative hypotheses

- **A Priori (aka Prior) Distribution on $H$**
  - $P(h_1) = 0.75, P(h_2) = 0.25$
  - Reflects learning agent's prior beliefs regarding $H$
  - Learning is revision of agent's beliefs

- **Collection of Evidence**
  - First piece of evidence: $d$ = a single coin toss, comes up Head
  - $Q$: What does the agent believe now?
  - $A$: Compute $P(d) = P(d | h_1) P(h_1) + P(d | h_2) P(h_2)$
Bayesian Learning Example: Unbiased Coin [2]

• Bayesian Inference: Compute $P(d) = P(d \mid h_1) P(h_1) + P(d \mid h_2) P(h_2)$
  – $P(Head) = 0.5 \cdot 0.75 + 0.6 \cdot 0.25 = 0.375 + 0.15 = 0.525$
  – This is the probability of the observation $d = Head$

• Bayesian Learning
  – Now apply Bayes’s Theorem
    • $P(h_1 \mid d) = P(d \mid h_1) P(h_1) / P(d) = 0.375 / 0.525 = 0.714$
    • $P(h_2 \mid d) = P(d \mid h_2) P(h_2) / P(d) = 0.15 / 0.525 = 0.286$
    • Belief has been revised downwards for $h_1$, upwards for $h_2$
  – The agent still thinks that the fair coin is the more likely hypothesis
  – Suppose we were to use the ML approach (i.e., assume equal priors)
    • Belief is revised upwards from 0.5 for $h_1$
    • Data then supports the bias coin better

• More Evidence: Sequence $D$ of 100 coins with 70 heads and 30 tails
  – $P(D) = (0.5)^{50} \cdot (0.5)^{50} \cdot 0.75 + (0.6)^{70} \cdot (0.4)^{30} \cdot 0.25$
  – Now $P(h_1 \mid d) \ll P(h_2 \mid d)$

Brute Force MAP Hypothesis Learner

• Intuitive Idea: Produce Most Likely $h$ Given Observed $D$

• Algorithm Find-MAP-Hypothesis ($D$)
  – 1. FOR each hypothesis $h \in H$ Calculate the conditional (i.e., posterior) probability:
    
    $$P(h \mid D) = \frac{P(D \mid h) P(h)}{P(D)}$$

  – 2. RETURN the hypothesis $h_{MAP}$ with the highest conditional probability
    
    $$h_{MAP} = \arg \max_{h \in H} P(h \mid D)$$
Terminology

- **Evolutionary Computation (EC):** Models Based on Natural Selection
- **Genetic Algorithm (GA) Concepts**
  - **Individual:** single entity of model (corresponds to hypothesis)
  - **Population:** collection of entities in competition for survival
  - **Generation:** single application of selection and crossover operations
  - **Schema aka building block:** descriptor of GA population (e.g., 10**0**)
  - **Schema theorem:** representation of schema proportional to its relative fitness
- **Simple Genetic Algorithm (SGA) Steps**
  - **Selection**
    - Proportionate reproduction (aka roulette wheel): \( P(\text{individual}) \propto r(\text{individual}) \)
    - Tournament: let individuals compete in pairs or tuples; eliminate unfit ones
  - **Crossover**
    - Single-point: \( 11101001000 \times 00001010101 \rightarrow \{ 11101010101, 00001001000 \} \)
    - Two-point: \( 1110101000 \times 00001010101 \rightarrow \{ 11001011000, 00101000101 \} \)
    - Uniform: \( 1110101000 \times 00001010101 \rightarrow \{ 10001000100, 01101011001 \} \)
  - **Mutation:** single-point (“bit flip”), multi-point

Summary Points

- **Evolutionary Computation**
  - Motivation: process of natural selection
  - Limited population; individuals compete for membership
  - Method for parallelizing and stochastic search
  - Framework for problem solving: search, optimization, learning
- **Prototypical (Simple) Genetic Algorithm (GA)**
  - **Steps**
    - Selection: reproduce individuals probabilistically, in proportion to fitness
    - Crossover: generate new individuals probabilistically, from pairs of “parents”
    - Mutation: modify structure of individual randomly
  - How to represent hypotheses as individuals in GAs
- **An Example:** GA-Based Inductive Learning (GABIL)
- **Schema Theorem:** Propagation of Building Blocks
- **Next Lecture:** Genetic Programming, The Movie