

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

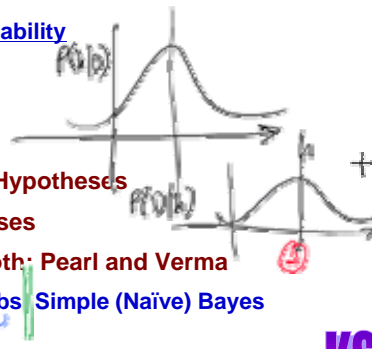


CIS 732: Machine Learning and Pattern Recognition

Kansas State University
Department of Computing and Information Sciences

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



CIS 732: Machine Learning and Pattern Recognition

Kansas State University
Department of Computing and Information Sciences

Simple Genetic Algorithm (SGA)

- Algorithm *Simple-Genetic-Algorithm* (*Fitness*, *Fitness-Threshold*, *p*, *r*, *m*)

// *p*: population size; *r*: replacement rate (aka generation gap width), *m*: string size

- $P \leftarrow p$ random hypotheses // initialize population
- FOR each h in P DO $f[h] \leftarrow \text{Fitness}(h)$ // evaluate *Fitness*: *hypothesis* $\rightarrow R$
- WHILE ($\text{Max}(f) < \text{Fitness-Threshold}$) DO

- 1. **Select**: Probabilistically select $(1 - r)p$ members of P to add to P_S

$$P(h_i) = \frac{f[h_i]}{\sum_{j=1}^p f[h_j]}$$

$h_i \in D$ where $\bar{x} \in D$

- 2. **Crossover**:

- Probabilistically select $(r \cdot p)/2$ pairs of hypotheses from P

- FOR each pair $\langle h_1, h_2 \rangle$ DO

$P_S += \text{Crossover}(\langle h_1, h_2 \rangle)$ // $P_S[t+1] = P_S[t] + \langle \text{offspring}_1, \text{offspring}_2 \rangle$

- 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 $a_1 = T \wedge a_2 = F$ THEN $c = T$; IF $a_2 = T$ THEN $c = F$

- Bit string encoding: $a_1 [10] \cdot a_2 [01] \cdot c [1] \cdot a_1 [11] \cdot a_2 [10] \cdot c [0] = 10011 11100$

$$1111 \dots 10 \dots 0$$

$$b_2 \leftarrow (b_1 \& m) | (b_2 \& \bar{m})$$

$$b_2 \leftarrow (b_1 \& \bar{m}) | (b_2 \& m)$$

- Genetic Operators**

- Want variable-length rule sets
- Want only well-formed bit string hypotheses

$$1111000 \dots 0111 \dots 1$$



Crossover: Variable-Length Bit Strings

- **Basic Representation**

- Start with

	a_1	a_2	c		a_1	a_2	c
h_1	1[0	01	1		11	1]0	0
h_2	0[1	1]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	11	10	0				
h_4	00	01	1	1	11	0	10 01 0



GABIL Extensions

- **New Genetic Operators**

- Applied probabilistically
- 1. AddAlternative: generalize constraint on a_i by changing a 0 to a 1
- 2. DropCondition: 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

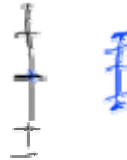
a_1	a_2	c		a_1	a_2	c	<u>AA</u>	<u>DC</u>
01	11	0		10	01	0	1	0

- So now the learning strategy also evolves!
- aka genetic wrapper



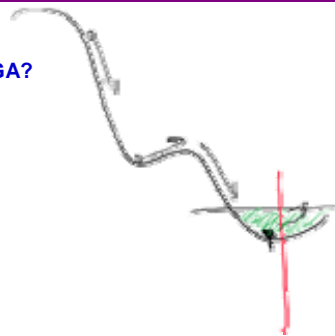
GABIL Results

- **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
 - $m(s, t) \equiv$ number of instances of schema s in population at time t



Selection and Building Blocks

- **Restricted Case: Selection Only**

- $\bar{f}(t)$ \equiv average fitness of population at time t
- $m(s, t)$ \equiv number of instances of schema s in population at time t
- $\hat{u}(s, t)$ \equiv average fitness of instances of schema s at time t

- **Quantities of Interest**

- Probability of selecting h in one selection step

$$P(h) = \frac{f(h)}{\sum_{i=1}^n f(h_i)}$$

- Probability of selecting an instance of s in one selection step

$$P(h \in s) = \sum_{h \in (s, p_i)} \frac{f(h)}{n \cdot \bar{f}(t)} = \frac{\hat{u}(s, t)}{n \cdot \bar{f}(t)} \cdot m(s, t)$$

- Expected number of instances of s after n selections

$$E[m(s, t+1)] = \frac{\hat{u}(s, t)}{\bar{f}(t)} \cdot m(s, t)$$



Schema Theorem

- **Theorem**

$$E[m(s, t+1)] \geq \frac{\hat{u}(s, t)}{\bar{f}(t)} \cdot m(s, t) \cdot \left(1 - p_c \frac{d_s}{l-1}\right) \cdot (1 - p_m)^{o(s)}$$

- $m(s, t)$ \equiv number of instances of schema s in population at time t
- $\bar{f}(t)$ \equiv average fitness of population at time t
- $\hat{u}(s, t)$ \equiv average fitness of instances of schema s at time t
- p_c \equiv probability of single point crossover operator
- p_m \equiv probability of mutation operator
- l \equiv length of individual bit strings
- $o(s)$ \equiv number of defined (non “*”) bits in s
- $d(s)$ \equiv 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: $Pr(\text{Acceptance}) \equiv p$
 - 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 (Ω): Range of a Random Variable X**
- **Probability Measure $Pr(\bullet)$**
 - Ω denotes a range of “events”; $X: \Omega$
 - **Probability Pr , or P , is a *measure* over Ω**
 - In a general sense, $Pr(X = x \in \Omega)$ is a measure of belief in $X = x$
 - $P(X = x) = 0$ or $P(X = x) = 1$: plain (aka categorical) beliefs (can’t be revised)
 - All other beliefs are subject to revision
- **Kolmogorov Axioms**
 - 1. $\forall x \in \Omega . 0 \leq P(X = x) \leq 1$
 - 2. $P(\Omega) \equiv \sum_{x \in \Omega} P(X = x) = 1$
 - 3. $\forall X_1, X_2, \dots \ni i \neq j \Rightarrow X_i \wedge X_j = \emptyset .$
$$P\left(\bigcup_{i=1}^{\infty} X_i\right) = \sum_{i=1}^{\infty} P(X_i)$$
- **Joint Probability: $P(X_1 \wedge X_2) \equiv$ Probability of the Joint Event $X_1 \wedge X_2$**
- **Independence: $P(X_1 \wedge X_2) = P(X_1) \cdot P(X_2)$**



Bayes's Theorem

- **Theorem**

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} = \frac{P(h \wedge D)}{P(D)}$$

- $P(h) \equiv$ **Prior Probability of Hypothesis h**
 - Measures **initial beliefs (BK) before** any information is obtained (hence **prior**)
- $P(D) \equiv$ **Prior Probability of Training Data D**
 - Measures probability of obtaining sample D (i.e., expresses D)
- $P(h | D) \equiv$ **Probability of h Given D**
 - $|$ denotes **conditioning** - hence $P(h | D)$ is a **conditional (aka posterior)** probability
- $P(D | h) \equiv$ **Probability of D Given h**
 - Measures probability of observing D given that h is correct (“**generative**” model)
- $P(h \wedge D) \equiv$ **Joint Probability of h and D**
 - Measures probability of observing D and of h being correct



Choosing Hypotheses

- **Bayes's Theorem**

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} = \frac{P(h \wedge D)}{P(D)}$$

- **MAP Hypothesis**

- Generally want most probable hypothesis given the training data
- Define: $\arg \max_{x \in \Omega} [f(x)] \equiv$ the value of x in the sample space Ω with the highest $f(x)$
- **Maximum a posteriori hypothesis, h_{MAP}**

$$\begin{aligned} h_{MAP} &= \arg \max_{h \in H} P(h|D) \\ &= \arg \max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg \max_{h \in H} P(D|h)P(h) \end{aligned}$$

- **ML Hypothesis**

- Assume that $p(h_i) = p(h_j)$ for all pairs i, j (**uniform priors**, i.e., $P_H \sim$ Uniform)
- Can further simplify and choose the **maximum likelihood hypothesis, h_{ML}**

$$h_{ML} = \arg \max_{h_i \in H} P(D|h_i)$$



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)

$P(\text{Cancer}) = 0.008$	$P(+ \text{Cancer}) = 0.98$	$P(+ \neg \text{Cancer}) = 0.03$
$P(\neg \text{Cancer}) = 0.992$	$P(- \text{Cancer}) = 0.02$	$P(- \neg \text{Cancer}) = 0.97$
 - $P(+ | H_0) P(H_0) = 0.0078$, $P(+ | H_A) P(H_A) = 0.0298 \Rightarrow h_{MAP} = H_A \equiv \neg \text{Cancer}$



Basic Formulas for Probabilities

- **Product Rule (Alternative Statement of Bayes's Theorem)**

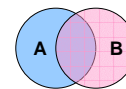
$$P(A | B) = \frac{P(A \wedge B)}{P(B)}$$

- Proof: requires axiomatic set theory, as does Bayes's Theorem

- **Sum Rule**

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

- Sketch of proof (immediate from axiomatic set theory)
 - Draw a Venn diagram of two sets denoting events *A* and *B*
 - Let $A \cup B$ denote the event corresponding to $A \vee B$...



- **Theorem of Total Probability**

- Suppose events A_1, A_2, \dots, A_n are mutually exclusive and exhaustive
 - Mutually exclusive: $i \neq j \Rightarrow A_i \wedge A_j = \emptyset$
 - Exhaustive: $\sum P(A_i) = 1$
- Then $P(B) = \sum_{i=1}^n P(B | A_i) \cdot P(A_i)$
- 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**
 - <http://www.cs.brown.edu/research/ai/dynamics/tutorial/Documents/HiddenMarkovModels.html>
 - Emphasis on a particular H (the space of hidden Markov models)



CIS 732: Machine Learning and Pattern Recognition

Kansas State University
Department of Computing and Information Sciences

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
 - $h_1 \equiv$ fair coin: $P(Head) = 0.5$
 - $h_2 \equiv$ 60% bias towards Head: $P(Head) = 0.6$
 - 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: $d \equiv$ a single coin toss, comes up Head
 - Q: What does the agent believe now?
 - A: Compute $P(d) = P(d | h_1) P(h_1) + P(d | h_2) P(h_2)$



CIS 732: Machine Learning and Pattern Recognition

Kansas State University
Department of Computing and Information Sciences

Bayesian Learning Example: Unbiased Coin [2]

- **Bayesian Inference: Compute $P(d) = P(d | h_1) P(h_1) + P(d | h_2) P(h_2)$**
 - $P(\text{Head}) = 0.5 \cdot 0.75 + 0.6 \cdot 0.25 = 0.375 + 0.15 = 0.525$
 - This is the probability of the observation $d = \text{Head}$
- **Bayesian Learning**
 - Now apply Bayes's Theorem
 - $P(h_1 | d) = P(d | h_1) P(h_1) / P(d) = 0.375 / 0.525 = 0.714$
 - $P(h_2 | d) = P(d | h_2) P(h_2) / P(d) = 0.15 / 0.525 = 0.286$
 - *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 $P(h_1 | d) \ll P(h_2 | d)$



Kansas State University
Department of Computing and Information Sciences

CIS 732: Machine Learning and Pattern Recognition

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:

$$P(h | D) = \frac{P(D | h)P(h)}{P(D)}$$

- 2. RETURN the hypothesis h_{MAP} with the highest conditional probability

$$h_{MAP} = \arg \max_{h \in H} P(h | D)$$



Kansas State University
Department of Computing and Information Sciences

CIS 732: Machine Learning and Pattern Recognition

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):** $P(\text{individual}) \propto f(\text{individual})$
 - **Tournament:** let individuals compete in pairs or tuples; eliminate unfit ones
 - **Crossover**
 - **Single-point:** $11101001000 \times 00001010101 \rightarrow \{ 11101010101, 00001001000 \}$
 - **Two-point:** $11101001000 \times 00001010101 \rightarrow \{ 11001011000, 00101000101 \}$
 - **Uniform:** $11101001000 \times 00001010101 \rightarrow \{ 10001000100, 01101011001 \}$
 - **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**

