

# COSC 5555

## Exam Review Sheet

Sunil Kothari

April 27, 2006

### 1. Introduction and Terminology

- (a) **Concept Learning** - Inferring a boolean valued function from training examples of its input and output. It can be viewed as the task of searching through a large hypotheses implicitly defined by the hypothesis representation.
- (b) Set of Instances - Set of items over which the concept is defined denoted by  $X$ .
- (c) **Target Concept** - The target or concept to be learned denoted by  $c$ .
- (d) Set of training example denoted by  $D$
- (e) Set of all possible hypotheses - Hypothesis Space denoted by  $H$
- (f) **Inductive Learning Hypothesis** - Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.
- (g) A hypothesis  $h$  is consistent with a domain theory  $B$  if  $B$  does not entail the negation of  $h$ .
- (h) The *weakest preimage* of a conclusion  $C$  with respect to a proof  $P$  is the most general set of initial assertions  $A$ , such that  $A$  entails  $C$  according to  $P$ .
- (i) The *Inductive Bias* of a learning algorithm is a set of assertions that, together with the training examples, *deductively* entail subsequent predictions made by the learner. In other words it characterizes how the learner generalizes beyond the observed training examples.

- (j) The set of all predictions entailed by a set of assertions Y is often called the *deductive closure* of Y.

## 2. Analytical learning vs. Inductive learning

- (a) The desired output in inductive learning is a hypothesis h from H that is consistent with the training examples whereas in analytical learning the desired output is a hypothesis h from H that is consistent with *both* the training examples D and the domain theory B.

## 3. Supervised concept learning

- (a) Givens
  - i. Labeled training examples
  - ii. Space of possible hypothesis
  - iii. Test Examples
  - iv. A set of attributes (features)
  - v. A set of possible values of each attribute
  - vi. A set of classes (eg + -) (target concept depends on the hypothesis space and classes)
- (b) Find
  - i. Target concept or an approximation of it

## 4. Induction versus Deduction

- (a) Induction - Requires leap of faith
- (b) Deduction - truth preserving, based on logic

## 5. Examples of supervised learning algorithms

- (a) Candidate Elimination Algorithm
  - i. Outputs a description set of all consistent hypothesis (with the training example) without enumerating all the members of the set
  - ii. This set, as mentioned above, is called version space
  - iii. S - boundary more general + ve examples
  - iv. G - boundary less specific - ve examples
  - v. Ends when  $S = G$
  - vi. Hypothesis space in conjunctive normal forms

- vii. Not robust to noisy training data
  - viii. Inductive bias- the target concept can be found in the provided hypothesis space
  - ix. All members of the current version space agree on the classification
  - x. Stronger bias than rote-learner
- (b) Find-S
- i. One at a time
  - ii. Consider positive and ignores negative examples
  - iii. For positive examples, if it is not covered by the existing hypothesis H then generalise H.
  - iv. There is no guarantee that it can find the target concept
  - v. Not robust to noisy training data
  - vi. More stronger bias than CEA (It assumes all instances are negative unless the opposite is entailed by its other knowledge)
  - vii. At each stage the hypothesis is the most consistent with the training examples seen so far
  - viii. No criteria to choose in case of multiple maximally specific consistent hypothesis.
- (c) Decision Trees
- i. Specific information theory algorithm but doesn't look at the (greedy algorithm) whole tree
  - ii. Only for discrete valued functions
  - iii. Top-down greedy search through the space of possible trees with no backtracking
  - iv. Robust to errors, both error in classification of training examples and errors in the attribute values
  - v. Preference Bias vs. Restriction Bias
    - A. Preference bias is preferred as it allows the learner to work within a complete hypothesis space that is assured to contain the target concept
    - B. Restriction bias (like the one used in CEA) introduces the possibility of excluding the unknown target function altogether
  - vi. ID3 Algorithm:

- A. **Inductive bias in ID3:** preference for shorter trees. Trees that place high information gain attributes close to the root are preferred over those that do not
- B. Uses information gain to select the attribute at each step of growing the tree
- C.  $Gain(S, A) \stackrel{\text{def}}{=} Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$  where  
  - Values(A) - Set of all possible values for attribute A
  - S - Entropy of the original collection
  - $S_v$  - Expected Entropy after the S is partitioned using attribute A
- D. ID3 can be described as hill-climbing search through a set of possible decision trees
- E. ID3 maintains only a single current hypothesis as it searches through the space of decision trees
- F. By not maintaining a set of hypotheses it cannot determine other alternate decision trees (consistent with the training data) or how to resolve between multiple hypotheses.
- G. Performs no backtracking on its search; follows a hill-climbing approach and thus risks converging to local optimal solutions
- H. Less sensitive (compared to FIND-S and CEA) to errors in training examples
  - I. ID3 searches a *complete* hypothesis space but searches *incompletely* through this hypothesis space.
- vii. Overfitting of training data is an important issue
  - A. Solution: 1. Post-pruning of decision tree ( allow overfitting to occur)
  - B. Solution 2. Stop before overfitting occurs
- viii. Criteria to be used to determine correct final tree size
  - A. Training and validation set approach
  - B. Use an explicit measure of the complexity for encoding the training example and the decision tree
  - C. Use a statistical test to estimate improvements accrued by expanding a particular node

- (d) Neural Networks
  - i. learn using backpropagation robust to noisy data
  - ii. greedy
  - iii. search in through error space
  - iv. Perceptrons guaranteed to find global minima ( there is only one)
  - v. Perceptron
    - A. Linear activation function
    - B. No hidden nodes and no hidden layer
    - C. output layer has 1 node
    - D. Preceptron representation theorem  
They can only classify linearly separable sets of examples
    - E. The hypotheses space consists of a set of all possible real-valued weight vectors
  - vi. The hypothesis space considered by a neural network (back-propagation algorithm) is the space of all functions that can be reprsented by assigning weights to the given, fixed network of interconnected units
  - vii. Overfitting of training data → Generalize poorly to new data despite excellent performace over trg. data  
Possible solution - cross validation
  - viii. Sigmoid function is used as it is a nonlinear function of its output and it is differentiable
- (e) Backpropagation algorithm
  - i. applies gradient descent algorithm

## 6. Unsupervised Learning

## 7. Explanation Based Learning ( An instance of Instance Based Learning)

- (a) Advantages: Theoretical bounds in case of purely inductive learning algorithms due to insufficient data. No such bounds
- (b) Prior knowledge (also known as domain theory) is used to analyze, or explain, how eah observed training example satisfies the target concept. The domain theory is descibed by a set of Horn clauses.
- (c) Hypothesis space consists of a set of first order Horn clauses

- (d) Domain theories used in EBL are perfect - i.e. correct and complete. A domain theory is *correct* if each of its assertions is a truthful statement about the world. A domain theory is *complete* w.r.t. a given target concept and instance space if the domain theory covers every possible example in the instance space.
- (e) Used to segregate relevant features from non-relevant features
- (f) Used prior instances to reduce the complexity of the hypotheses space to be searched
- (g) PROLOG-EBG algorithm
  - i. For each positive example not covered by the set of learned Horn clauses a new Horn rule is created.
  - ii. The new horn rule is created by
    - A. explaining the training example in terms of domain theory
    - B. analyzing the explanation to determine the relevant features (create a most general rule or the weakest preimage of the explanation) of the example
    - C. constructing a new Horn clause that concludes the target concept
- (h) The learner must output an hypothesis that is consistent with training examples and domain theory
- (i) A hypothesis H is consistent with a domain theory B provided B does not entail the negation of h.
- (j) PROLOG-EBG algorithm can formulate new features that are not explicit in the description of training examples.
- (k) Inductive bias of Prolog EBG: Among the sets of Horn clauses entailed by domain theory, the bias is for small sets of maximally general Horn clauses.

## 8. Instance Based Learning

- (a) Simply stores the training examples
- (b) Generalizing beyond the training examples is postponed until new examples are seen (lazy learner, backprop etc. are eager learners)
- (c) Classifies new instances by looking at similar instances and ignore dissimilar instances.

- (d) Represents instances as real valued points in an n-dimensional Euclidean space.
- (e) Inductive bias for lazy vs. eager learning methods
  - i. Lazy methods (global approximations to the target function) must consider the query instance  $x_q$  when deciding how to generalize beyond the training data
  - ii. Eager methods (local approximations to the target concept) cannot.
- (f) Advantages
  - i. Can estimate the target function locally (other learning methods approximate globally) and differently for each new instance to be classified. This is useful if the target concept is very complex.
  - ii. Less computation during training
  - iii. Information present in the training examples is never lost
- (g) Disadvantages
  - i. Cost of classifying new instances are high (all computation takes place at classification time)
  - ii. Similar instances are described by considering ALL attributes and therefore even similar instances can be quite dissimilar
- (h) K-nearest Neighbor
  - i. Assumes all instances as points in n- dimensional euclidean space
  - ii. Robust to noisy data
  - iii. The inductive bias corresponds to an assumption that the classification of an instance  $x_q$  will be most similar to the classification of other instances that are nearby in euclidean space.
  - iv. Suffers from *curse of dimensionality* problem and the solution is stretching the axes to suppress the effect of irrelevant attributes.
  - v. Computational cost of classifying a new instance can be lowered by indexing the stored training data but this has memory overheads
- (i) Locally Weighted Regression
  - i.

(j) Case Based Reasoning

- i. Instances are represented by complex logical descriptions. This may require a similarity matrix different from Euclidean distance
- ii. Multiple retrieved cases can be combined to form the solution to the new problem. This combination process is purely statistical in K-NEAREST NEIGHBOR but in Case Based Reasoning it could be knowledge based reasoning.

9. Bayesian Learning

- (a) New instances can be classified by combining the predictions of multiple hypotheses weighted by their probabilities.
- (b) Requires initial knowledge of many probabilities
- (c) Naive bayesian classifier
  - i. Naive because it assumes attribute values are conditionally independent, given the classification of the instance.

- Computational learning

10. Unsupervised Concept Learning

- (a) Instances are not labelled

11. Reinforcement Learning

- (a) Main idea is to choose sequences of actions that result in maximum cumulative rewards for an agent with future rewards discounted exponentially by their delay
- (b) Exploration (of unknown states and actions) versus Exploitation (of known states and actions that will yield high reward) tradeoff
- (c) Markov decision process
  - i. The functions  $\delta(s,t,a,t)$  and  $r(s,t,a,t)$  depend only on the current state
- (d) Here the techniques may have non-deterministic outcomes and the learner lacks a domain theory (as in case of explanation based learning)
- (e) The task of the agent is to learn a policy  $\pi : S \rightarrow A$  for selecting its next action  $a_t$  based on the current observed state

(f) Q- Learning

- i.  $Q(s, a) = r(s, a) + \gamma \max_{a'} Q(s, a')$
- ii. We choose Boltzmann distribution for choosing actions so that all actions have a non-zero probability and this assures convergence
- iii. Q-Learning can be applied even when the learner has no prior knowledge of how its actions affect its environment

12. Evolutionary Learning