Introduction to Machine Learning

CSE474/574: Lecture 5

Varun Chandola <chandola@buffalo.edu>

4 Feb 2015

Outline

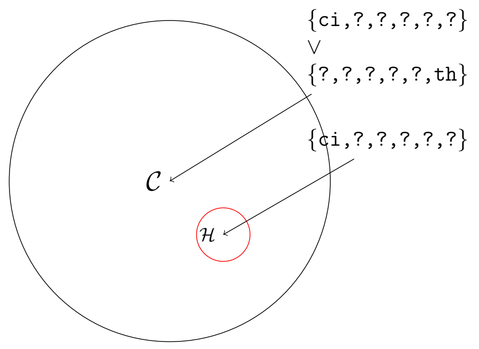
Contents

1 Inductive Bias

- Target concept labels examples in X
- $2^{|X|}$ possibilities ($\mathcal{C}$)
- $|X| = \prod_{i=1}^{d} n_i$
- Conjunctive hypothesis space $\mathcal{H}$ has $\prod_{i=1}^{d} n_i + 1$ possibilities
- Why is this difference?

For our running example with 5 binary attributes, there can be 32 possible instances and 2^{32} possible concepts ($\approx$ 4 Billion possibilities)! On the other hand, the number of possible conjunctive hypotheses are only 244. The reason for this discrepancy is the assumption that was made regarding the hypothesis space.

Hypothesis Assumption
Target concept is *conjunctive*.



The conjunctive hypothesis space is *biased*, as it only consists of a specific type of hypotheses and does not include others.

Obviously, the choice of conjunctive concepts is too narrow. One way to address this issue is to make the hypothesis space *more expressive*. In fact, we can use the entire space of possible concepts ($\mathcal{C}$) as the hypothesis space in the candidate elimination algorithm. Since the target concept is guaranteed to be in this $\mathcal{C}$, the algorithm will eventually find the target concept, given enough training examples. But, as we will see next, the training examples that it will require is actually X, the entire set of possible examples!

- Simple tumor example: 2 attributes - size (sm/lg) and shape (ov/ci)
- Target label - malignant (+ve) or benign (-ve)
- $|X| = 4$
- $|\mathcal{C}| = 16$

Here are the possible instances in X:

x_1: sm,ov

x_2: sm,ci

x_3: lg,ov

x_4: lg,ci

The possible concepts in the entire concept space $\mathcal{C}$:

- $\{\}$ – *most specific*
- $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}$
- $\{x_1 \vee x_2\}, \{x_1 \vee x_3\}, \{x_1 \vee x_4\}, \{x_2 \vee x_3\}, \{x_2 \vee x_4\}, \{x_3 \vee x_4\}$
- $\{x_1 \vee x_2 \vee x_3\}, \{x_1 \vee x_2 \vee x_4\}, \{x_1 \vee x_3 \vee x_4\}, \{x_2 \vee x_3 \vee x_4\}$
- $\{x_1 \vee x_2 \vee x_3 \vee x_4\}$ – *most general*

Start with $G = \{x_1 \vee x_2 \vee x_3 \vee x_4\}$ and $S = \{\}$. Let $\langle x_1, +ve\rangle$ be the first observed example. S is modified to $\{x_1\}$. Let $\langle x_2, -ve\rangle$ be the next example. G is set to $x_1 \vee x_3 \vee x_4$. Let $\langle x_3, +ve\rangle$ be the next example. S becomes $\{x_1 \vee x_3\}$.

In fact, as more examples are observed, S essentially becomes a disjunction of all positive examples and G becomes a disjunction of everything except the negative examples. When all examples in X are observed, both S and G converge to the target concept.

Obviously, expecting all possible examples is not reasonable. Can one use the intermediate or partially learnt version space? The answer is no. When predicting an unseen example, exactly half hypotheses in the partial version space will be consistent.

- **A learner making no assumption about target concept cannot classify any unseen instance**

While the above statement sounds very strong, it is easy to understand why it is true. Without any assumptions, it is not possible to "generalize" any knowledge inferred from the training examples to the unseen example.

Inductive Bias
Set of assumptions made by a learner to generalize from training examples.

For example, the inductive bias of the *candidate elimination* algorithm is that the target concept c belongs to the conjunctive hypothesis space $\mathcal{H}$. Typically the inductive bias restricts the search space and can have impact on the efficiency of the learning process as well as the chances of reaching the target hypothesis.

Another approach to understand the role of the inductive bias is to first understand the difference between *deductive* and *inductive* reasoning. Deductive reasoning allows one to attribute the characteristics of a general class to a specific member of that class. Inductive reasoning allows one to generalize a characteristic of a member of a class to the entire class itself. An example of deductive reasoning:

- All birds can fly. A macaw is a type of bird. Hence, a macaw can fly.

Note that deductive reasoning assumes that the first two statements are true. An example of inductive reasoning is:

- Alligators can swim. Alligator is a type of reptile. Hence, all reptiles can swim.

The inductive reasoning, even when the first two statements are true, is untrue. Machine learning algorithms still attempt to perform inductive reasoning from a set of training examples. The inference on unseen examples of such learners will not be provably correct; since one cannot *deduce* the inference. The role of the inductive bias is to add assumptions such that the inference can follow deductively.

- *Rote Learner* – No Bias
- *Candidate Elimination* – Stronger Bias
- *Find-S* – Strongest Bias

2 Online Learning

- $X = \{true, false\}^d$
- $D = X^{(1)}, X^{(2)}, \ldots$
- $D \subseteq X$
- $c \in \mathcal{C}, c : X \rightarrow \{0, 1\}$

```
1: for $i = 1, 2, \ldots$ do
2:     Learner given $x^{(i)} \in X$
3:     Learner predicts $c_*(x^{(i)})$
4:     Learner is told $c(x^{(i)})$
5: end for
```

Learning Objective
"Discover" c with minimum number of prediction mistakes

While the objective of the learner is to learn the target concept while making as few mistakes as possible, we are also interested in estimating the bounds on the number of mistakes for learning the concept.

This can also be treated as the quality of the learning algorithm. The algorithm's learning behavior is measured by counting the number of mistakes it makes while learning a function from a specified class of functions. Obviously, the computational complexity (space and time) is also considered. The idea is to put a lower and upper bound on the number of mistakes.

2.1 Online Learning of Conjunctive Concepts

- Target concept c is conjunctive
- Examples are denoted using binary variables v_i
  - Example: $v_1\bar{v}_2v_3v_4$
  - v_i means attribute i is *true* (or 1 or *circular*) and $\bar{v}_i$ means attribute i is *false* (or 0 or *oval*)
- Initialize $L \rightarrow \{v_1, \bar{v}_1, v_2, \bar{v}_2, \ldots, v_d, \bar{v}_d\}$
- *Match* input x and L to get prediction, $c_L(x)$
- *If* $c_L(x) \neq c(x)$ (a **mistake**)
  - Remove *offending* entries from L
- Consider next input
- L is the learnt concept when finished

For example, let us assume that $d = 4$. So we initialize $L = \{v_1, \bar{v}_1, v_2, \bar{v}_2, v_3, \bar{v}_3, v_4, \bar{v}_4$. This concept translates to – assign 1 to any input which takes value *true* and value *false* for every attribute. Obviously, this concept will assign a value 0 to every input. Let the first input be $\langle\{false, true, true, true\}, 1\rangle$. Since L makes a mistake on this input (regardless of its label), the entries in L that do not match with the input will be eliminated. Thus, L will become $\{\bar{v}_1, v_2, v_3, v_4\}$. Let next input be $\langle\{true, true, true, false\}, 1\rangle$. The match between the input and L will yield a 0 (on attribute 1 and 4), and hence a mistake will be made. This will result in elimination of the first and last literal of L to yield $\{v_2, v_3\}$.

2.1.1 Properties

- Always makes mistake on the first example
- First mistake causes d entries to be removed from L
- Number of literals to be removed to reach target concept of length p is $2d - p$
- No entry in c is removed from L
- Mistakes are made only on positive examples
- Each mistake causes ≥ 1 entry to be removed from L

Mistake Bound
Concept c can be learnt with at most $d + 1$ prediction mistakes

This result puts an upper bound on the number of mistakes that any algorithm should make to learn conjunctive concepts. Next we study another specific concept class - disjunction of a fixed number of attributes.

3 Optimal Mistake Bounds for a Concept Class

- $\mathcal{L}$ - Learning algorithm
- c - Target concept ($c \in \mathcal{C}$)
- D - One possible sequence of training examples
- $M_{\mathcal{L}}(c, D)$ - Number of mistakes made by $\mathcal{L}$ to learn c with D examples
- $M_{\mathcal{L}}(c) = \max_{\forall D} M_{\mathcal{L}}(c, D)$
- Worst case scenario for $\mathcal{L}$ in learning c
- $M_{\mathcal{L}}(\mathcal{C}) = \max_{\forall c \in \mathcal{C}} M_{\mathcal{L}}(c)$
- Worst case scenario for $\mathcal{L}$ in learning any concept in $\mathcal{C}$

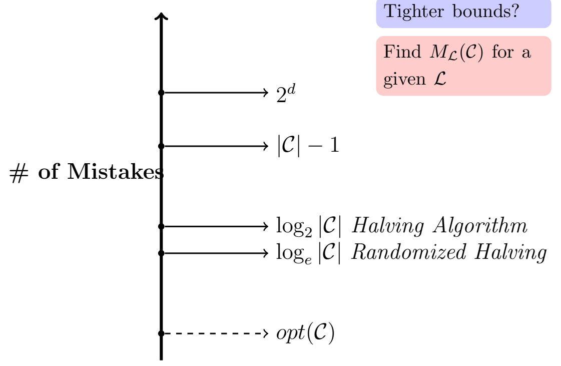
Optimal Mistake Bound
$opt(\mathcal{C}) = \min_{\forall \mathcal{L}} M_{\mathcal{L}}(\mathcal{C})$

Note that the optimal mistake bound is independent of the training examples and the algorithm and even the actual concept to be learnt. It essentially puts a upper bound on any algorithm that is designed to learn a concept belonging to $\mathcal{C}$. So if one designs a new algorithm, it should not, for any sequence of training examples, have more than $opt(\mathcal{C})$ mistakes when learning any concept $c \in \mathcal{C}$. In other words, to find an optimal learning algorithm for a concept class, one should guarantee that it can learn *any* concept in the concept class, using *any* example sequence, with at most $O(opt(\mathcal{C}))$ mistakes.

Another important benefit of mistake bounds is that the process of estimating the bounds often indicates strategies to design better algorithms.

3.1 Bounds on the Optimal Mistake Bound

It is clear that exact estimation of $opt(\mathcal{C})$ is not possible as it requires knowledge of all possible learners. But one can definitely impose bounds on this bound. Obvious bounds are 2^d, the maximum possible number of examples and $\mathcal{C} - 1$ which is obtained by a learner which "tries" every possible concept and eliminates them one by one.


Tighter bounds?
Find $M_{\mathcal{L}}(\mathcal{C})$ for a given $\mathcal{L}$
2^d
$|\mathcal{C}| - 1$
# of Mistakes
$\log_2 |\mathcal{C}|$ *Halving Algorithm*
$\log_e |\mathcal{C}|$ *Randomized Halving*
$opt(\mathcal{C})$

For instance, for the space of conjunctive concepts, we have already seen that there exists an algorithm that can learn any concept with $d+1$ mistakes. In general, one can prove that $opt(\mathcal{C}) \leq \log(|\mathcal{C}|)$ using the *Halving Algorithm*.