CSCI 632 Notes

Clay L. McLeod

February 1, 2016

1 Machine Learning Overview

1.1 Supervised Learning

An **observation** is a d-dimensional vector X such that $X \in \mathbb{R}^d$.

The unknown nature of observation is called a **class**. We denote it by Y where $y \in \{1, 2, ..., M\}$. For the purpose of this course, only discrete classes are considered (no regression).

The goal is to create a function $g(x) : \mathbb{R}^d \to \{1, ..., M\}$ $g(x)$ one's guess of y given x. The classifier is $g(x)$. If $g(x) \neq y$.

Questions

1. How does one construct a good classifier?
2. How good can a classifier be?
3. Is classifier A better than classifier B?
4. Can we estimate how good a classifier can be?
5. What is the best classifier?

The answer to all of these questions is yes: there are ways to find an upper bound on the performance of each algorithm and evaluate it empircally.

1.2 Unsupervised Learning

Same definition for an observation, except we don't have labels for the class in X. What approaches might this help us tackle?

1.2.1 Clustering

Unsupervised learning is directly related supervised learning. For instance: feature selection is probably the most important part of designing Machine Learning algorithms. Unsupervised learning helps us find good features for supervised learning algorithms.

1.2.2 Dimensionality reduction

As you increase the number of dimensions, you loss the ability to distinguish between two examples. Also, run time increases exponentially.

1.3 Semisupervised Learning

Partially labelled data where we try to gain some intuition. Usually involves a cost function instead of a solution set.

1.4 References

1. *A Probability Theory of Pattern Recognition* for Theoretical Design
2. *Machine Learning* for History of ML
3. *The Elements of Statistical Learning* for Statistical Vantagepoint
4. *Pattern Recognition and Machine Learning* (Textbook)
5. *Kernel Methods for Pattern Analysis* for Kernel Methods

2 Probability Review

In order to correctly analyze machine learning models and their correctness, we should first address some basic concepts in probability.

Definition: A probability space has 3 components.

1. A sample space, Ω, which is a set of all of the possible outcomes of a random process.
2. A family of sets, $\Im$ representing the allowable events, where each set in $\Im$ is a subset of Ω. $\Im$ is a powerset of Ω.
3. A probability function $P_r : \Im \rightarrow R$ satisfying
   (a) $\forall E \in \Im, 0 \leq P_r(E) \leq 1$
   (b) $P_r(\Omega) = 1$

(c) $P_r(\bigcup_{i\geq 1} E_i) = \sum_{i\geq 1} P_r(E_i)$ if the RVs are independent.

Example: toss two dice

- $\Omega = \{(1,1),(1,2),\cdots,(6,6)\}$
- $\Im = \{\cdots\} = |\Im| = 2^{36}$
- $P \to R$
  - $P((a,b)) = \frac{1}{36}, 1 \leq a, b \leq 6$
  - $P(E) = \sum_{(x,y)\in E} P((x,y)) = |E| \cdot \frac{1}{36}$

2.1 Lemma (Union bound)

Given: $\forall E_1, E_2 \subset \Omega$
Derived: $P(E_1 \bigcup E_2) = P(E_1) + P(E_2) - P(E_1 \bigcap E_2) \Rightarrow P(E_1 \bigcup E_2) \leq P(E_1) + P(E_2)$

2.2 Lemma (Independence)

Given: $\forall$ finite or countably infinite sequence of events $E_1, E_2, \cdots$
Derived:$P_r(\bigcup_{i\geq 1} E_i) = \sum_{i\geq 1} P_r(E_i)$

2.3 Lemma (Inclusion-Exclusion principle)

Given: Let $E_1, \cdots, E_n$ be any of n events.
Derived: $P(\bigcup_{i=1}^{n} E_i) = \sum_{i=1}^{n} P(E_i) - \sum_{i<j} P(E_i \bigcap E_j) + \sum_{i<j<k} P(E_i \bigcap E_j \bigcap E_k) \cdots$

<u>Definition</u>

Two events E and F are independent if and only if

$$P(E \bigcap F) = P(E) \cdot P(F)$$

or, more generally the probability that *all* the events will happen is the same as the probability that *each* event will happened multiplied together.

Note: Independence $\neq$ uncorrelated.

Definition

The conditional probability that the event E occurs given that event F occurs is

$$P(E|F) = \frac{P(E \bigcap F)}{P(F)}$$

or, written another way,

$$P(E \bigcap F) = P(E|F) \cdot P(F)$$

However,

$$P(E|F) = P(E)$$

when E and F are independent.

2.4 Theorem (Law of total probability)

Let $E_1, \cdots, E_n$ be mutually disjoint elements in Ω.

$$P(A) = \sum_n P(A|E_n) \cdot P(E_n)$$

2.5 Theorem (Bayes' Law)

Assume that $E_1, \cdots, E_n$ are mutually disjoint sets such that

$$\bigcup_{i=1}^{n} E_n = E$$

Then

$$P(E_j|B) = \frac{P(B|E_j) \cdot P(E_j)}{\sum\limits_{i=1}^{n} P(B|E_i) \cdot P(E_i)}$$

This is proven by the combination of the law of conditional probability on the top and the law of total probability on the bottom.

Example

Two fair coins, biased coin($P(H) = \frac{2}{3}$). Assume that the output is `HHT`. What is the probability that the first coin was the biased coin?

- $B = \texttt{HHT}$
- E_i = ith coin toss is biased, $P(E_i) = \frac{1}{3}$.
- $P(E_1|B) = \frac{P(B|E_1) \cdot P(E_1)}{P(B)}$

2.6 Random Variables

2.6.1 Bernoulli

Example Toss a fair coin where p is the probability that the outcome is heads. Written as

$$X \sim \text{ Bernouilli}(p)$$

2.6.2 Binomial

Example Number of heads in n coin tosses. Written as

$$X \sim \text{ Binomial}(n, p)$$

3 Supervised Learning

Given observations

$$X_i \in \mathbb{R}^d; i = 1, \cdots, n$$

and their classes y_i (discrete) such that

$$y_i \in 1, \cdots, M$$

Find

$$g : \mathbb{R}^d \to \{1, \cdots, M\}$$

that can predict the class of X. That supervised function is defined as

$$\Im = \{\text{set of funcs } \mathbb{R}^d \to \{1, \cdots, M\}\}$$

.

3.1 Performance of a classifier

How do we determine the effectiveness of g as a classifier? At first, one might assume that this means the "probability of an error". This is also known as the **generalized error**. Before we can discuss the different error measures, we must first define a few baseline facts:

1. Assume $(X, y) \sim P(X, y)$.
2. Assume $y \in \{w_1, \cdots, w_M\}$.
3. We assume all observations and class pairs $(X,\ y)$ are generated by a join probability distribution $P(X, y)$. In other words, we assume that this data is *learnable*. Clearly by the law of conditional probability,

$$P(X, y) = P(X|y) \cdot P(y)$$

4. Although it is unrealistic in practice, assume that we know the probability distribution that generates X and y. So knowing $P(X, y)$ is equivalent to knowing

$$P(w_i) : i = 1, \cdots, M \qquad (\textbf{prior})$$

and

$$P(x|w_i) : i = 1, \cdots, M \qquad (\textbf{conditional density})$$

Given that

$$P(x|w_i), P(w_i) : i = 1, \cdots, M$$

design a classifier $g(x)$ with minimal $P(\text{error})$. We start with binary classification $y \in \{w_1, w_2\}$.

$$P(\text{error}) = \int_{\mathbb{R}^d} P(\text{error}, x)dx = \int_{\mathbb{R}^d} P(\text{error}|x) \cdot P(x)dx$$

Note that the conditional error $P(\text{error}|x)$ depends on the choice of $g(x)$. X's class is either w_1 or w_2 with probabilities

$$P(w_1|x)$$

and

$$P(w_2|x)$$

These two probabilities obviously sum to 1 for binary applications.

Example What is our probability if $g(x)$ predicts x as w_2?

Answer $P(error|x) = 1 - P(w_2|x) = P(w_1|x)$

More generally speaking, $P(error|x) = P(w_1|x)$ if $g(x) = w_2$ and vice-versa for $g(x) = w_1$. From this, we can derive that

$$P(error|x) \geq \ \min\{P(w_1|x), P(w_2|x)\}$$

Recall that in our calculations, we left off with

$$\int_{\mathbb{R}^d} P(\text{error}|x) \cdot P(x)dx$$

Thus, we can say

$$\int_{\mathbb{R}^d} P(\text{error}|x) \cdot P(x)dx \geq \int_{\mathbb{R}^d} \ \min\{P(w_1|x), P(w_2|x)\} \cdot P(x)dx$$

.

3.2 Bayes' Error

The definition of Bayes' error is

$$P^*(\text{error}) = \int_{\mathbb{R}^d} \ \min\{P(w_1|x), P(w_2|x)\} \cdot P(x)dx$$

There are classifiers that will minimize Bayes' error. Assume that our decision rule is

$$g(x) = \begin{cases} w_1 & \text{if } P(w_1|x) > P(w_2|x) \\ w_2, & \text{otherwise} \end{cases}$$

.

3.3 Formalized Theory

Given **the priors** $P(w_1), P(w_2)$ and the **conditional densities** $P(x|w_1), P(x|w_2)$, we want the **posterior** $P(w_1|x), P(w_2|x)$. From the law of total probability, we have

$$P(w_1|x) = \frac{P(x,w)}{P(x)} = \frac{P(X|W_1) \cdot P(w_1)}{P(x)} = \frac{P(X|w_1) \cdot P(w_1)}{\sum_{i=1}^{2} P(X|w_i) \cdot P(w_i)}$$

In the machine learning literature, $P(x)$ is called the **evidence**. From this, we can compute the **likelyhood ratio**

$$\frac{P(X|w_1) \cdot P(w_1)}{P(X|w_2) \cdot P(w_2)}$$

In practice, we check to see if this is > 1 for our classification. However, we use **log likelyhood ratio**. This is expressed as

$$ln \cdot P(X|w_1) + ln \cdot P(w_1) - ln \cdot P(X|w_2) - ln \cdot P(w_2)$$

These two expressions are mathematically equivalent, but log likelyhood allows us to avoid an *underflow* problem when computed.

3.4 Looking ahead

We will consider three types of problems:

1. More than two classes
2. More than two decisions
3. More general cost function