CSCI 632 Notes

Clay L. McLeod

February 1, 2016

1 Machine Learning Overview

This class will mainly be focused on the theoretical underpinnings of machine learning. We will be by answering many questions about the validity of machine learning models, such as:

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 empirically. The inner workings of these algorithms will be explained later in the course.

Before discussing the topic of machine learning in general, it will be useful to give a brief overview of the facets of machine learning as it relates to this class. There are three main areas of machine learning: (1) supervised learning (2) unsupervised learning (3) semi-supervised learning.

1.1 Supervised Learning

Supervised learning deals with mapping known inputs to known outputs. Below are some definitions imperative to understanding supervised machine learning.

- An **observation** is a d-dimensional vector X such that $X \in \mathbb{R}^d$. In supervised learning, X is a known variable.
- A **class** is the unknown nature of an observation. We denote it by Y where $Y_i \in \{1, 2, ..., M\}$ maps to each input vector X_i. In supervised learning, Y is a known variable.

  - *Note:* For the purpose of this course, only discrete classes are considered (no regression).

- A **classifier** is a function $g(X) : \mathbb{R}^d \to \{1, ..., M\}$. $g(X)$ is the classifier's estimation of Y given X The classifier is $g(X)$. If $g(X) \neq Y$.

1.2 Unsupervised Learning

Unsupervised learning is similar in that X and $g(X)$ all have the same definitions. However, there is one difference for Y: the class labels are not known. This type of learning helps us find structure in the data rather than learn the structure in the data.

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.

Some approaches that we will cover to unsupervised machine learning are:

1.2.1 Clustering

Imposing the input data X into n different clusters. This is useful when trying to find structure within the data.

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. Thus, it is useful to try to condense the information in a dataset to only that which is pertinent.

1.3 Semisupervised Learning

Partially labelled data where we try to gain some intuition. Usually involves a cost function instead of a solution set. This type of algorithm learns as it goes along and makes adjustments based on its performance.

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

2.1 Basic definitions in probability

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 $P_r : \Im \to R$ satisfying
   (a) $\forall E \in \Im, 0 \leq P_r(E) \leq 1$
   (b) $P_r(\Omega) = 1$
   (c) $P_r(\bigcup\limits_{i \geq 1} E_i) = \sum\limits_{i \geq 1} P_r(E_i)$ if the RVs are independent.

Example: toss two dice

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

2.2 Lemmas and Theorems

2.2.1 Independence

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. Independence essentially means that the outcomes each either event does not affect the other.

Note: Independence $\neq$ uncorrelated.

Lemma 2.1 (Independence) *For all finite or countably infinite sequence of independent events $E_1, E_2, \cdots, E_n$, we can compute the union of the probability of each event by the formula*

$$P_r(\bigcup_{i\geq 1} E_i) = \sum_{i\geq 1} P_r(E_i)$$

Succinctly, the probability of all of the events happening is the combination of the probability of each event happening.

Lemma 2.2 (Union bound) *For all events $E_1, E_2 \subset \Omega$, we know that*

$$P(E_1 \bigcup E_2) = P(E_1) + P(E_2) - P(E_1 \bigcap E_2)$$

or

$$P(E_1 \bigcup E_2) \leq P(E_1) + P(E_2)$$

Succinctly, we know that the probability that both E_1 and E_2 occur together is less than or equal to the probability that E_1 and E_2 occur independently of one another.

Lemma 2.3 (Inclusion-Exclusion Principle) *Let $E_1, \cdots, E_n$ be any of n events, we derive that*

$$P(\bigcup_{i=1}^{n} E_i) = \sum_{i=1}^{n} P(E_i) - \sum_{i<j} P(E_i \bigcap E_j) + \sum_{i<j<k} P(E_i \bigcap E_j \bigcap E_k) \cdots$$

Take for instance the case of three events, A, B, and C. The equation for computing the union of these three events is

$$P(A \bigcup B \bigcup C) = P(A)+P(B)+P(C)-P(A \bigcap B)-P(A \bigcap C)-P(B \bigcap C)+P(A \bigcap B \bigcap C)$$

This lemma is nothing more than an understanding of how to combine an arbitrary number of states in a union.

Theorem 2.4 (Conditional Probability) *The probability that the event E occurs given as a base truth that event F occurs is*

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

or, written another way,

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

Succinctly, the probability that E occurs given that F occurred is the same as restricting your probabilistic universe to all outcomes where F occurred and taking the ratio of all outcomes where E occurred over that sample space.

This axiom is a foundational lemma of machine learning. Note that $P(E|F) = P(E)$ when E and F are independent.

Theorem 2.5 (Law of total probability) *Let $E_1, \cdots, E_n$ be mutually disjoint elements in Ω. We can define the total probability that event A occurred as*

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

Succinctly, we can slice our sample space into n segments and combine together each probability when we know the event E_i happened. In this way, we cover our whole sample space and, thus, get the whole probability of A.

Theorem 2.6 (Bayes' Rule) *Assume that $E_1, \cdots, E_n$ are mutually disjoint sets such that $\bigcup_{i=1}^{n} E_n = E$. Then*

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

Succinctly, we restrict our probability universe to all outcomes where B happened (like in the Law of total probability). Next, we evaluate the ratio of when both E_j and B happen to our previously defined universe.

Note: the bottom is merely the law of total probability.

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?

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

2.3 Random Variables

The different types of RVs are left to previous courses. However, we will very briefly review two main RVs that are useful in this course. For the expectation, standard deviation, or any other characteristics of these RVs, google them.

2.3.1 Bernoulli Distribution

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

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

2.3.2 Binomial Distribution

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