CS 480 Course Notes

Introduction to Machine Learning

Michael Socha

University of Waterloo
Winter 2019

Contents

1 Course Overview

This is an applied introductory machine learning course covering the basics of machine learning algorithms and data analysis. Topics covered include:

- Regression analysis
- Probabilistic modeling
- Support vector machines
- Supervised vs unsupervised learning
- Reinforcement learning
- Neural networks

2 Introduction - What is Machine Learning?

Machine learning is the field of study of how computers can improve their performance at tasks (i.e. learn) without being explicitly programmed to do so. Machine learning can be useful for tasks for which it is difficult to write a step-by-step imperative program. Sample applications include optical character recognition, computer vision, and game playing.

2.1 Learning Frameworks

- **Supervised Learning:** Goal is to learn a function based on its input and output (e.g. determining if an email is spam based on a set of emails labeled as spam or not spam).
- **Unsupervised Learning:** Goal is to learn a function based on its input alone (e.g. organizing data into clusters).
- **Reinforcement Learning:** Goal is to learn a sequence of actions that maximize some notion of reward (e.g. learning how to control a vehicle to perform some maneuver).

2.2 Challenges

Some of the challenges facing machine learning today are:

- Dealing with large amounts of data (algorithm complexity and distributed computing become very relevant)
- Generating reproducible results
- Challenges concerning real-world adoption of work (e.g. human computer interaction, robustness, ethical concerns)

3 Decision Trees

Decision trees contain questions as nodes and answers as edges, and serve to guide to an answer based on a set of observations.

3.1 Predictions Using Decision Trees

Consider a data set of employees and their job satisfaction, where we use the data set to predict whether an employee is satisfied with their job. An employee can have many features, such as age, salary, seniority, working hours. It is infeasible to build all possible decision trees when there are many features due to the exponential growth of the tree. Instead, for predictive purposes, it makes sense to focus on those questions that are informative to the prediction. For example, if there is no major difference in job satisfaction based on an employee's age, then it is unnecessary to include age in a decision tree, since the answer is not informative for the prediction.

In general, supervised learning problems have a set of possible inputs X, an unknown target function $f : X \to Y$, and a set of function hypotheses $\{h|h : X \to Y\}$. Supervised learning algorithms accept training examples (a pair (x, y) where $x \in X$, $y \in Y$) and output a function hypotheses that approximates f. When using decision trees for learning, each decision tree is a function hypothesis.

3.2 Encoding Functions in Decision Trees

Boolean functions can be fully expressed in decision trees, with one branch for true and another for false. Other function can be approximated as a boolean function. For example, instead of a node asking what an employee's salary is, it could ask whether the salary is above a certain amount.

3.3 Training and Testing

The key idea behind decision tree generation algorithms is to grow a tree until it correctly classifies all training examples. The rough procedure followed is:

1. If all training data has the same class, create a leaf node and return.
2. Else, create the best (i.e. most informative) node on which the split the data.
3. Split the training set over the above node.
4. Continue the procedure on each subset of training data generated.

3.3.1 Information Content

Criteria for finding the "best" split of data can be defined mathematically. If event E occurs with probability $P(E)$, then when E occurs, we receive $I(E) = \log_2 \frac{1}{P(E)}$ bits of information. This can be interpreted as that less likely events yield more information.

3.3.2 Entropy

An information source S which emits results $s_1, s_2, ..., s_i$ with probabilities $p_1, p_2, ..., p_i$ produces information at $H(S) = \sum_i^k p_i I(s_i)$. $H(S)$ is known as the information entropy of S. Information entropy can vary between 0, which indicates no uncertainty (i.e. all members of S in same class), and 1, which indicates high uncertainty (i.e. equal probability of all classes). The best split of data for classification is one that maximally decreases its entropy (a concept known as information gain).

3.4 Overfitting

A hypothesis $h_1 \in H$ is said to overfit training data if there is some alternative hypothesis $h_2 \in H$ that has a larger error over the training data but a smaller error over a larger set of inputs. Overfitting can occur due to errors in a data set or just due to coincidental irregularities (especially in a small dataset). Overfitting can be avoided by removing nodes with low information gain, either by stopping decision tree construction early or by pruning such nodes after the tree is constructed.

3.5 Inductive Bias

Inductive bias refers to the assumptions made about the target function to predict future outputs. Common inductive biases for decision trees are:

- Assumption that simplest hypothesis is the best (i.e. Occam's razor).
- Decision trees with information gain closer to the root are considered better.

These are examples of preference bias, which influence the ordering of the hypothesis space. This is distinct from restriction bias, which limits the hypothesis space.

3.6 Advantages and Disadvantages

Decision trees are good for:

- Ease of interpretation
- Speed of learning

Limitations of decision trees include:

- High sensitivity, with tree output changing significantly due to small changes in input.
- Not good for learning data sets without axis-orthogonal (i.e. constant in at least 1 dimension) decision boundaries.