CO70050 Introduction to Machine Learning

Lectured by Josiah Wang

Typed by Aris Zhu Yi Qing

October 18, 2021

Contents

1 Definitions **1**

2 Classification **2**

2.1 Instance-based Learning 2

2.1.1 k Nearest Neighbours (k-NN) classifier 2

2.1.2 Distance-weighted k-NN 3

2.1.3 k-NN regression (quick intro) 3

2.2 Decision Trees 3

2.2.1 Intuitions and Introduction 3

2.2.2 Selecting the optimial splitting rule 3

2.2.3 Comments 4

1 Definitions

1. **Artificial Intelligence**: Techniques that enable computers to mimic human behaviour and intelligence. It could be using logic, if-then rules, machine learning, etc.

2. **Machine Learning(ML)**: Subset of AI techniques using statistical methods that enable the systems to learn and improve with experience.

   - More data means more accurate predictions.
   - Select/Extract good features for predictions. more feature $\not\Rightarrow$ better prediction (curse of dimensionality: increased computational complexity, data sparsity, overfitting)
   - Pipeline: feature encoding, ML algorithm, and evaluation.

3. **Deep Learning**: Subset of machine learning techniques using multilayer Artificial Neural Networks(ANN) and vast amounts of data for learning.

4. **Supervised learning**: Take input variables and correct output labels as inputs, feed them into a supervised learning algorithm to generate a model which can be used to estimate labels of other input variables.

   - **Semi-supervised learning**: Some data have labels, some do not.

- **Weakly-supervised learning**: Inexact output labels.

5. **Unsupervised learning**: Take input variables only, feed them into an unsupervised learning algorithm to generate a model which can be used to estimate labels of other input variables.

   - discover hidden/latent structure within the data ("lossy data compression")

6. **Reinforcement learning**: Largely the same as unsupervised learning, except that the estimated labels at the end "interact with an environment" and send reward signal back to the reinforcement learning algorithm such that the algorithm will take the reward signal into consideration when learning the model next time.

   - find which action an agent should take, depending on its current state, to maximise the received rewards (Policy search)

7. **Classification**: The task of approximating a mapping function from input variables to discrete output variables.

8. **Regression**: The task of approximating a mapping function from input variables to continuous output variables.

9. **Lazy Learner**: Stores the training examples and postpones generalising beyond these data until an explicit request is made at test time.

10. **Eager Learner**: Constructs a general, explicit description of the target function based on the provided training examples.

11. **Non-parametric model**: Assume that data distribution cannot be defined in terms of a finite set of parameters. The distribution depends on the data themselves.

12. **Underfitting/high bias**: a lot of errors, oversimplified assumptions.

13. **Overfitting/high variance**: fits "perfectly" the training data, and may not fit the test data well.

2 Classification

2.1 Instance-based Learning

2.1.1 k Nearest Neighbours (k-NN) classifier

1. Non-parametric model
2. Lazy learner
3. Procedure: Obtain k nearest data, classify the instance under the class which the most number of neighbours belong to.
4. k is usually an odd number.
5. Increasing k will make the classifier
   - have a smoother decision boundary (higher bias)
   - less sensitive to training data (lower variance)
6. Various distance metrics, such as:
   - Manhattan distance (L^1-norm):
   $$d(x^{(i)}, x^{(q)}) = \sum_{k=1}^{K} \left| x_k^{(i)} - x_k^{(q)} \right|,$$
   - Euclidean distance (L^2-norm):
   $$d(x^{(i)}, x^{(q)}) = \sqrt{\sum_{k=1}^{K} \left(x_k^{(i)} - x_k^{(q)} \right)^2},$$
   - Chebyshev distance (L^∞-norm):
   $$d(x^{(i)}, x^{(q)}) = \max_{k=1}^{K} \left| x_k^{(i)} - x_k^{(q)} \right|.$$
7. the curse of dimensionality
   - k-NN relies on distance metrics, which may not work well if using all features in high dimensional space.
   - If many features are irrelevant, instances belonging to the same class may be far from each other.
   - Solution: Weight each feature differently, or perform feature selection/extraction.

2.1.2 Distance-weighted k-NN

1. Any measure favouring the votes of nearby neighbours will work, such as:
   - Inverse of distance: $w^{(i)} = \frac{1}{d(x^{(i)}, x^{(q)})}$,
   - Gaussian distribution: $w^{(i)} = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{d(x^{(i)}, x^{(q)})^2}{2}\right)$.
2. k is not so important in distance-weighted k-NN. Distant data will have small weights and won't greatly affect classification.
3. if $k = N$ (size of the training set): global method. Otherwise, it is a local method.
4. Robust to noisy training data — the impact of isolated noise is smoothened out.
5. simple yet powerful, but might be slow for large datasets

2.1.3 k-NN regression (quick intro)

1. compute the (weighted) mean value across k nearest neighbours.

2.2 Decision Trees

2.2.1 Intuitions and Introduction

1. Eager learner
2. Decision Tree learning (or construction/induction) is a method for approximating discrete classification functions by means of a tree-based representation.
3. A decision tree can be represented as a set of if-then rules.
4. Decision Tree learning algo employ top-down greedy search through the space of possible solutions.
5. General Algorithm:
   (a) Search for an 'optimal' splitting rule on training data.
   (b) Split your dataset according to the chosen splitting rule
   (c) Repeat 5a and 5b on each new splitted subset unless the subset contains data of only one class.

2.2.2 Selecting the optimial splitting rule

1. several metrics:
   - Information gain: Used in ID3, C4.5
     - quantifies the reduction of information entropy
   - Gini impurity: Used in CART
   - Variance reduction: Used in CART
2. information entropy:
   - **Information Content** I is a quantity derived from the probability of an event occurring from a random variable X as
     $$I(x) = \log_2(\frac{1}{P(x)}) = -\log_2(P(x)),$$
     which could be interpreted as the amount of information required to fully determine the state of a random variable, and the definition should satisfy several conditions as specified here.
   - **Information Entropy** H of a discrete random variable X with its p.m.f. being $P(X)$ is defined as the expected amount of information content as following:
     $$H(X) = E[I(X)] = -\sum_{k=1}^{n} P(x_k) \log_2(P(x_k)),$$
     where n is the number of possible discrete values of X. For a p.d.f. $f(X)$, we can define the **continuous entropy** as
     $$H(X) = -\int_x f(x) \log_2(f(x)) \mathrm{d}x.$$
     - The analogy in continuous case is imperfect (it can have negative values) but is still often used in deep learning.
     - The p.d.f. is often unknown, but can be approximated with density estimation algorithms for instance.
3. Use Information Entropy to select the 'optimal' split rule.

- **Information Gain (IG)** is the difference between the initial entropy and the (weighted) average entropy of the produced subsets.
- Mathematically,

$$\mathrm{IG}(D, S) = H(D) - \sum_{s \in S} \frac{|s|}{|D|} H(s),$$

where D stands for the dataset, S stands for the subsets after splitting, and $|\cdot|$ is the cardinality operator.

2.2.3 Comments

1. prevent overfitting
   - Early stopping, e.g. max depth, min examples.
   - Pruning
     (i) Go through nodes which are connected only to leaf nodes.
     (ii) Turn each into a leaf node (with majority class label).
     (iii) Evaluate pruned tree on validation set. Prune if accuracy higher than unpruned.
     (iv) Repeat until all such nodes have been tested.
2. Random Forests
   - Many decision trees voting on the class label.
   - Each tree generated with random samples of training set (bagging) and random subset of features.
3. Regression Trees
   - Instead of a class label, each leaf node now predicts an $x \in \mathbb{R}$.
   - Use a different metric for splitting, e.g. variance reduction.