ECE M146 Introduction to Machine Learning

Prof. Lara Dolecek

ECE Department, UCLA

Today’s Lecture

Recap:

- Linear methods: Perceptron, logistic regression, linear regression

New topic:

- Decision Trees

Today's Lecture

Recap:

- Linear methods: Perceptron, logistic regression, linear regression

New topic:

- Decision Trees

Techniques we have learnt so far

- Perceptron algorithm for binary classification
- Linear regression via linear least squares or gradient descent
- Logistic regression for binary classification via gradient descent
- All these methods derive weight vector w, and the output value is a function of $w^T x$.

Recap: Linear models for classification and regression

- These methods are called **parametric methods**. They are parametrized by w and the choice of the function that relates it to the output.
- In particular, the complexity of vector w, and the decision boundary, does not grow with the number of points, N.

- There are also non-parametric methods.
- Main examples are decision trees and k nearest neighbors.
- (Complexity of the) decision boundary grows with N.

Today's Lecture

Recap:

- Linear methods: Perceptron, logistic regression, linear regression

New topic:

- Decision Trees

Decision Tree Overview

- Performs sequential inquiries on data attributes to arrive at the classification.
- Can be used for binary or multiclass classification.
- Can also be extended to regression.

- Example: medical diagnosis: operate on a patient, yes or no ?
  - Does the patient have high blood pressure?
  - Does the patient have high cholesterol ?
  - Etc.

Decision Tree Overview

- Good for capturing logical expressions (T/F)
- Often provide human interpretability

- Not good for parity functions
  - Example:

- Not good for non-axis aligned decision boundary
  - Example:

Decision Tree Example

- Creating a decision boundary on two attributes, x_1 and x_2.
- Decision Tree: Picture:

- Once we have the tree built, at testing time, go down the tree until a leaf node corresponding to the test point is reached.

Properties of the Decision Tree Classification

- This approach allows for modeling of fairly complicated decision boundaries.

- This approach is expressive in the sense that any Boolean combination of attributes can be represented.

- Example:

Now we have a sense of a decision tree can do. But how do we create one ?

- What is a sequence of questions that should be asked ?
  - Say we have K possible binary attributes. That is 2^K combinations!
- How does the size N of the training data set impact the answer ?
  - There could be several solutions that agree on the available part of the combinations.

- Finding the smallest decision tree that correctly classifies all training points is NP hard.

- Instead, we build the tree **greedily**.

Procedure

- Notation: Node n = root of the decision tree.
  Set $\mathcal{D}$ is the set of unclassified examples.
- While $\mathcal{D}$ is not an empty set:
  - Pick A as the “best” decision attribute
  - Assign A to n
  - For each value of A create a new descendant of n
  - Assign class values to descendants based on $\mathcal{D}$
  - Remove from $\mathcal{D}$ examples that are perfectly classified
- If $\mathcal{D}$ is empty, stop. Else, recurse over new leaf nodes.

How to pick the best attribute ?

1. Random choice: query on any attribute, chosen at random

2. Least/most value: choose an attribute with least/most possible values.
   - Example:

3. Maximum gain: choose the attribute that has the largest expected **information gain**.
   - Unlike 1 and 2, it captures how informative attributes are (statistical measure of goodness).

Capturing information gain -- example

- Suppose we have 80 data points, each specified by the vector $x=(x_1,x_2)$ of attributes and its label y. Assume that x_1,x_2, and y are all in the {T,F} set.

- If we split on x_1:
  - Under T, we can conclusively say that y=T. We have reduced uncertainty in y.
- If we split on x_2:
  - Under either T or F, we have not reduced uncertainty in y. Useless split.

Formalize the notion of the information gain

- Mathematical expression for the information gain (IG):

- H(Y) is called the entropy of Y and represents intrinsic uncertainty in Y.
- H(Y|X) is called the conditional entropy of Y given X, and represents the uncertainty in Y once X is revealed.

Entropy – binary case

- Consider a Bernoulli RV Z with parameter p.

- Entropy H(Z) is a measure of surprise.
- Formula and picture:

Entropy – non binary case

- Suppose now Z has PMF given as $P(Z = z_k) = p_k$ for $1 \leq k \leq K$.
- Entropy of H(Z) is written as:

- Note that the entropy does not depend on values z_k, but only on probabilities z_k.

Conditional Entropy

- We next discuss conditional entropy $H(Y|X)$.
- Let's first consider $H(Y|X=x_j)$:

- Average over all values of X to get:

Back to our binary example

- Recall the set up.

- We want to compare $IG(x_1,y)$ and $IG(x_2,y)$.

- General rule:

If we query on x_2

- Convert to probability

- Compute $H(y|x_2)$:

- What is $IG(x_2,y)$?

If we query on x_1

- Convert to probability

- Compute $H(y|x_1)$:
  - Why do we expect it to be < 1?
- What is $IG(x_1,y)$?

If we query on x_2

- Convert to probability

- Compute $H(y|x_2)$:

- What is $IG(x_2,y)$?

Relationship to cross entropy

- Today we discussed entropy/conditional entropy.
- Binary entropy:

- Last time, we discussed cross entropy loss in the logistic regression.
- Binary cross-entropy:

Let’s now look at a bigger example

- Our first example was fairly apparent. But in practice we don’t always have such clear cut choices.
- Bigger example:
- Note that some combinations didn’t even appear and note that some are contradictory.

How do we build the decision tree now ?

- First, let's compute the entropy of y, H(y):

- Next, compare IG(y,x_1) vs. IG(y,x_2):

Conditional entropy calculation

- To compute $H(y|x_1)$ we need both $H(y|x_1 = T)$ and $H(y|x_1 = F)$.

Conditional entropy calculation

- To compute $H(y|x_1)$ we need both $H(y|x_1 = T)$ and $H(y|x_1 = F)$.

Conditional entropy calculation

- To compute $H(y|x_2)$ we need both $H(y|x_2 = T)$ and $H(y|x_2=F)$.

Conditional entropy calculation

- To compute $H(y|x_2)$ we need both $H(y|x_2 = T)$ and $H(y|x_2 = F)$.

Sanity check on calculations

- If the attribute x is binary, and the label y is binary, we can check if the following is satisfied:

Now, let’s build our decision tree

Termination rules

1. If a node is pure, i.e., all its examples have the same label y, assign that label to the examples, and terminate.
   - Example:

2. If all the examples have the same attribute values, but different labels, assign the one given by the majority, and terminate:
   - Example:

   - In case of a tie, declare the label the one governed by the parent and terminate.

Termination rules – ctd.

- Otherwise, keep going until there is nothing left to query on.
- Note that in our last example, we could not get the training error to be zero. Why ?

Common issues with decision trees and how to overcome them

- Decision trees are prone to overfitting.
- E.g. Root node split governs the rest (recall that we are doing a greedy procedure).
- Solution #1: split the data into training set and validation set. Train on the training set, prune using the validation set.
- Example: out of N examples, 0.7*N are the training set, and the remaining 0.3*N are the validation set.
- These two sets are mutually exclusive!
- Solution #2: bagging = bootstrap aggregation (ensemble method)

Training DTs with a validation set

- Example:
- Suppose this is what the training set gives:

- Suppose this what the validation set gives:

Training DTs with a validation set

- How many are misclassified under the original rule?

- How many are misclassified if we didn’t split on x ?

- Prune the tree !

Decision trees for regression

- The discussion thus far was focused on the binary classification problem. DTs can also be done for regression.
- Illustration:

- All data points in the given region are assigned the same label.