CS 181 Notes

Bannus Van der Kloot

February 4, 2013

Contents

1 January 30th

1.1 Decision Trees

Loss Functions

Decision Trees Particular representation of a classifier. Node has attributes, edge has value of attribute, leaf has label prediction.

How can we construct a 'good' decision tree? Finding the best possible tree is NP-complete (iterating over all possible trees). Each decision partitions the data set. Data sets shrink exponentially quickly (if attributes are uniformly distributed).

ID3 Algorithm Given a set of data and attributes, which criterion should we use to split?

We need a quantitative measure of information. Shannon developed 'Information Theory.'

Information Content If we have discrete random variable X that takes K possible values, the **information content** of outcome x is given by:

$$I(x) = \log_2(\frac{1}{p(x)})$$

The units are **bits**. Imagine we have a message written with a four character alphabet? What is the best way to represent the message? If the distribution is uniform, we can represent each with two bits.

Information content: how much did my view of the world change when I observed this event?

Shannon Entropy is the expected information content under $p(x)$. Note: $\mathbb{E}$ refers to expectation:

$$H(x) = \mathbb{E}[I(x)] = \sum_x p(x) \log -2(\frac{1}{p(x)})$$

In the compression view, the entropy refers to the average code word length. With the four character alphabet, uniform distribution yields an entropy of 2. With the 1/2, 1/4, 1/8, 1/8 distribution, the entropy is 1.75.

Specific Conditional Entropy Considering two random variables X and Y given by $p(x, y)$. Then the **specific conditional entropy** is given by:

We can compute the marginal for Y:

$$p(Y) = \sum_x p(X, Y)$$

We can find the **conditional entropy**

Mutual Information The number of bits we know about X by knowing Y.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \end{aligned}$$

Similar to correlation, but correlation is sort of limited. Just talks about linear dependencies.

If we could compute the mutual information between each attribute and the label, we could maximize this. Problem: we don't know the true distribution. We have a set of discrete outcomes to compute distribution estimates though.

Decision Forests Collection of decision trees that are all helping to learn something. Kinect uses this to estimate player pose from depth information.

2 More on Decision Trees: Overfitting, Description Length, and Cross-Validation

Application: determining depth information from a single image.

ID3 ID3 algorithm: number of bits yielded from a decision as we move down the tree.

Observations about entropy: with a bernoulli R.V., the slopes are interesting. Deviations around 0 and 1 yield a large entropy difference, while deviations around .5 are not significant.

ID3 tends to prefer extreme partitions, also prefers classifying larger subsets well.

Note that ID3 is a greedy algorithm. Doesn't look ahead or backtrack. This could be annoying if there is a lot of interesting coupling, i.e. if there are a few attributes that taken together yield good information (**copredictors**).

Decision trees can represent concepts with copredictors, but ID3 is not good at learning them. Must understand both the representation and the algorithm to understand behavior or learning framework.

Consistent/inconsistent data: exactly the same features a second time with different data, is it possible to get different labels. Is there any label noise, or do we believe it 100% of the time?

Pure/impure data: All positive/all negative is pure (all the same label, independent of features), impure is mixed.

Overfitting Conjecture: if A has lower training error than B, it should generalize better to unseen instances. WRONG

This is due to overfitting. At a certain point, we design the hypothesis too specifically to the training data, and the algorithm gets worse on test data.

Two patterns in data: **true** patterns in domain, **spurious** patterns in training set (perhaps due to noise).

At the beginning, ID3 has a large amount of data, tends to discover "true" patterns

Dealing with Overfitting Can be caused by:

- Small training set
- Non-deterministic domain (noisy/inconsistent data)
- Many features
- Weak inductive bias

Weaker inductive bias: Depends more on what you've seen than on what you expect. Algorithm more influenced by data, can more easily learn patterns, may overfit.

Stronger inductive bias: More attention to hypothesis, less attention on data, focus on real signal. Possibility of underfitting.

Increasing inductive bias:

Restriction bias: Take some subset of hypothesis space under consideration, stick with that. Idea would be choosing hypotheses that are considered "simple" by some metric.

Preference bias: rank the hypotheses in some sense. Rather than having a hard truncation, come up with a penalty that prefers shallower trees, or prefers lower order polynomials, or prefers coefficients closer to 0, etc.

Preference bias in decision trees

- Pre-pruning
  - Prune the subtree before you grow it
  - Add termination condition to ID3
- Post-pruning
  - Grow complete tree, prune afterward

Pre-pruning How likely is it we'd see something this extreme under the null hypothesis? If we come up with a threshold at which if the p value is small, we reject the null hypothesis that something is spurious and accept it as actual signal.

This is called **chi-squared pruning**. This plugs right into our greedy procedure (simple and fast). However, you have to tune the threshold (another parameter in the model), it also doesn't use much information.

Pre-pruning Generate and prune.

Cross validation Take a big data set, chop into partitions. Run algorithm 5 times (train on union of 4 folds, use each partition as test set).

Common thing to do is 10-fold cross validation. At any given time, you get to see 90% of the data.