COMP 9 / EN47 - Exploring Computer Science

Lecture 6: Introduction to Machine Learning

February 21, 2011

1 What is machine learning?

Machine learning is the discipline of computer science focused on *algorithms* for automatically finding patterns in data, allowing *classification* and *prediction*.

2 Motivation

Machine learning has applications in many fields, from the sciences to marketing, and from medicine to music.

Examples: - Netflix and Amazon recommendations - iTunes genius playlists - Pandora - Medical diagnostics - Is a tissue sample cancerous? Is a spot in an MRI worth alerting a doctor about? - U.S. Customs record data mining - Panjiva.com - Natural language processing and document classification - turnitin.com

3 Basics

Classification is when two or more distinct classes are learned from data: - Cancer vs. non-cancer - Music genre - Will you like or dislike a particular movie

Regression is when a *continuous-valued function* is learned from the data: - What will the revenue from a pharmaceutical company be based on usage data? - How long will a patient live based on the phenotype of her cancer? - What is the mass of a star based on information about brightness, distance, and motion?

Labels are the classes we learn in a classification scheme, or the value we learn in a regression scheme.

Features are all the other data we know about the samples in our data set: - brightness, distance, motion of a star - genetic markers for a tumor, or expression levels for various genes - they are the data we can *use to predict the label*

Training data is data where we know the label; in *supervised learning* we must have some training data with known labels, or there is nothing to learn from. *Training data* is what we use to learn some pattern.

Test data is data where we know the label, and we do not learn a pattern from it; we use test data to evaluate the performance of a machine learning system.

In *unsupervised learning*, labels are not used; we simply look for patterns based on the *features* in the data, so there is no separation of training data and test data.

4 Decision trees

A *decision tree* is simply a flow chart, within which each node represents a *question* to be asked about the features of the data. Leaf nodes represent assignments of specific samples to some class.

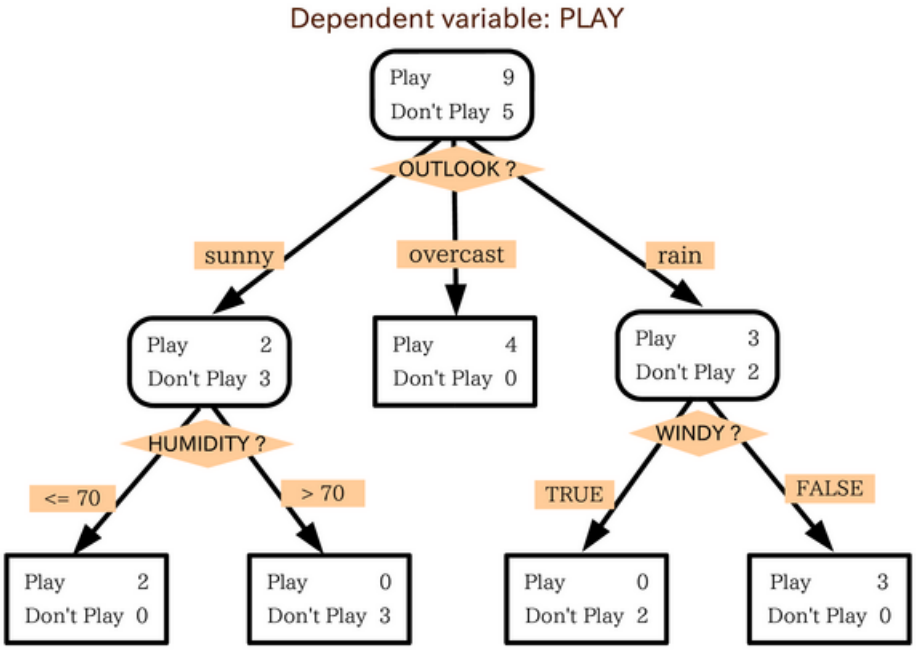

Figure 1: decision tree

The ID3 algorithm, which stands for Iterative Dichotomizer version 3, was developed by Ross Quinlan in 1975. The algorithm is as follows (note that `Attributes` are *features* and `Target_Attribute` is the *label*)

```
ID3 (Examples, Target_Attribute, Attributes)
```

```
Create a root node for the tree
If all examples are positive, Return the single-node tree Root, with label = +.
If all examples are negative, Return the single-node tree Root, with label = -.
If number of predicting attributes is empty, then
    Return the single node tree Root, with label = most common
        value of the target attribute in the examples.
Otherwise Begin
    A = The Attribute that best classifies examples.
    Decision Tree attribute for Root = A.
    For each possible value, v_i, of A,
        Add a new tree branch below Root, corresponding to the test A = v_i.
        Let Examples(v_i), be the subset of examples that have the value v_i for A
        If Examples(v_i) is empty
            Then below this new branch add a leaf node
                with label = most common target value in the examples
        Else
            below this new branch add the subtree
                ID3(Examples(v_i), Target_Attribute, Attributes { {A})
End
Return Root
```

The glaring question in this algorithm is how to determine which attribute *best* classifies the examples? In practice, *information gain* is used:

$$E(S) = -\sum_{j=1}^{n} f_S(j) \log_2 f_S(j)$$

Figure 2: entropy

In this equation:

- E(S) is the information entropy of the subset S ;
- n is the number of different values of the attribute in S (entropy is computed for one chosen attribute)
- $f_S(j)$ is the frequency (proportion) of the value j in the subset S

Given Entropy, Information Gain is:

In this equation:

- G(S,A) is the gain of the subset S after a split over the A attribute

$$G(S, A) = E(S) - \sum_{i=1}^{m} f_S(A_i) E(S_{A_i})$$

Figure 3: gain

- E(S) is the information entropy of the subset S
- m is the number of different values of the attribute A in S
- $f_S(A_i)$ is the frequency (proportion) of the items possessing A_i as value for A in S
- A_i is ith possible value of A
- S_{Ai} is a subset of S containing all items where the value of A is A_i

For our purposes, we can simply *count* how many new examples are clarified by each attribute, and choose this as the best example.

5 Pitfalls

5.1 Insufficient sample size

Without enough training data, no machine learning algorithm will be able to learn an adequate model. Yet sometimes, the realities of data sources (laboratory experiments, telescope time, money) mean we have barely adequate training data. Much of the research in the field of machine learning revolves around how to perform well with less training data, or how to augment training data.

5.2 Noisy data

Often, training data are not entirely *correct*. Noise due to instruments, measurement uncertainty, or human error means that machine learning algorithms need to tolerate occasionally noisy data; this is another active area of research.

5.3 Overfitting or overtraining

Most machine learning algorithms *terminate* when they have reached some threshold of performance on the *training data* or a seperate *test data* set. The risk is that the algorithm learns too many irrelevant specifics of the training

data set, and performs poorly on other data. Detecting overfitting is another active area of research.