Fabinger's Data Science Class Notes, July 8th-14th 2019

Jose Zero

July 15, 2019

Abstract

Decision Trees Class / A short topic in this course. Given a tabulated dataset we want to split it one feature at a time to gain insight in the classification. We measure the gain in the split of by each feature and we choose first the one that gains most information. We use Entropy or Gini measures to decide how much information has been gained. JZero's Note: Example in video 1 below is very good. Comparison between Gini, Entropy and q-Tsalling entropies added.

1 Introduction

In a similar fashion as the game of twenty questions we want to decide if a dgiven student speaks Japanese. Given a training dataset ("the dataset")for which the answers are known we can statistically compute (hypothesis test) if the result splitting coincides with the answer sought for.

We can ask –for example– if the student's last name is Japanese sounding. If it is then we can measure the number of 1) how many students from the whole dataset that represents and from those how many do speak Japanese. From the ones that do not have a Japanese sounding last name we can measure the same and test two areas. How much of the dataset we are measuring and how nicely the split coincides with the split of Japanese speaking students.

The representation is usually a question represented as an oval with yes and now arrows coming of it. The first question on top is called "root", the following questions are called internal nodes and the ends (final counts with no further questions "leaves."

If the dataset contains several features we decide which feature to split by ("ask") first according to how much information is gained. We call this process "growing the tree." Also see video 1.

More formally:

Given a tabulated type of a dataset with K rows $X_i^j, j \leq K, i \leq n$ and category Y^i where $X^i \in R^n$. That means we have n features and K elements in our dataset and at most K categories given by the values of Y.

Let's say you have a variable V with values zero or one and probabilities $p(V=0)=.1$ and $p(V=1)=.9$ the values $-log(p(V=0)$ and $-log(p(V=1)$ correspond to the "surprise" of obtaining one of those values. Note that the minus sign is used to flip the negative value of a log of a number less than one. Keeping this in mind we define Entropy.For a given splitting V we can compute the Entropy as

$$Entropy(V) = -\sum_{\nu} p(V=\nu) log p(V=\nu) \tag{1}$$

In many cases the distribution, mean and standard variations are not know. In those cases z-tests and t-tests can be used. See discussion on recommended materials.

Where ν runs over the different partitions. We can also calculate the total entropy as the weighted average of partition entropies according to number of members of t

Entropy gives an idea of how diverse the dataset is. The less entropy, less diversity, more homogeneity and hence better predictions.

The information gained in a partition is the difference between the weighted totality of the dataset entropy minus the entropy of the partition. The attribute that generates the largest information gained should be used as root.

$$I(A,B) = E_B - E_{partitions} = H(B) - \sum_{a} p(A=a) H(B|A=a) \tag{2}$$

Some sports with high scoring need to boost entropy to avoid predictability. For example tennis. If left for scoring the results would approximate players statistical characteristics and would be predictable (or more predictable) and hence not so interesting. By grouping scoring into matches the game has greater entropy and hence less predictability (more interesting to the audience.)

a more general formula for entropy is the q-Tsallis entropy:

$H(V) = [1 - \sum_{\nu} p(V=\nu)^q]/(q-1)$

For $q=1$ is called the Shannon Entropy and for $q=2$ the Gini impurity. For $q=2$ it is trivial to verify. To prove this for $q=1$ we use implicit logarithmic derivative and LHopitals rule. Given $y=p^x$ taking logarithms on both sides get $ln(y) = x$ *ln(p) and deriving both members and using the chain rule $y^{'}/y = ln(p)$ from which

$$y^{'} = p^x ln(p) \tag{3}$$

This together with L'Hopital Rule and the fact that the sum is finite (not true for infinite series in general)get:

$$H^1(V) = \lim_{q\to 1}[1 - \sum_{\nu} p(V=\nu)^q]/(q-1) = \lim_{q\to 1}[1 - \sum_{\nu} p(V=\nu)^q]^{'}/(q-1)^{'} \tag{4}$$

Hence

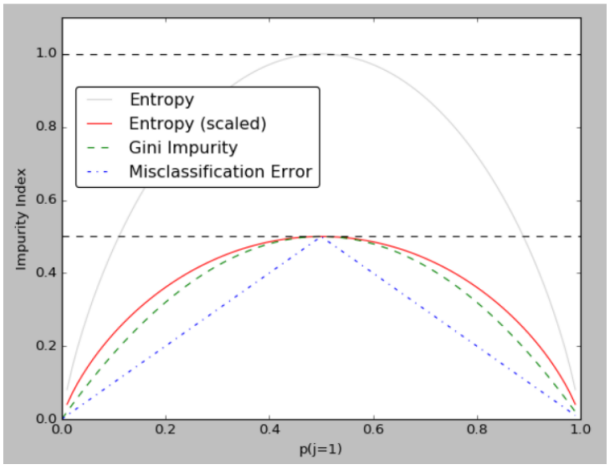

Figure 1: Entropy vs Gini in function op p for binary variable V See recommended link.

$$H^1 = -\sum_{\nu} log(p(V = \nu)p(V = \nu) \tag{5}$$

Let's develop some intuition. Let's say that you have come up with two partition of a 100 element classified dataset and for the first split you have:

A:50 examples, 3 misclassified B:50 examples, 30 misclassified (total 33 misclasified)

and for the second

A: 50 examples, 13 misclassified and B: 50 examples, 20 misclassified (total 33 misclassified)

which one do you take?

The answer lies in the first one as the first partition gives a very good result (only 3 misclassified.) Subsequent partitions (leaves) of the B partition in the first case will improve the scoring of the 30 misclassified.

Let's consider a partition according to a single variable V with binary values. Let p be the probabilityfor V=0 and hence (1-p) for V=1. In this case,

$$Gini(p) = 1 - p^2 - (1 - p)^2 = (1 - p)(1 + p) - (1 - p^2) = 2p(1 - p) \tag{6}$$

and

$$E(p) = -p\, log(p) - (1 - p)\, log(1 - p) \tag{7}$$

For the case in which p is relatively small, say .01 Gini(p) will be .198 and E(p)=0.0559

1.1 Recommended materials

Video 1 on splitting datasets and how to decide by what feature to split first, entropy and Gini measure: https://youtu.be/IPkRVpXtbdY

Very good discussion about Gini impurity vs Entropy:
https://datascience.stackexchange.com/questions/10228/when-should-i-use-gini-impurity-as

For a nice discussion on z-tests and t-tests see:
http://www.stat.yale.edu/Courses/1997-98/101/sigtest.htm

Graphics for gini vs shannon: https://devopedia.org/decision-trees-for-machine-learning

L'Hopital's rule reference:
https://en.m.wikipedia.org/wiki/L'H\unhbox\voidb@x\bgroup\let\unhbox\
voidb@x\setbox\@tempboxa\hbox{o\global\mathchardef\accent@spacefactor\
spacefactor}\accent94o\egroup\spacefactor\accent@spacefactorpital'
s_rule