NLP for Smart Baseball Scouting

Stephen Cha, Tyra Pitts, Daniel Crawford,
MATH 1103 - B.I.G. Problems, Department of Mathematics, University of Pittsburgh

Department of Mathematics

Motivation

Scouting professional athletes has become increasingly important with an ever-growing financial investment. In baseball, many strategies are employed to diagnose a player's skill and potential. One useful avenue is the analysis of comments given by experts on that player. Each player generates some level of "buzz" and baseball scouts, commentators, and journalists capture that buzz in text/speech, thus encoding important information about that player.

Preprocessing

In baseball, players are rated on a 20-80 scale, following an approximately normal distribution. The ratings are only multiples of 5, e.g. 35, 60. One of the most important tasks is for us to contextualize ratings. Some commentators score differently, thus, their score distributions are not all the same. So, we sought to transform these instances of data so that all the distributions followed $N(50, 10)$.

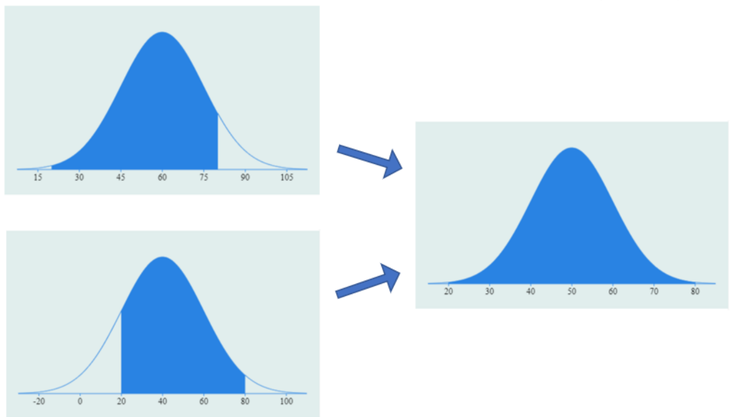


Figure 1: Sources with different distributions (left) were normalized to follow N(50,10) (right).

Skills Learned

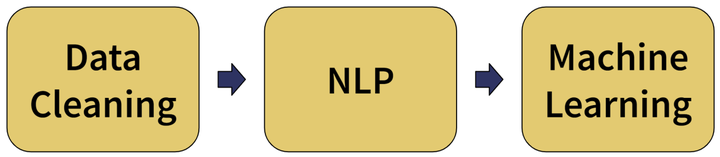
Machine learning, natural language processing, Python, R, communication with client, effective teamwork, cooperative collaboration, and perseverance despite two previous project cancellations.

Problem Statement

We would like to know if, **given just a string of text from a sports commentator about a particular player, a reasonable grade for a player on the 20-80 scale can be made.** That is, we need to predict what grade a commentator might give a player in the absence of a 20-80 scale score.

Strategies

This problem can be broken down into three large steps:



We use machine learning to predict what grade a commentator might give a player in the absence of a 20-80 scale score. Some algorithms we are using are below:

Support Vector Machines: Each data point lies in hyperspace which we can then classify with support vectors:

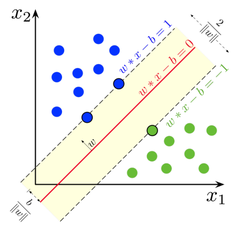


Figure 2: A 2-dimensional example

We are also using **ensemble learning methods** such as **decision trees**:

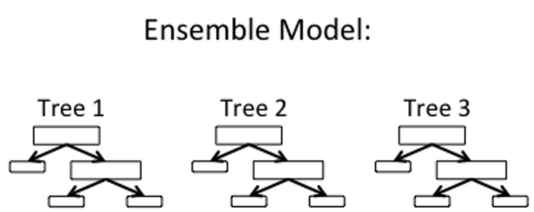


Figure 3: In decision trees, we have multiple models run on the data and base our classification on different attempts, "trees", to achieve higher accuracy.

Natural Language Processing

Machine learning algorithms cannot parse text. So, we need to convert the text data into something compatible with machine learning algorithms. We do this by transforming each comment into a **bag-of-words**. A bag-of-words is a representation of text that is compatible with machine learning algorithms. There are three types we are using: **presence**, **frequency**, and **term frequency-inverse document frequency (TF-IDF)**. Presence and frequency bag-of-words are illustrated in the table below:

"great baseball player, good catching, great hitting"

TYPE\TERM	"good"	"great"	"bad"
PRESENCE	True	True	False
FREQUENCY	1	2	0

We compute TF-IDF as follows:

$$w(t,c) = tf(t,c) \times \ln\left(\frac{N}{1+df(t)}\right)$$

where t is a term, c is a comment, $tf(t,c)$ is the number of times term t occurs in comment c, $df(t)$ is the number of comments that contain term t, N is the number of comments, and $w(t,c)$ is the TF-IDF value of term t in comment c.

TF-IDF increases proportionally to the number of times a term appears in a given comment, but is offset by the number of documents that have the term.

Acknowledgements

Joe Datz of the Pittsburgh Pirates, **Nathan Ong** of the University of Pittsburgh Computer Science, **Pitt Math Department** for funding and support, and our instructor **Jeff Wheeler**.

References

Timothy P. Jurka, Loren Collingwood, Amber E. Boydstun, Emiliano Grossman an Wouter van Atteveldt (2014). RTextTools: Automatic Text Classification via Supervised Learning. R package version 1.4.2 https://CRAN.R-project.org/package=RTextTools
Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

Images open on WikiPedia, Tutorial guides from TowardsDataScience.com