

Predicting YouTube Comedy Slam Winners

STAT 222 Class 2013

Department of Statistics
University of California, Berkeley

7 May 2013

YouTube Comedy Slam Data

- from UCI ML Repository; donated by Google employee
- a trial is an ordered pair of video IDs
- data: ordered pair + “left” or “right” found funnier by viewer
- order of the pair in each trial was random
- repository has training data and test data
- YouTube has metadata about videos
- no data about viewers

Goal

Predict which video in a pair will be funnier, from metadata

(Is it OK to use the video IDs in the prediction?)

Tools

- Github
- IPython notebook
- numpy, scipy, matplotlib, nltk, scikit-learn

Descriptive statistics of training data

- 912,969 records
- 18,474 distinct videos
- 267,211 distinct video pairs
- 359,874 distinct ordered pairs of videos
- right video won 51.77% of the time. P -value: nil
- judgments often discordant:
e.g., '-iuk0PbfaHY wDx28Y2RcCl' left and right each “funnier”
119 times
- accuracy of ideal classifier for training data: 73.449% (includes videos with no comments)
- for individual videos, directed graph of “funnier than”
- can summarize that graph by PageRank

PageRank

- Assign each video (node) in the graph a numerical value between 0 to 1, known as its *PageRank*.

At $t = 0$, $PR(p_i; 0) = \frac{1}{N}$, N is the total number of nodes.
$PR(p_i; t + 1) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j; t)}{L(p_j)}$
Algorithm ends when $ PR(t + 1) - PR(t) \leq \epsilon$
d is a damping factor, default 0.85 in scikit-learn.

Acquiring metadata

- queried YouTube with video IDs using Google APIs w/i Python
- speedbumps: had to throttle the requests
- stored “snapshot” data as pickled Python dict

Prediction using metadata: Feature selection

- used mutual information to screen potential features
- quantitative metadata (e.g., #views, rating, #raters) unhelpful
- comments more promising

Issues with comments

- Data Encoding: encoded the words to ASCII format and omit the garbled words
- Inconsistent Spellings
 - Correct Spelling: tried Norvig's spelling corrector, ported to AWS.
(e.g.: "spelng" → "spelling")
drawback: not accurate on this corpus
(e.g.: "youtube" → "couture", "lol" → "ll", "haha" → "hata")
 - Text-speak: used RegExp to standardize words

Patterns	As
lol, lolll, llol, lollloll	lol
ha, hahahh, ahaha, jhajha	ha

- Stemming: used Porter Stemmer from NLTK package
- Addressing Emoticons: used RegExps to replace happy faces (e.g., ":-]") with "happyface" before stripping other punctuation

Bags of Bags of Words

- ordered pair of videos reduced to two “bags of bags of words”
each comment is a bag, each video has a bag of bags
- features derived from bags of bags:
 - presence of a word in any comment
 - frequency of comments with a given word
 - relative frequency of comments with a given word
 - logOdds of frequencies and relative frequencies
 - etc.
- another derived feature: PageRank predicted by linear regression using (among other things) $\arctan(\text{difference in LOL counts})$

Classifiers

Logistic regression

Explain

CART

Explain

Classifiers

Bayes Classifiers

- 1 determine each video id to winning or losing ids' groups.
- 2 for each features, calculate $\mathbb{P}_1(w)$ = percentage of occurrence among the winning ids' comments and $\mathbb{P}_0(w)$ = percentage of occurrence among the losing ids' comments.
- 3 calculate the $\text{logOdds} = \log\left(\frac{\mathbb{P}(\text{winner}|\text{comments})}{\mathbb{P}(\text{loser}|\text{comments})}\right)$ for each video id:

$$\text{logOdds} = \sum_{w \in \text{features}} m_w \log\left(\frac{\mathbb{P}_1(w)}{\mathbb{P}_0(w)}\right) + (n - m_w) \log\left(\frac{1 - \mathbb{P}_1(w)}{1 - \mathbb{P}_0(w)}\right)$$

where n is the number of comments in the video and m_w is the number of comments contains w .

Performance on the training set

The Test data

- 225,593 records
- 75,447 distinct ordered pairs of videos

data?

- Accuracy of the ideal classifier: 0.6946 (excluding pairs with no comments)
- right video won: 51.62% of the time

Performance on test data

- Bayes classifier from training data applied to test data
- logistic regression
- CART

Conclusions

the problem and the data

- Hard problem: taste in comedy is personal, but no data on viewers
- Even Bayes classifier only gets about 70% accuracy
- Surprising that “right” has such a big advantage
-

Lessons learned

Tools & environment (github, IPython, etc.)