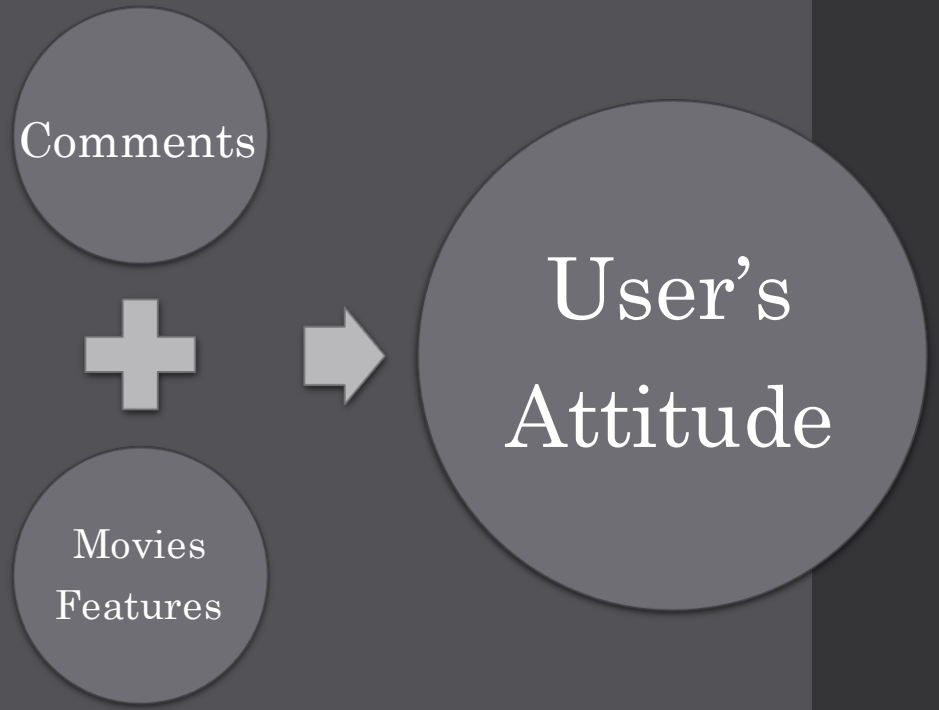
Classify user's rating based on IMDB data

Huiyu Bi, Miao Wang, Yuan Tian

June 1st 2016

Predictors:

- From users' comments:
  - The counts of each word in Bow (Bag of Words) appearing in one user's one comment of a movie
  - The total number of "positive word"/ "negative word" in the comment
  - The length of the comment
  - The total number of transitional words (but, though…) in the comment
- From features of the movie:
  - Production year
  - Director
  - Actors
  - Runtime
  - IMDB rating and Votes
  - Languages
  - …



Classification goal:

A user's attitude towards a movie –
positive (rating ≤ 4) or negative (rating ≥ 7)

Source Data & Data Processing – Part 1

- Source: Large Movie Review Dataset
- Raw data: 50000 comments (train 25000, test 25000) along with corresponding user's rating (either ≤ 4 or ≥ 7) and movie ID
- Bag of Words: 89527 words appearing in all the comments
- Data processing:
  - X: Counts of each word in Bow in each comment.
  - If rating ≤ 4, $y = 0$; if rating ≥ 7, $y = 1$.
- Processed Data: Each comment is turned into a vector of length 89528, including y and 89527 x.

Source Data & Data Processing – Part 1 (Cont'd) – Eliminating word variables

- 89527 is too much. We only kept the "important" words.
- What words are "important"?

- 1. We eliminated 161 stop words such as "I", "again", "most", "when"…
- 2. We eliminated "rare words" whose proportion in the total number of words is < 0.00001.
- 3. We kept the "intense sentiment words", that is the ratio of this word's total number in positive (negative) comments to its total number in negative (positive) comments is > 2.

Finally, only 3139 words left.

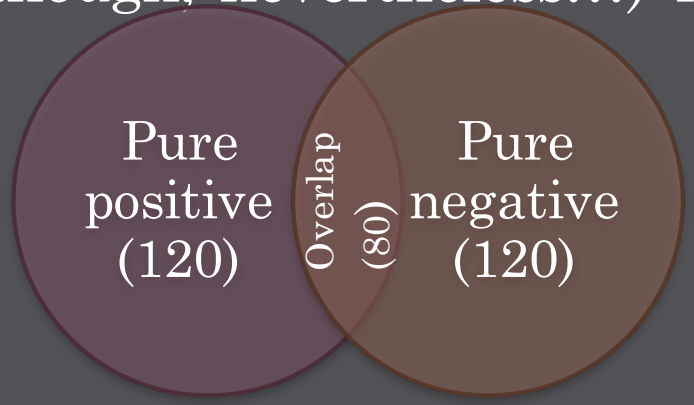
- Final data form: Each comment is turned into a vector of length 3140, including y and 3139 x.

Source Data & Data Processing – Part 1 (Cont'd) – Feature engineering

We also created 4 new comment feature variables according to the comments data.

- 1. Total number of pure positive words and pure negative words in one comment
  - Pure positive words: 120 exclusive most frequently appeared words in positive comments. E.g. fantastic, superb, perfectly, powerful, incredible, sweet, awesome…
  - Pure negative words: 120 exclusive most frequently appeared words in negative comments. E.g. Awful, waste, horrible, crap, ridiculous, dull, lame, poorly, badly…
- 2. length of the comment
- 3. Total number of transitional words (but, though, nevertheless…) in the comment.

We also tried tf-idf transformation.



Pure positive words

Pure negative words

Source Data & Data Processing – Part 2

- Source: OMDb API (Open Movie dataset)
- Raw data: Information of 7036 Movies rated by 50000 comments
- Potential variables:
  - Numeric: Production year, Runtime, Awards, imdbRating, imdbVotes
  - Categorical: Type, MPAA rate, Language, Genre, Director, Actors

{"Title":"The Guardian","Year":"2006","Rated":"PG-13","Released":"29 Sep 2006","Runtime":"139 min","Genre":"Action, Adventure, Drama","Director":"Andrew Davis","Writer":"Ron L. Brinkerhoff","Actors":"Kevin Costner, Ashton Kutcher, Sela Ward, Melissa Sagemiller","Plot":"A high school swim champion with a troubled past enrolls in the U.S. Coast Guard's \"A\" School, where legendary rescue swimmer Ben Randall teaches him some hard lessons about loss, love, and self-sacrifice.","Language":"English","Country":"USA","Awards":"1 win & 4 nominations.","Poster":"http://ia.media-imdb.com/images/M/MV5BMTI0MDkzMzQ1M15BMl5BanBnXkFtZTcwMDQ3MTQzMQ@@._V1_SX300.jpg","Metascore":"53","imdbRating":"6.8","imdbVotes":"74,281","imdbID":"tt0406816","Type":"movie","Response":"True"}

Source Data & Data Processing – Part 2 (Cont'd) – Variable Transformation

- Numerical variables transformation:

- 1. We transformed "Award" to 4 variables depending on if the award is famous or not : "famous_wins", "other_wins", "famous_nominations", "other_nominations".
- 2. We created a new variable called "number of languages".
- 3. We replaced NA in "Runtime" and "imdbVotes" by their median.

- Categorical variables transformation:

- 1. We transformed "Type", "MPAA rate", "Language", "Genre" into dummy variables. We also created several new dummy variables relating to the movie's main language.
- 2. We transformed "Director" and "Actors" into numerical variables.

Source Data & Data Processing – Part 2 (Cont'd) – Variable Transformation

How did we transform "Director" and "Actors" into numerical variables?

- Source: Box Office Mojo (Website)
- Raw data: Top 852 directors and top 787 actors ordered by their total gross box office.
- Data processing: We transformed the directors and actors to a rating of scale 10 according to the following table:

Box Office Rank	1-50	50-100	100-200	200-400	400-800	Not in the list
Our rating	10	9	7	4	1	0

Candidate Datasets

-We want to know which combination of predictors has the best prediction performance

1. ORI (3139 predictors)
– Containing all the word counts variables (3139)

2. Plus1 (3143 predictors)
– Containing all the word counts variables (3139) + comment feature variables (4)

3. Plus2 (3279 predictors)
– Containing all the word counts variables (3139) + movie feature variables (140)

4. PlusPlus (3283 predictors)
– Containing all the word counts variables (3139) + comment feature variables (4) + movie feature variables (140)

5. Plusplus w/ tf-idf (3283predictors)
– Containing tf-idf transformed word counts variables (3139) + comment feature variables (4) + movie feature variables (140)

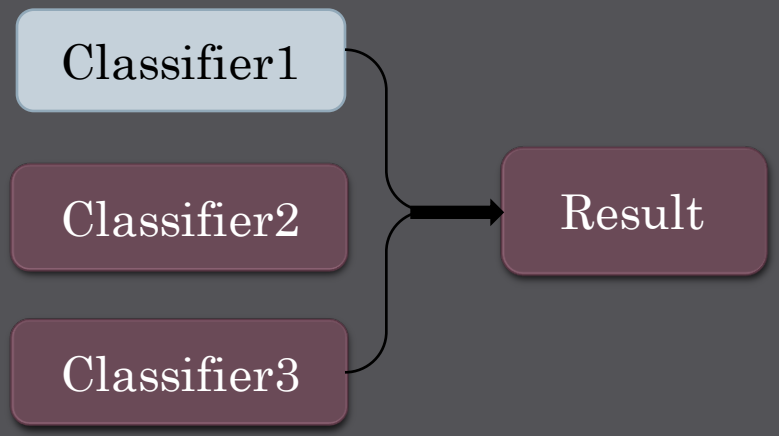
Model Selection

Test error	Logistic (c = 0.0001)	Random Forest	XGBoost (md = 6, eta = 1, nth = 3, nround = 14)	SVM RBF
ORI	0.1489	0.1672 (n = 200)	0.1875	0.1861
Plus1	0.1492	0.1590 (n = 160)	0.1825	0.1855
Plus2	0.1494	0.1639 (n = 180)	0.1930	0.1824
PlusPlus	0.1486	0.1543 (n = 190)	0.1741	0.1833
Plusplus w/ tf-idf	0.1575	0.1545 (n = 190)	0.1826	0.1835

We found that dataset “PlusPlus” has the best prediction performance.
And tf-idf transformation did not improve the performance.

Voting – Final Model

We conducted voting among the first 3 best methods (logistic, RF, and XGBoost) to get a better prediction.



Probability	Class 1	Class2
Classifier1	0.3	0.7
Classifier2	0.5	0.5
Classifier3	0.6	0.4

Assign three weight to classifier w1,w2,w3 = 1,3,3

Class 1	Class2
0.3*1	0.7*1
0.5*3	0.5*3
0.6*3	0.4*3
1.2	1.13

Result

Hard Voting
Test error = 0.1473

Soft Voting
Test error = 0.1432

Further Discussion

1. Feature selection and tune parameters.
2. Feature engineering

- Explore more variables of use to explain the rating behavior.

  E.g. box office of the movie, production country…

- Take the relationships among variables into consideration.

  E.g. The collaboration of certain director and actor, the combination of certain director/actor and movie genre may affect the user's rating.