Popularity Prediction of Hacker News Posts

Zhenglin Geng, Yang Yuan, Chao Wang ({zhenglin, yyuan16, cwang17}@stanford.edu)

Summary

A large amount of news are produced every day but only a small portion become popular. News popularity prediction is a meaningful task that can benefit content providers. And trying to predict in advance using only news content is even more challenging. We're building a popularity prediction system by analyzing the corpus of Hacker News posts.

Data

Source: Our dataset is obtained from kaggle, which contains title, url, num_points, num_comments, author and create_at datafields.

Preprocess:

- Grab web content and extract main contents;
- Filter out bad outputs;
- Remove numbers, symbols, stop words;
- Tokenize and stem text;
- For efficiency, 21000 posts are picked to be training data and 9000 to be test data.

Labeling: Posts with more than 30 votes are labeled 1 (popular) and others 0 (unpopular). For multiclass, split further into 5 classes using thresholds 1, 2, 3, 30.

Resampling: Since our dataset is extremely imbalanced (1:9.1), we applied both under sampling and over sampling to our training set.

Models

Binary classifiers:

- Linear support vector classification with L2 loss, L1, L2 or Elastic-Net regularization
- Stochastic gradient descent SVM with L1 or L2 regularization
- Multinomial and Bernoulli Naïve Bayes, Ridge, Perceptron, Passive Aggressive, KNN, Random Forest Tree

Multinomial classifiers:

- Inherent multiclass models: Multinomial Naïve Bayes, Decision Tree
- One Vs Rest Classifier using Linear support vector classification and SGD SVM as classifier for each class against the rest

Features

N-gram: A contiguous sequence of n tokens are used as n-gram.

Counts & tf-idf: We both count n-gram patterns and transform it into term frequency–inverse document frequency.

Word2vec: Token is represented as a vector and Document vector is calculated by taking mean or tf-idf of word vectors.

Topics: We use non-negative matrix factorization to extract topics in the corpus and use distance to top 10 topics as features of each document.

Topic 1	js javascript node react component python config
Topic 2	apple iPhone iPad FBI mac cook sir headphone
Topic 3	invest startup fund venture capital entrepreneur
Topic 4	game Pokémon VR Nintendo virtual oculus
Topic 5	email Clinton website inbox Gmail slack click
Topic 7	tesla vehicle musk autopilot elect autonomous
Topic 8	Facebook twitter advertise Zuckerberg messenger
Topic11	docker deploy cloud kubernet cluster config
Topic 12	ai artificial neurual deep alphago deepmind
Topic 14	uber ride city lyft vehicle hail self did transport

Table1: Some Selected Topics When Topic Number Set To 20

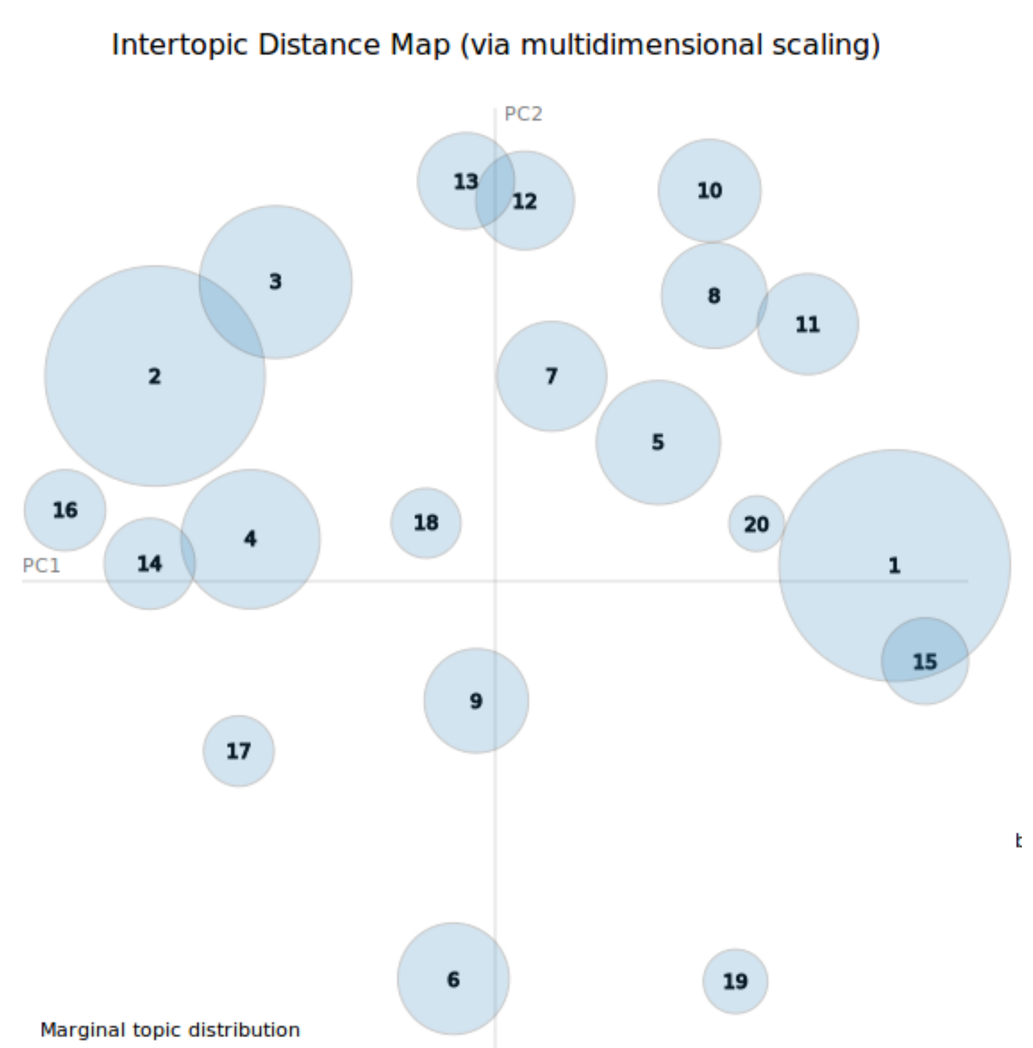

Figure1: Topic Visualization Using pvLDAvis[1]

Results

Binary classification

Since predicting the popularity based on news text is difficult in nature[2], statistical significance is used to measure the effectiveness of our classifiers against random classifiers.

We focus on precision on predicting popular news. We choose $\alpha = 0.05$, the corresponding p-value = 0.1098, which means any classifier that achieves precision higher than p-value should be considered better than random guess.

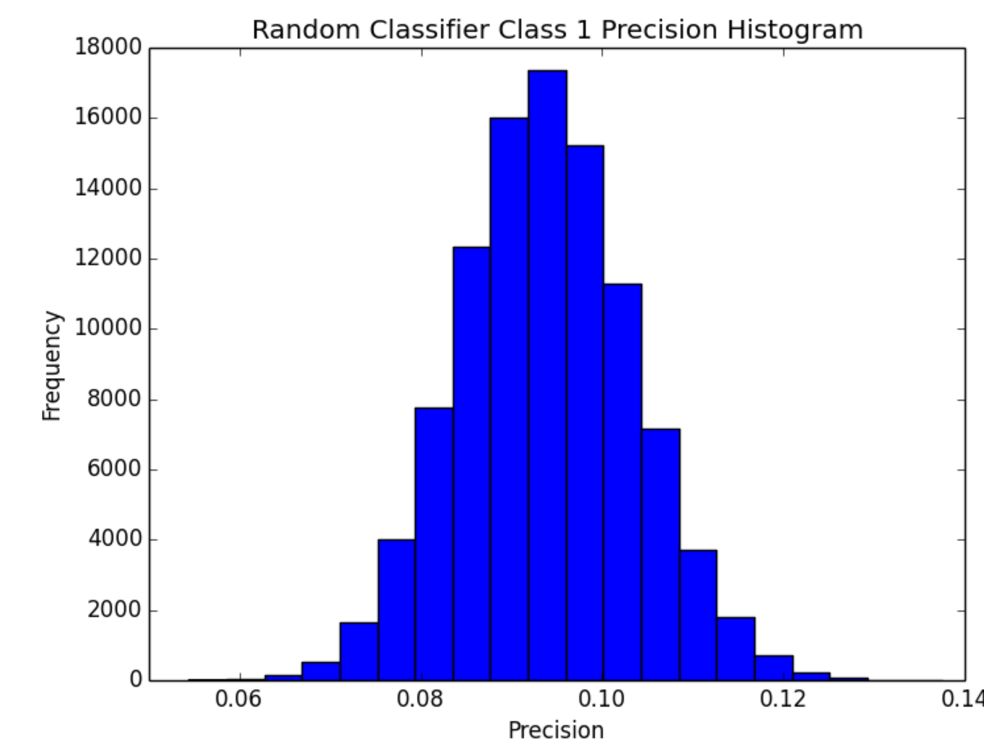

Figure2: Random Classifier's Popular Class Precision Distribution

Model	Precision	Recall
SVM-EN-2gram[1]	0.18	0.04
SVM-EN-2gram-tfidf-o[2]	0.16	0.10
Bernoulli-NB-tfidf[3]	0.16	0.20
Bonoulli-NB-tfidf-o[4]	0.16	0.18
PA-w2v(500)-mean[5]	0.15	0.13
SVM-w2v(200)-EN-tfidf-o[6]	0.14	0.32
SVM-w2v(1500)-tfidf-o[7]	0.14	0.30

1. SVM, Elastic Net Penalty, 2-gram, no resampling
2. SVM Elastic Net Penalty, 2-gram tf-idf features, oversampled
3. Bernoulli Naïve Bayes, tf-idf features, no resampling
4. Bernoulli Naïve Bayes, tf-idf features, oversampled
5. Passive aggressive, word2vec, vector size=500, take mean of vectors
6. SVM, word2vec, vector size=200, tf-idf features, oversampled
7. SVM, word2vec, vector size = 1500, tf-idf features, oversampled

Table2: Best Models We Get

Multinomial classification

Not better. Precision for each class is approximately the ratio of that class in our dataset.

Discussion

Despite the fact that predicting the popularity using news text is in general a difficult task[2] because of the noise of data, we are still able to find predictors that achieve significantly better precision than random predictors. To better understand our dataset, we also build topic visualization tools, which automatically generate meaningful topics. Our experiment has yielded very promising results on this part.

Future Work

We can obtain larger data set, incorporate more features such as title and publication time and further fine tune different classifiers to get better prediction results.

Reference

[1] C. Sievert and K. Shirley, "LDAvis: A method for visualizing and interpreting topics," Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, 2014.

[2] I. Arapakis, B. B. Cambazoglu, and M. Lalmas, "On the Feasibility of Predicting News Popularity at Cold Start," Lecture Notes in Computer Science Social Informatics, pp. 290–299, 2014.