Joint Sentiment Prediction via Review Text and Social Network Analysis

Xiao (Cosmo) Zhang, Bing Yu, Jiasen Yang, Ellen Lai

Departments of Computer Science and Statistics
Purdue University

December 11, 2014

Outline

1. Data Processing
2. Naive Bayes Classifier
3. Natural Language Processing
4. Social Network Analysis

PURDUE UNIVERSITY

Outline

1. Data Processing
2. Naive Bayes Classifier
3. Natural Language Processing
4. Social Network Analysis

Data Pre-processing

- Yelp Dataset Challenge
- Comprises 42,153 businesses, 320,002 business attributes, 31,617 check-in sets, 252,898 users, 403,210 tips, 1,125,458 reviews, as well as a social network consisting of 955,999 edges.
- Focus on businesses with enough ratings among Scottsdale, Temple, and Phoenix
- Sentiment: We would consider 4-star and 5-star ratings as positive, and 1-star and 2-star ratings as negative.

Data Processing

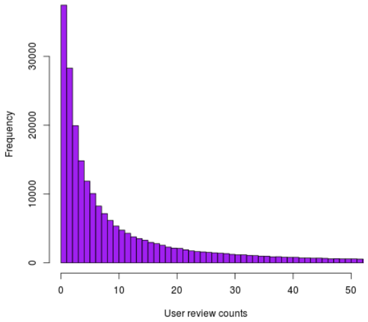


Figure : User review counts

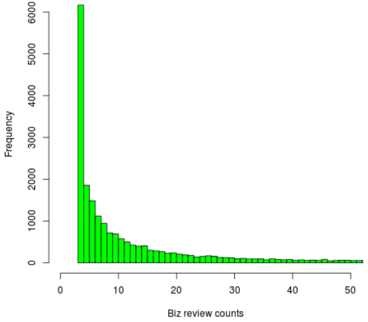


Figure : Business review counts

Outline

1 Data Processing

2 Naive Bayes Classifier

3 Natural Language Processing

4 Social Network Analysis

Naive Bayes Classifier — Intuition

- Goal: To predict the user's rating to a particular business individually
- Available information: user's ratings and reviews on other businesses, and business's ratings and reviews from other users
- Assumption: Independence among features given rating.

Naive Bayes Classifier — Probabilistic model

- Rating:$C = \{+, -\}$, User U, Business B, Review feature R.
- Basic model:

$$P(C|U, B) = \frac{P(C) \times P(U|C) \times P(B|C)}{P(U, B)}$$

	Accuracy	Recall	Precision	F-score
Test set	87.97%	97.77%	89.74%	93.59%

Table : Results for basic model

Naive Bayes Classifier—Probabilistic model

- Take into account common features of user's review on other businesses and business's review from other users
- Advanced model:

$$P(C|U,B,R) = \frac{P(C) \times P(U|C) \times P(B|C) \times P(R|C)}{P(U,B,R)}$$

Outline

1. Data Processing
2. Naive Bayes Classifier
3. Natural Language Processing
4. Social Network Analysis

PURDUE UNIVERSITY

Dependency Tree

The Stanford dependencies provide a representation of grammatical relations between words in a sentence. They have been designed to be easily understood and effectively used by people who want to extract textual relations.

- *NLTK package* for sentence tokenize for the entire review text in a (user-business) pair example.
- *Stanford CoreNLP package* with a self-modified python module for sentence parsing

Parsed Sentence Example 1

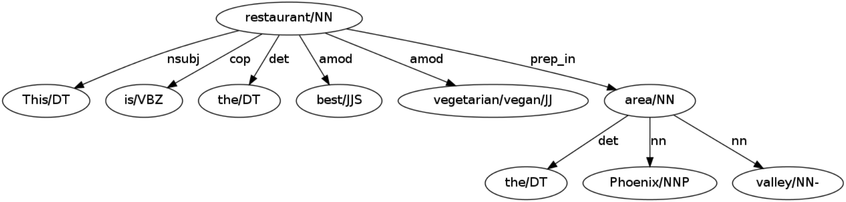


Figure : Parsed Sentence 1-positive

Parsed Sentence Example 2

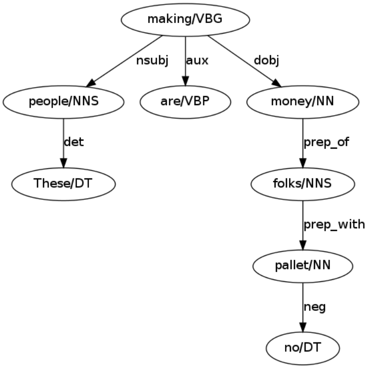


Figure : Parsed Sentence 2-negative

Dynamic Latent Conditional Random Field

Conditional random fields (CRFs) are a class of statistical modelling method often applied in pattern recognition and machine learning, where they are used for structured prediction.

- *Dynamic:* The structures are variable among sentences in training examples.
- *Latent (Hidden):* Some labels are observable, while some are unknown.

From Tree Graph to Factor Graph

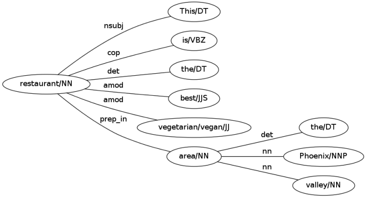


Figure : Graph of the parsed tree

⇒

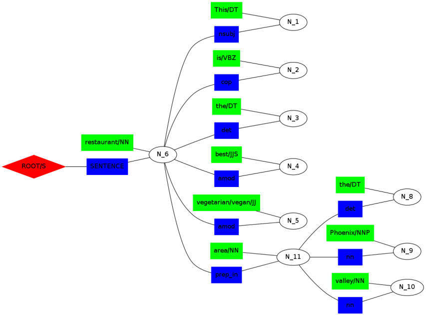


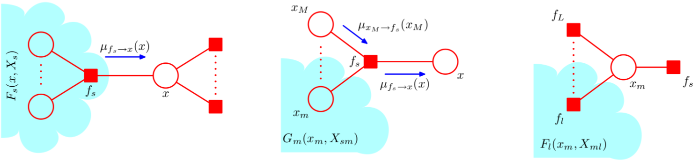
Figure : Converted to factor graph

Known and Unknown Labels

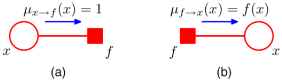
- $y_\diamond$ is the root label of the entire tree, which has a known label propagated from the rating.
- $w_\circ$s are the labels of nodes, which are the roots of subtrees, whose labels are unknown.

Belief Propagation Algorithm

Belief propagation, also known as sum-product message passing is a message passing algorithm for performing inference on graphical models. It calculates the marginal distribution for each unobserved node, conditional on any observed nodes.



(a) (b) (c)



(d)

Breadth First Search

Breadth first search (BFS) is a graph search algorithm that begins at the root node and explores all the neighboring nodes.

- Then we can sort the nodes in a *Topological Order* (Levels) from the root.
- Belief propagation first from root to all nodes.
- Second from leaves to the root in a reversed order.

Features

Node factors and edge factors have different features.

- *Node Factor*: $f(w_n)$, $f(w_n, q_n)$, $f(w_n, tag_n)$, $f(w_n, word_n)$, $f(w_n, q_n, r_n)$
- *Edge Factor*: $f(w_n, w_{n_h})$, $f(w_n, w_{n_h}, r_{n_h})$, $f(w_n, w_{n_h}, r_{n_h}, q_{n_h})$, $f(w_n, word_{n_h})$, $f(w_n, word_n)$

w is the label, *q* is the polarity of word, *tag* is the part of speech tag, *word* is the stemmed form of the word.

- *q* dictionary from Bin Liu.
- *r* dictionary as a combination of “NotLw" and “Decreas" from inquirer.
- *word* by NLTK snowball stemmer using English grammar

Maximum a Posteriori

- $$\text{objective function: } \mathcal{L}(\boldsymbol{\lambda}) = \log \sum_{\mathbf{w}} p(y, \mathbf{w}|\mathbf{x}) - \frac{1}{2\sigma^2} \sum_k \lambda_k^2$$
- $$\text{partial derivative: } \frac{\partial \mathcal{L}}{\partial \lambda_k} = \sum_{c \in F} \sum_{\mathbf{w}_c} p(\mathbf{w}_c|y, \mathbf{x}) f_k$$
  $$- \sum_{c \in F} \sum_{\mathbf{w}_c} p(\mathbf{w}_c, y|\mathbf{x}) f_k - \frac{1}{\sigma^2} \lambda_k$$

Stochastic BFGS Optimization

The *BFGS* method approximates Newton's method, a class of hill-climbing optimization techniques that seeks a stationary point of a (preferably twice continuously differentiable) function.

- We ran it in a stochastic fashion for each parameter, in each factor, in each sentence, and in each example.
- It converges very fast.

Majority Vote

- We used the same *belief propagation algorithm* for inference, to get the label of the root of the tree, representing the polarity of the whole sentence.
- Then we applied a simple *majority vote* on all the sentences in a review to get the predicated label of the entire review.

Final Result

The results were shown below after our algorithm ran 6 iterations on the training set.

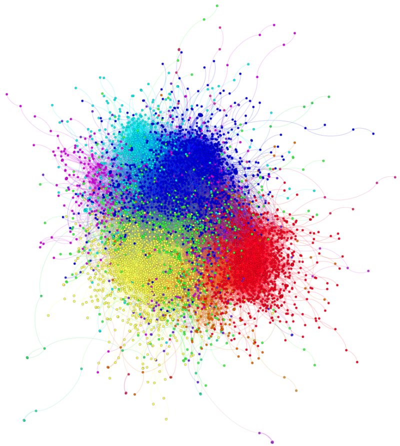
	Accuracy	Recall	Precision	F-score
Validation set	90.11%	100.00%	90.11%	94.80%
Test set	88.99%	100.00%	88.99%	94.17%

Table : Final Results

Outline

1. Data Processing
2. Naive Bayes Classifier
3. Natural Language Processing
4. Social Network Analysis

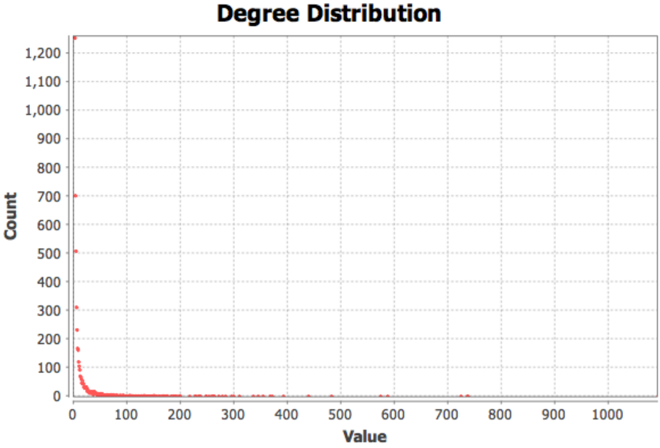
Network Visualization & Statistics



Network statistics:

- $|V| = 9186$, $|E| = 36737$
- Average degree: 14.877
- Average path length: 3.281
- Clustering coefficient: 0.342
- Diameter: 10
- Modularity: 0.396
- Number of communities: 92

Network Visualization & Statistics (Cont'd)



Rating/Sentiment Prediction

- Intuition: Users' rating behaviors are correlated with their friends' rating behaviors (*homophily* and *social influence*).
- Mathematically, user u_i's rating to business b_k is modeled by

$$R[u_i, b_k] \leftarrow \sum_{u_j \in \text{Friends}(u_i)} \frac{w(u_i, u_j)\, R[u_j, b_k]}{|\text{Friends}(u_i)|} \qquad (\star)$$

where Friends(u_i) may be replaced by Community(u_i), and

$$w(u_i, u_j) = \sum_{b_k \in \text{SharedBiz}(u_i, u_j)} \frac{\|R[u_i, b_k] - R[u_j, b_k]\|_{0,1}}{|\text{SharedBiz}(u_i, u_j)|}.$$

- Augment edges by exploiting observed rating information.
- Algorithm: Impute missing ratings by iteratively applying $(\star)$.

Questions?

http://cosmozhang.github.io/ml_final_project_yelp/