Project Proposal

Piyush Shinde
Saurabh Kumar
Shubhankar Mitra

*

SCHOOL OF INFORMATICS AND COMPUTING

INDIANA UNIVERSITY

Department of Information and Library Science
Bloomington

Overview

Task 1:
Restaurant Recommendations
- Experiment Design
- User similarity based recommendations:
- Item similarity based recommendations

TF-IDF
NMF
Doc2Vec
- Apache Spark ALS Matrix factorization
- Factorization Machine using tensorflow

Task 2 :
REVIEW IDENTIFICATION FOR USER SUPPLIED WORD USING LDA AND WORD2VEC
- LDA topic generation
- Experimentation to adjust parameters with reviews from some 200 restaurants by seeing texts for top 20 reviews obtained for a keyword
- Evaluation using categories column for user provided keyword from review text of 1200 restaurants

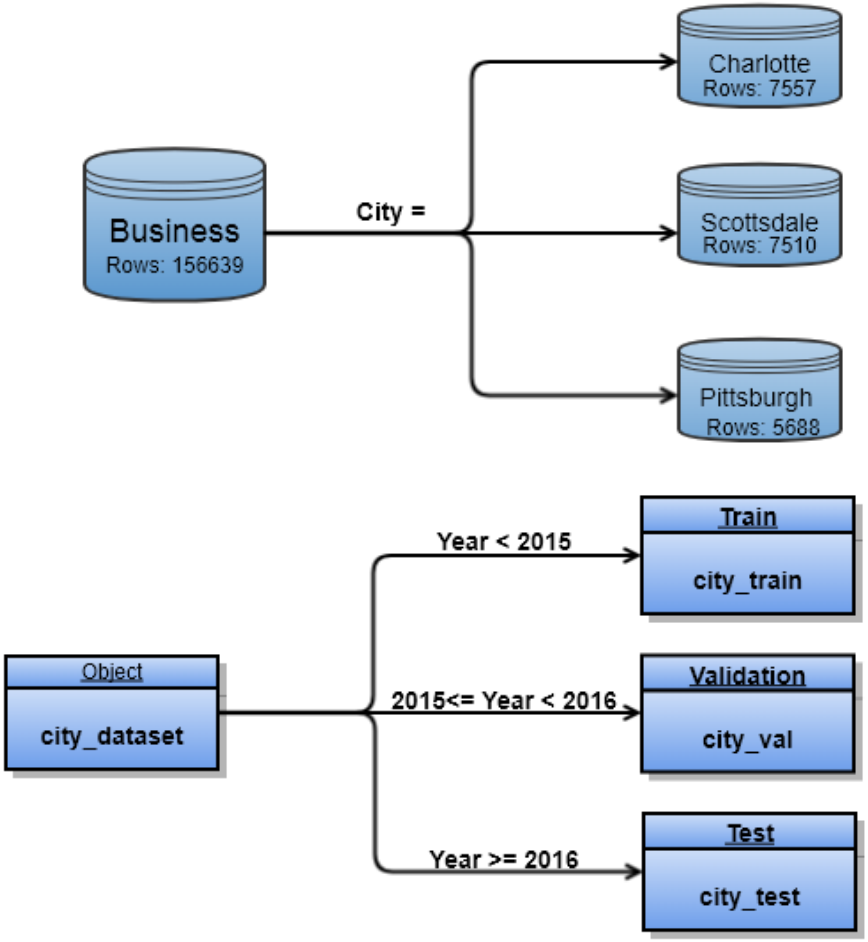
Task Allocation

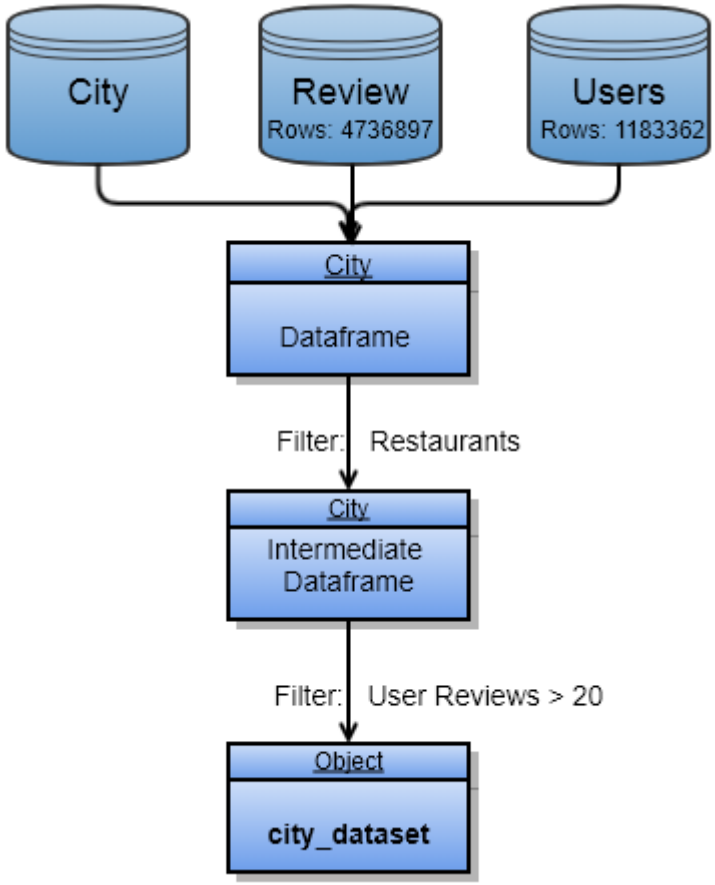
Piyush Shinde	Saurabh Kumar	Shubhankar Mitra
Training, evaluation, and test dataset creation (Data Manipulation code) for Task 1 according to experiment design specifications. Code – Complete data manipulation, complete user based filtering using NMF, Doc2Vec and TF-IDF Report – Data Generation, Model Creation, Content Based Recommendation, User Based Similarity, Evaluation (User Based)	Task 1: Item Similarity Based Recommendations using TF-IDF NMF Doc2Vec Code – Complete item based filtering using NMF, Doc2Vec, TF-IDF Report – Abstract, Introduction, block diagrams in Data Generation, Item Based Similarity, Collaborative Filtering, Evaluation (Item Based), Results, Conclusion and Future Enhancements	Code – Complete Code of Task 2 Ran code to convert Yelp Json data to csv Wrote function for finding user and item similarity (Function name in code: calc_sim_user_rating) in Task 1 Provided example code for Dec2Vec training, NMF for User similarity and Item similarity in Task 1 Wrote and ran code for collaborative filtering (Task1 - ALS Matrix Factorization) Report – Complete part of Task 2
Task 1: User Similarity Based Recommendations using TF-IDF NMF Doc2Vec	Additional: Executed Task 1 on Burrow.	Task 2 Complete algorithm design, writing code and evaluation.

Task 1: Experiment Design

- Merged datasets business, reviews and users.
- Extracted users with atleast 20 reviews.
- Cities – Pittsburgh, Scottsdale, Charlotte (high and similar number of restaurant ratings)
- Divided a city data set into training, validation and testing set year based.
- Time Based - <2015,>=2015 & <2016, >=2016
- Constraint – User_id and Business_id present in validation and training set should be present in training set.

Task 1: Experiment Design





Task 1: Feature Extraction Techniques

We used 3 feature extraction techniques:

1. Non-Negative Matrix Factorization (NMF)

class sklearn.decomposition.**NMF**(*n_components=None, init=None, solver='cd', beta_loss='frobenius', tol=0.0001, max_iter=200, random_state=None, alpha=0.0, l1_ratio=0.0, verbose=0, shuffle=False*)

2. Term Frequency–Inverse Document Frequency (TF-IDF)

class sklearn.feature_extraction.text.**TfidfVectorizer**(*input='content', encoding='utf-8', decode_error='strict', strip_accents=None, lowercase=True, preprocessor=None, tokenizer=None, analyzer='word', stop_words=None, token_pattern='(?u)\b\w\w+\b', ngram_range=(1, 1), max_df=1.0, min_df=1, max_features=None, vocabulary=None, binary=False, dtype=<class 'numpy.int64'>, norm='l2', use_idf=True, smooth_idf=True, sublinear_tf=False*)

Task 1: Feature Extraction Techniques

3. Doc2Vec

Bases: gensim.models.word2vec.Word2Vec

class gensim.models.word2vec.Word2Vec(*sentences=None, size=100, alpha=0.025, window=5, min_count=5, max_vocab_size=None, sample=0.001, seed=1, workers=3, min_alpha=0.0001, sg=0, hs=0, negative=5, cbow_mean=1, hashfxn=<built-in function hash>, iter=5, null_word=0, trim_rule=None, sorted_vocab=1, batch_words=10000, compute_loss=False*)

Task 1: Algorithms

1) User Based Similarity

Find average rating of Restaurant3 (R3) given by User1 (U1)

$$Avg.\ Rating\ of\ U_1 + \Sigma_{U\varepsilon(Users\ similar\ to\ U_1\ who\ rated\ R_3\ in\ tr.)} \frac{U_{R3}^{tr.} + U_{avg.}^{tr.}}{Number\ of\ Users}$$

2) Item Based Similarity

Find average rating of Restaurant3 (R3) given by User1 (U1)

$$Avg.\ Rating\ of\ R_3 + \Sigma_{R\varepsilon(Restaurants\ rated\ by\ U_1\ in\ tr.)} \frac{R_{U1}^{tr.} + R_{avg.}^{tr.}}{Number\ of\ Restaurants}$$

Similarity is found using **TF-IDF, NMF** and **Doc2Vec**

Task 1: Evaluation

Parameter Tuning was performed on the validation set on the parameters :

- *n_components* in NMF

class sklearn.decomposition.**NMF**(*n_components=None, init=None, solver='cd', beta_loss='frobenius' , tol=0.0001, max_iter=200, random_state=None, alpha=0.0, l1_ratio=0.0, verbose=0, shuffle=False*)

- *window* and *size* in Doc2Vec

Bases: gensim.models.word2vec.Word2Vec

class gensim.models.word2vec.Word2Vec(*sentences=None, size=100, alpha=0.025, window=5, min_ count=5, max_vocab_size=None, sample=0.001, seed=1, workers=3, min_alpha=0.0001, sg=0, hs=0, negative=5, cbow_mean=1, hashfxn=<built-in function hash>, iter=5, null_word=0, trim_rule=None, sorted_vocab=1, batch_words=10000, compute_loss=False*)

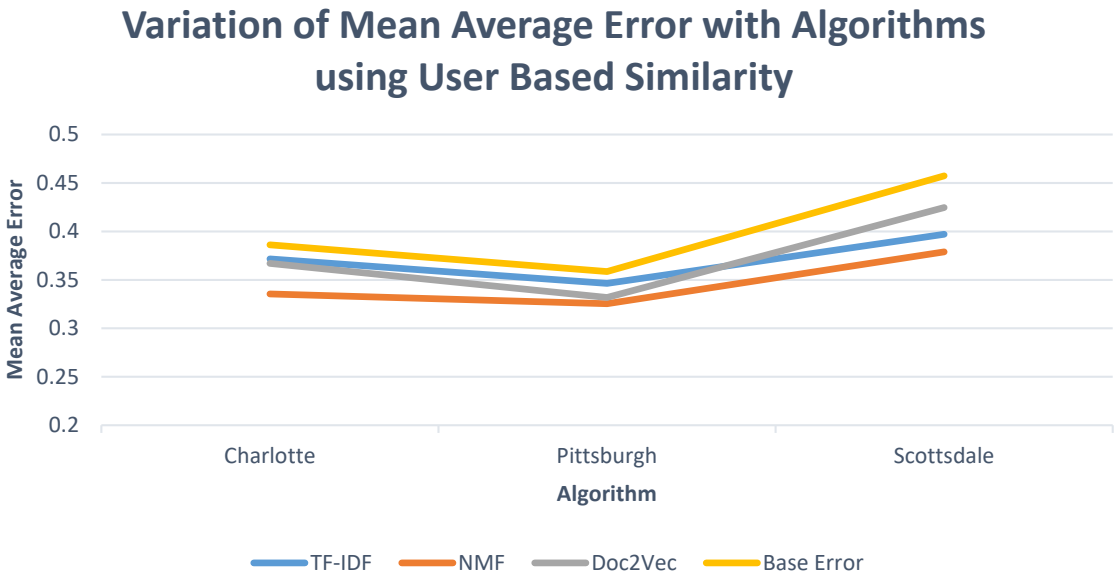
Task 1: Evaluation

1. The variance of mean square error was calculated :
   - by creating 10 samples (by bootstrap sampling)
   - on 80% of the test data (for each city)
   - resulting in 10 mean average errors (for each city)

2. We calculated mean of these errors to get the final mean error for each algorithm (Collaborative Filtering Algorithms (Item and User Based Similarity) & ALS Matrix factorization using Apache Spark (for each city)).

Each Collaborative Filtering Algorithm was performed using 3 feature extraction techniques (NMF, Doc2Vec, TF-IDF)

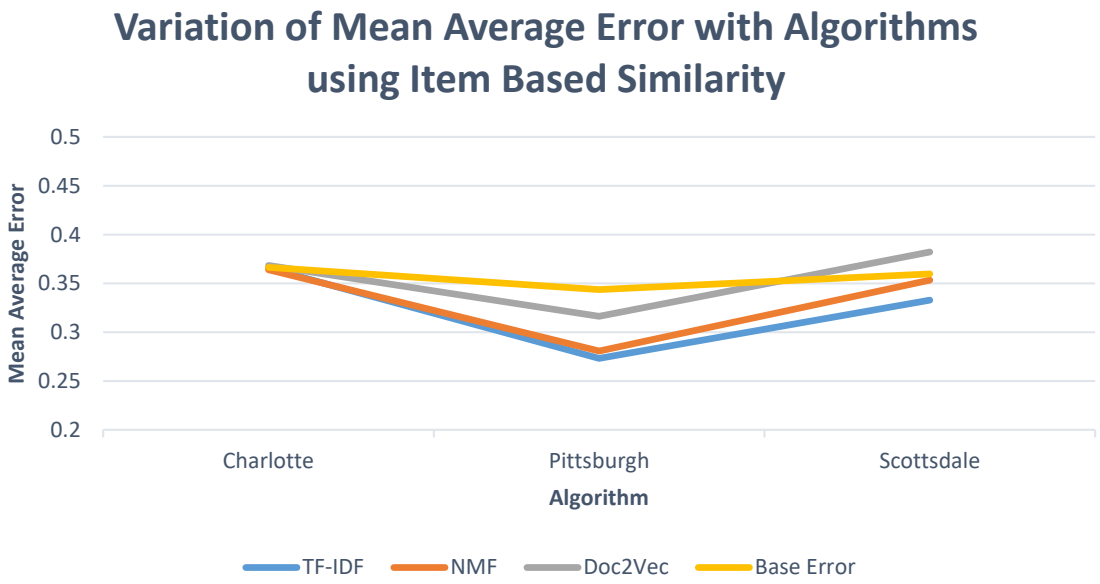
Task 1: Results for User Based Similarity



Least mean average square error – Non-Negative Matrix Factorization (NMF)

Highest mean average square error – Base Mean Average Error

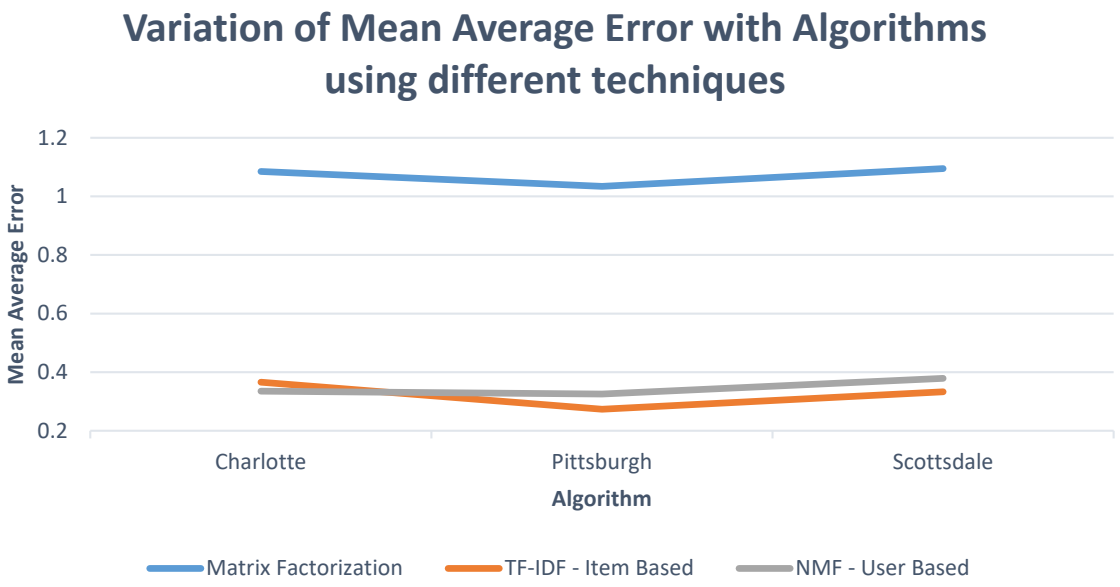
Task 1: Results for Item Based Similarity



Least mean average square error – Term Frequency–Inverse Document Frequency (TF-IDF)

Highest mean average square error – Base Mean Average Error

Task 1: Results for Algorithms using different techniques



Least mean average square error – Term Frequency–Inverse Document Frequency (TF-IDF) for Item Based Similarity

Highest mean average square error – Base Mean Average Error

Task 2: Proposed task and significance

- REVIEW IDENTIFICATION FOR USER SUPPLIED WORD USING LDA AND WORD2VEC

- Significance:
  - —Review topic tagging
  - —Quicker understanding of LDA topics
  - —Review retrieval for user supplied word

Task 2: Experiment Design

- Completely unsupervised task
- Pittsburg restaurant reviews
- Minor parameter tweaking (LDA topic count, number of topics assigned to review) using results from review text of an initial set of 200 restaurants.
- Evaluation with categories from review text of 1200 restaurants
- Evaluation metrics: accuracy, precision, area under ROC curve, and recall

Task 2: Algorithm details

Preprocessing

- Lower case all reviews, lemmatize review text using Wordnet lemmatizer, and remove punctuation from text reviews. Create bag of words of review texts after removing stop words.
- Extract LDA topic word vector and review text LDA topic vector.
- Train word2vec vector on review text

Review Tagging

- Calculate word2vec vector of each LDA topic by calculating the weighted average of word2vec vector for each topic with weights being the value of word importance for the topic.
- Calculate cosine similarity between word2vec representation of user supplied word and word2vec vector of each topic.
- Take cosine similarity between vector of similar topics and topic distribution of reviews to extract top relevant reviews for the user supplied word.

Task 2: Evaluation and Sample results

		pizza	italian	beer	sweet	show	entertainment	Metric Average
LDA	Accuracy	85%	85%	59%	90%	26%	21%	61%
LDA	Recall	3%	2%	0%	8%	81%	85%	30%
LDA	Precision	97%	95%	19%	8%	5%	5%	38%
LDA	Area under ROC curve	79%	79%	46%	56%	56%	56%	62%
LDA with word2vec	Accuracy	85%	87%	41%	79%	8%	9%	51%
LDA with word2vec	Recall	4%	14%	95%	20%	96%	95%	54%
LDA with word2vec	Precision	98%	95%	40%	6%	5%	5%	42%
LDA with word2vec	Area under ROC curve	81%	79%	55%	51%	56%	56%	63%

Task 2: Limitations and Future work

- Using word2vec with LDA shows some theoretical advantages of being able to use use contextual similarity of words from word2vec but this is reflected in our evaluation results when compared to just using LDA. For further research, we may need to use a more accurate proxy for evaluation and optimize parameters for word2vec and LDA.

Thank you