Project Kindle

Devoja Ganguli

Nazila Shafiei

Suji Yang

May 6, 2019

This project

- Sentiment analysis
  - Does the review says Good or Bad?
- Data: Amazon Kindle e-book reviews
  - http://jmacauley.ucsd.edu/data/amazon/index.html
- Techniques: Python library sci-kit learn for supervised machine learning

Procedure

1. Import all desired packages
2. Import Amazon Reviews data set using gzip and parse functions
3. Data Cleaning
4. Feature Scaling (Binary, Multi-label) of data
5. Tokenizing and Splitting the data
6. Word Embedding (Google pretrained word2vec model)
7. Convert word embedding into sentence embedding
8. Machine learning
9. Make predictions on new data with the saved models

Procedure

1. Import all desired packages
2. Import Amazon Reviews data set using gzip and parse functions
3. Data Cleaning
4. Feature Scaling (Binary, Multi-label) of data
5. Tokenizing and Splitting the data
6. Word Embedding (Google pretrained word2vec model)
7. Convert word embedding into sentence embedding
8. Machine learning
9. Make predictions on new data with the saved models

Amazon product data

Julian McAuley, UCSD

Files

"Small" subsets for experimentation

If you're using this data for a class project (or similar) ***please*** consider using one of these smaller datasets below before requesting the larger files. To obtain the larger files you will need to contact me to obtain access.

K-cores (i.e., dense subsets): These data have been reduced to extract the k-core, such that each of the remaining users and items have k reviews each.

Ratings only: These datasets include no metadata or reviews, but only (user,item,rating,timestamp) tuples. Thus they are suitable for use with mymedialite (or similar) packages.

Books	5-core (8,898,041 reviews)	ratings only (22,507,155 ratings)
Electronics	5-core (1,689,188 reviews)	ratings only (7,824,482 ratings)
Movies and TV	5-core (1,697,533 reviews)	ratings only (4,607,047 ratings)
CDs and Vinyl	5-core (1,097,592 reviews)	ratings only (3,749,004 ratings)
Clothing, Shoes and Jewelry	5-core (278,677 reviews)	ratings only (5,748,920 ratings)
Home and Kitchen	5-core (551,682 reviews)	ratings only (4,253,926 ratings)
Kindle Store	5-core (982,619 reviews)	ratings only (3,205,467 ratings)
Sports and Outdoors	5-core (296,337 reviews)	ratings only (3,268,695 ratings)
Cell Phones and Accessories	5-core (194,439 reviews)	ratings only (3,447,249 ratings)
Health and Personal Care	5-core (346,355 reviews)	ratings only (2,982,326 ratings)
Toys and Games	5-core (167,597 reviews)	ratings only (2,252,771 ratings)
Video Games	5-core (231,780 reviews)	ratings only (1,324,753 ratings)
Tools and Home Improvement	5-core (134,476 reviews)	ratings only (1,926,047 ratings)
Beauty	5-core (198,502 reviews)	ratings only (2,023,070 ratings)
Apps for Android	5-core (752,937 reviews)	ratings only (2,638,172 ratings)
Office Products	5-core (53,258 reviews)	ratings only (1,243,186 ratings)
Pet Supplies	5-core (157,836 reviews)	ratings only (1,235,316 ratings)
Automotive	5-core (20,473 reviews)	ratings only (1,373,768 ratings)
Grocery and Gourmet Food	5-core (151,254 reviews)	ratings only (1,297,156 ratings)
Patio, Lawn and Garden	5-core (13,272 reviews)	ratings only (993,490 ratings)
Baby	5-core (160,792 reviews)	ratings only (915,446 ratings)
Digital Music	5-core (64,706 reviews)	ratings only (836,006 ratings)
Musical Instruments	5-core (10,261 reviews)	ratings only (500,176 ratings)
Amazon Instant Video	5-core (37,126 reviews)	ratings only (583,933 ratings)

http://jmcauley.ucsd.edu/data/amazon/index.html

Procedure

1. Import all desired packages
2. Import Amazon Reviews data set using gzip and parse functions
3. Data Cleaning
4. Feature Scaling (Binary, Multi-label) of data
5. Tokenizing and Splitting the data
6. Word Embedding (Google pretrained word2vec model)
7. Convert word embedding into sentence embedding
8. Machine learning
9. Make predictions on new data with the saved models

Procedure

1. Import all desired packages
2. Import Amazon Reviews data set using gzip and parse functions
3. Data Cleaning
4. Feature Scaling (Binary, Multi-label) of data
5. Tokenizing and Splitting the data
6. Word Embedding (Google pretrained word2vec model)
7. Convert word embedding into sentence embedding
8. Machine learning
9. Make predictions on new data with the saved models

Feature Scaling

- Ratings of 1.0 to 5.0
- Binary classification {0: negative, 1: positive}

asin	reviewText	overall	1.0	2.0	3.0	4.0	5.0	score
B000FA64PK	Another well written eBook by Troy Denning, bu...	3.0	0	0	1	0	0	0
B000FA64PK	This one promises to be another good book. I h...	5.0	0	0	0	0	1	1
B000FA64PK	I have a version of "Star by Star" that does n...	4.0	0	0	0	1	0	1
B000FA64PK	Excellent! Very well written story, very excit...	5.0	0	0	0	0	1	1
B000FA64QO	With Ylesia, a novella originally published in...	2.0	0	1	0	0	0	0

Procedure

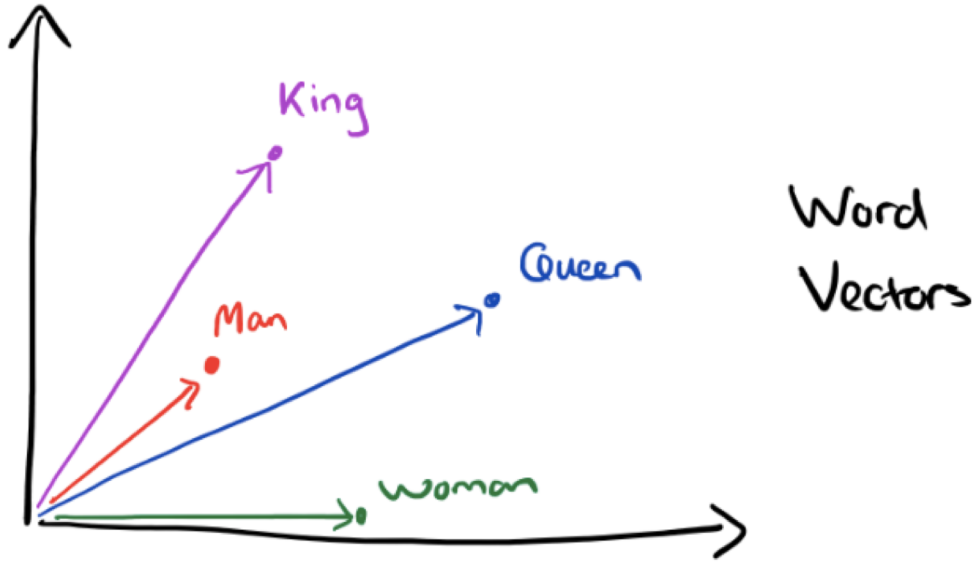
1. Import all desired packages
2. Import Amazon Reviews data set using gzip and parse functions
3. Data Cleaning
4. Feature Scaling (Binary, Multi-label) of data
5. Tokenizing and Splitting the data
6. Word Embedding (Google pretrained word2vec model)
7. Convert word embedding into sentence embedding
8. Machine learning
9. Make predictions on new data with the saved models

Procedure

1. Import all desired packages
2. Import Amazon Reviews data set using gzip and parse functions
3. Data Cleaning
4. Feature Scaling (Binary, Multi-label) of data
5. Tokenizing and Splitting the data
6. Word Embedding (Google pretrained word2vec model)
7. Convert word embedding into sentence embedding
8. Machine learning
9. Make predictions on new data with the saved models

Google Pretrained Word2Vec Model

- Word2Vec: array[3.005, -2.9867676, … 10.6] (size: 300)



https://blog.acolver.org/word2vec-king-queen-vectors/

Word to Sentence Embedding

```
1  from sklearn.preprocessing import normalize
2  semd=[]
3  for item in tokenized:
4      wemd=np.zeros(300)
5      for w in item:
6          if w in model:
7                  wemd = np.add(model[w],wemd)
8                  normed = wemd/np.linalg.norm(wemd)
9      semd.append(normed)
10 print(semd)
```

```
[array([ 0.061966  ,  0.03309556,  0.04097159,  0.10301728, -0.04806158,
       -0.01386412,  0.00504604, -0.04783289,  0.05137734,  0.08960725,
        0.00227883, -0.19542821, -0.01835176,  0.01710365, -0.10516773,
        0.06007068,  0.02582216,  0.09913436,  0.04086199, -0.07349749,
       -0.00431695,  0.06142346, -0.00561794, -0.03626602,  0.04766147,
       -0.03425278, -0.09543819,  0.06323872,  0.04431689, -0.00514124,
       -0.04096172,  0.00442071, -0.03436779,  0.0049927 ,  0.0187515 ,
        0.03075933, -0.00121615, -0.00484289,  0.01995584,  0.1078629 ,
```

Procedure

1. Import all desired packages
2. Import Amazon Reviews data set using gzip and parse functions
3. Data Cleaning
4. Feature Scaling (Binary, Multi-label) of data
5. Tokenizing and Splitting the data
6. Word Embedding (Google pretrained word2vec model)
7. Convert word embedding into sentence embedding
8. Machine learning
9. Make predictions on new data with the saved models

Machine Learning

Model	Precision	Recall	f1-score	Accuracy
Logistic Regression	0.83	0.77	0.68	0.77
SVM	0.59	0.77	0.67	0.77
Decision Trees	0.69	0.69	0.69	0.69
Random Forest	0.74	0.77	0.74	0.77

Figure 1: Model Results for Binary Classification

Model	Precision	Recall	f1-score	Accuracy
Logistic Regression	0.37	0.51	0.37	0.51
SVM	0.26	0.51	0.34	0.51
Decision Trees	0.51	0.47	0.48	0.47
Random Forest	0.42	0.50	0.45	0.50

Figure 2: Model Results for Multi-label Classification

Procedure

1. Import all desired packages
2. Import Amazon Reviews data set using gzip and parse functions
3. Data Cleaning
4. Feature Scaling (Binary, Multi-label) of data
5. Tokenizing and Splitting the data
6. Word Embedding (Google pretrained word2vec model)
7. Convert word embedding into sentence embedding
8. Machine learning
9. Make predictions on new data with the saved models

Making Prediction with models

```
Xnew=test_new_data("I love this book. It is so inspiring! \
I would definately recommend buying it")
```

```
ynew = classifier_lr_load.predict(Xnew)
print(ynew)
```

```
[5.]
```

Conclusion

a) We did not estimate the results multiple times using cross-validation on our data
b) Having limited computational power prevented us from running the model on the whole dataset
c) Last but not least, we plotted different training and testing sizes and compared the accuracy of classifiers

References

- Bengfort, Benjamin, Rebecca Bilbro, and Tony Ojeda. 2018. Applied Text Analysis with Python: Enabling Language-aware Data Products with Machi ne Learning. O'Reilly Media, Inc.
- He, Ruining, and Julian McAuley. 2016. Ups and downs: Modeling the visua l evolution of fashion trends with one-class collaborative filtering. In proce edings of the 25th international conference on world wide web, pp. 507-51 7. International World Wide Web Conferences Steering Committee
- Liu, Bing. 2012. Sentiment analysis and opinion mining. Synthesis lectures o n human language technologies 5 (1) : 1-167.
- McAuley, Julian. Amazon product data.