Foodie Impression

Yelp Review Analysis With Unsupervised Learning and Distributed Computing

Guoqiang Liang
Fang Wang
Bingyi Li
Xiaowen Zhang

Introduction

- Analytics Goal
- Data Usage
- Model Approach

Data Processing

- AWS S3
- MongoDB
- Data Frame Creation
- SparkSQL
- Machine Learning

Result Demonstration

Lessons Learned

Introduction

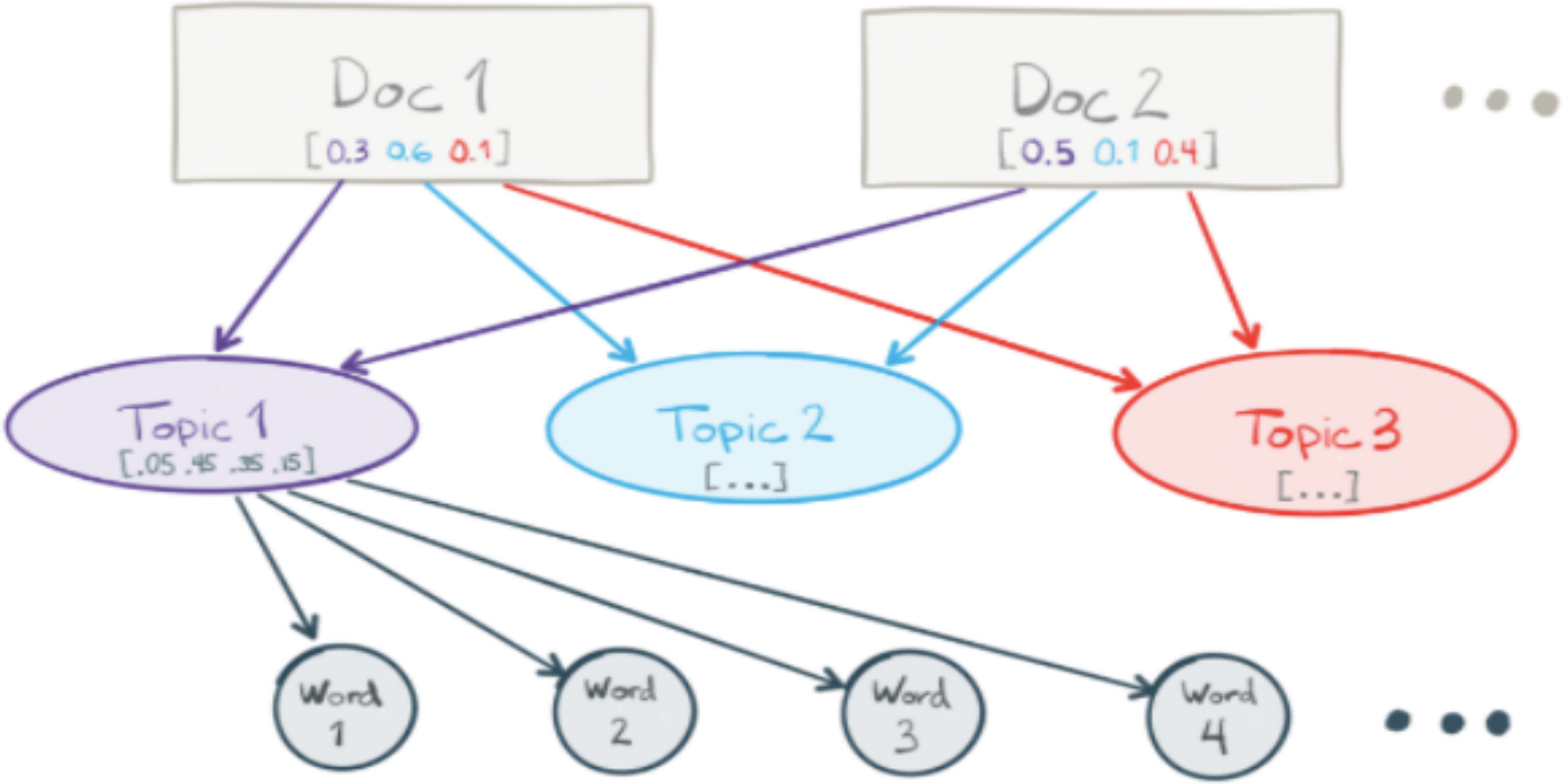
Goal - Find the most popular food and activities for various cities

Data - Yelp Open Dataset (business.json & review.json)

Model – Latent Dirichlet Allocation (LDA)

Methodology

Latent Dirichlet Allocation (LDA)



Data Processing

AWS S3

Amazon S3 > foodie-impression

Overview | Properties | Permissions (Public) | Management

Type a prefix and press Enter to search. Press ESC to clear.

Upload | + Create folder | More | US West (Oregon)

Viewing 1 to 2

Name	Last modified	Size	Storage class
business.json	Jan 14, 2018 11:07:57 PM GMT-0800	126.1 MB	Standard
review.json	Jan 14, 2018 9:31:55 PM GMT-0800	3.6 GB	Standard

Data Processing

MongoDB – Import Data

Review.json

```
2018-01-16T02:04:28.256+0000    [########################.] yelp.review  3.44GB/3.56GB
(96.8%)
2018-01-16T02:04:31.253+0000    [########################.] yelp.review  3.50GB/3.56GB
(98.5%)
2018-01-16T02:04:34.251+0000    [########################.] yelp.review  3.52GB/3.56GB
(98.9%)
2018-01-16T02:04:35.948+0000    [#########################] yelp.review  3.56GB/3.56GB
(100.0%)
2018-01-16T02:04:35.948+0000    imported 4736897 documents
[ec2-user@ip-172-31-30-254 ~]$
```

Business.json

```
[ec2-user@ip-172-31-30-254 ~]$ aws s3 cp s3://foodie-impression/business.json
. | mongoimport --db yelp --collection business business.json
2018-01-16T01:59:04.948+0000    connected to: localhost
2018-01-16T01:59:07.947+0000    [######..................] yelp.business
        36.2MB/126MB (28.7%)
2018-01-16T01:59:10.948+0000    [##################......] yelp.business
        92.3MB/126MB (73.2%)
2018-01-16T01:59:12.772+0000    [#########################] yelp.business
        126MB/126MB (100.0%)
2018-01-16T01:59:12.788+0000    imported 156639 documents
[ec2-user@ip-172-31-30-254 ~]$
```

Data Processing

MongoDB – Query Data

```
[ec2-user@ip-172-31-30-254 ~]$ mongo
MongoDB shell version: 3.2.18
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
> db
test
> use yelp
switched to db yelp
> db.review.findOne()
{
        "_id" : ObjectId("5a5d5d0a2d09f5bb172f815e"),
        "review_id" : "3zRpneRKDsOPq92tq7ybAA",
        "user_id" : "bjTcT8Ty4cJZhEOEo01FGA",
        "business_id" : "uYHaNptLzDLoV_JZ_MuzUA",
        "stars" : 3,
        "date" : "2016-10-02",
        "text" : "If you need an inexpensive place to stay for a night or two then you
 may consider this place but for a longer stay I'd recommend somewhere with better ame
nities. \n\nPros:\nGreat location- you're right by the train station, central location
 to get to old town and new town, and right by sight seeing his tours. Food, bars, and
 shopping all within walking distance. Location, location, location.\nVery clean and v
ery good maid service\n\nCons:\nTiny rooms \nUncomfortable bed \nAbsolutely no ameniti
es \nNo phone in room \nNo wardrobe \n\nWas given a lot of attitude about me and my hu
sband sharing a room which was quite strange and we were charged 15 pounds more for do
uble occupancy not sure why that matters I felt like it was a money grab. It was just
handled in a kind of odd manner to me... \n\nIf you book this hotel all you get is a b
ed, desk, and a bathroom. It isn't awful but know what you're getting into.",
        "useful" : 0,
        "funny" : 0,
        "cool" : 0
}
```

Data Processing

EMR – One Master and Two Salves

Clone | Terminate | AWS CLI export

Cluster: yelp3 Waiting Cluster ready after last step completed.

Summary | Application history | Monitoring | Hardware | Events | Steps | Configurations | Bootstrap actions

Connections: Enable Web Connection – Zeppelin, Spark History Server, Ganglia, Resource Manager … (View All)
Master public DNS: ec2-34-217-16-85.us-west-2.compute.amazonaws.com SSH
Tags: -- View All / Edit

Summary
ID: j-2BIXESPFRO5CO
Creation date: 2018-01-18 16:23 (UTC-8)
Elapsed time: 1 hour, 46 minutes
Auto-terminate: No
Termination protection: Off Change

Configuration details
Release label: emr-5.11.0
Hadoop distribution: Amazon 2.7.3
Applications: Ganglia 3.7.2, Spark 2.2.1, Zeppelin 0.7.3
Log URI: --
EMRFS consistent view: Disabled
Custom AMI ID: --

Network and hardware
Availability zone: us-west-2a
Subnet ID: subnet-d949f2bf
Master: Running 1 m3.xlarge
Core: Running 2 m3.xlarge
Task: --

EC2 Instances

Launch Instance | Connect | Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
MongoDB	i-028f013f2e043d9f1	t2.large	us-west-2a	running	2/2 checks …	None	ec2-54-186-189-193.us…
yelp-master	i-01d78aae5f57af5e0	m3.xlarge	us-west-2a	running	2/2 checks …	None	ec2-34-217-16-85.us-w…
yelp-slave-1	i-009134fa2226ab7d2	m3.xlarge	us-west-2a	running	2/2 checks …	None	ec2-54-201-108-245.us…
yelp-slave-2	i-0a2a09760dac96243	m3.xlarge	us-west-2a	running	2/2 checks …	None	ec2-54-190-51-239.us-…

Data Processing

Data Frame Creation

Review.json

In [4]:
```
format_string = "com.mongodb.spark.sql.DefaultSource"

business = spark.read.format(format_string).option("uri","mongodb://{}/yelp.business".format(ip)).load()
review = spark.read.format(format_string).option("uri","mongodb://{}/yelp.review".format(ip)).load()

print "ip = %s" % ip
review.printSchema()
```

```
ip = 54.186.189.193
root
 |-- _id: struct (nullable = true)
 |    |-- oid: string (nullable = true)
 |-- business_id: string (nullable = true)
 |-- cool: integer (nullable = true)
 |-- date: string (nullable = true)
 |-- funny: integer (nullable = true)
 |-- review_id: string (nullable = true)
 |-- stars: integer (nullable = true)
 |-- text: string (nullable = true)
 |-- useful: integer (nullable = true)
 |-- user_id: string (nullable = true)
```

Data Processing

SparkSQL

Query Data

```
In [4]: business.groupby('city').count().orderBy('count', ascending=False).show(10)
```

```
+----------+-----+
|      city|count|
+----------+-----+
| Las Vegas|24768|
|   Phoenix|15656|
|   Toronto|15483|
| Charlotte| 7557|
|Scottsdale| 7510|
|Pittsburgh| 5688|
|  Montréal| 5175|
|      Mesa| 5146|
| Henderson| 4130|
|     Tempe| 3949|
+----------+-----+
only showing top 10 rows
```

Data Processing

Machine Learning

Count Vectorizer : Mesa, AZ

In [27]:
```
# list_to_vector_udf = udf(lambda l: Vectors.dense(l), VectorUDT())
word_occurence = train_raw.map(lambda x: [tokenize(x)])
word_occurence = word_occurence.toDF()

cv = CountVectorizer(inputCol='_1', outputCol='features', vocabSize=2000)
cv_model = cv.fit(word_occurence)
train = cv_model.transform(word_occurence).cache()

train.show(5)
```

```
+--------------------+--------------------+
|                  _1|            features|
+--------------------+--------------------+
|[star, jennifer, ...|(2000,[2,7,17,67,...|
|[nope, promise, h...|(2000,[75,98,162,...|
|[don, mar, review...|(2000,[2,4,7,14,3...|
|[haircut, experie...|(2000,[0,1,3,8,9,...|
|[absolute, haircu...|(2000,[75,162,217...|
+--------------------+--------------------+
only showing top 5 rows
```

Data Processing

Machine Learning

LDA Application - Mesa, AZ

In [28]:
```
lda_model = LDA(k=10, maxIter=10).fit(train)
```

In [29]:
```
def indices_to_terms(vocabulary):
    def indices_to_terms(xs):
        return [vocabulary[int(x)] for x in xs]
    return udf(indices_to_terms, ArrayType(StringType()))

# Describe topics.
topics = lda_model.describeTopics(8)
topics.withColumn("topics_words", \
                  indices_to_terms(cv_model.vocabulary)("termIndices")).select('topics_words').show(truncate=False)
```

```
+------------------------------------------------------------+
|topics_words                                                |
+------------------------------------------------------------+
|[love, pho, bread, items, store, meat, menu, market]        |
|[chicken, sauce, night, soup, don, rice, pork, spicy]       |
|[work, experience, shop, massage, water, phone, air, family]|
|[car, manager, don, pizza, business, call, didn, phone]     |
|[work, job, experience, house, problem, guys, business, repair]|
|[hair, salon, nails, job, cut, rolls, color, car]           |
|[breakfast, lunch, fries, pizza, burger, chicken, taste, bar]|
|[room, care, office, guys, don, appointment, didn, business]|
|[restaurant, menu, bar, didn, eat, drinks, family, chips]   |
|[store, kids, thai, selection, items, shop, job, tea]       |
+------------------------------------------------------------+
```

Result Demonstration

City: Montreal

cluster	word1	word2	word3	word4	…
1	burger	wings	fries	bien	…
2	brunch	egg	breakfast	toast	…
3	bagel	cheese	crepe	cream	…
4	beer	bar	night	music	…
5	gras	foie	duck	quality	…
6	room	hotel	montreal	bathroom	…
7	pas	des	les	tres	…
…	…	…	…	…	…

Result Demonstration

City: Mesa

cluster	word1	word2	word3	word4	…
1	hair	salon	color	cut	…
2	thai	massage	hotel	son	…
3	pizza	wings	crust	pie	…
4	fries	burger	lunch	tacos	…
5	chicken	soup	rice	thai	…
6	chicken	sauce	restaurant	meat	…
7	car	guys	vehicle	problem	…
…	…	…	…	…	…

Result Demonstration

City: Toronto

cluster	word1	word2	word3	word4	…
1	store	sandwich	lunch	mall	…
2	restaurant	steak	bill	experience	…
3	sushi	roll	salmon	menu	…
4	cream	ice	chocolate	gelato	…
5	tea	coffee	milk	latte	…
6	ramen	wings	pork	noodles	…
7	bar	friends	drinks	beer	…
…	…	…	…	…	…

Result Demonstration

City: Las Vegas

cluster	word1	word2	word3	word4	…
1	buffet	quality	crab	restaurant	…
2	casino	strip	pool	hotel	…
3	steak	experience	lobster	filet	…
4	fries	burger	sandwich	coffee	…
5	strip	casino	night	beer	…
6	music	drinks	dance	girls	…
7	menu	wine	stars	pasta	…
…	…	…	…	…	…

Lessons Learned

- Similarity
  - most common words are the same for almost all the regions.
- Infrastructure
  - Python 2.6.9 does not support nltk package.
  - Default configuration leads to memory issue.
  - The version of mongodb connector and spark on ec2 does not match.

Bibliography

- LDA Topic Modeling in Spark Mllib, Zero Gravity Lab, Jul 14, https://zerogravitylabs.ca/lda-topic-modeling-spark-mllib/
- Yelp Open Dateset, 2004–2018 Yelp Inc. Yelp, https://www.yelp.com/dataset
- Stopwords.txt, Gerard Salton and Chris Buckley, The experimental SMART information retrieval system, Cornell University

Thank You !
Q & A