Project: Amazon sales meta data analysis

Enhong Ma course: 3030

Contents

[Objective 1](#_Toc425775224)

[Data source 1](#_Toc425775225)

[Infrastructure 1](#_Toc425775226)

[GREMLIN code 2](#_Toc425775227)

[Display topic related node and edge 5](#_Toc425775228)

[Conclusion 6](#_Toc425775229)

Objective**: define schema in graph DB, analyze meta data, present information visually for personalize marketing.**

Data source**:** [**http://snap.stanford.edu/data/#amazon**](http://snap.stanford.edu/data/#amazon)

**Here is the data format:**

* **Id:** Product id (number 0, ..., 548551)
* **ASIN:** [Amazon Standard Identification Number](http://en.wikipedia.org/wiki/Amazon_Standard_Identification_Number)
* **title:** Name/title of the product
* **group:** Product group (Book, DVD, Video or Music)
* **salesrank:** Amazon [Salesrank](http://www.amazon.com/gp/help/customer/display.html?nodeId=525376)
* **similar:** ASINs of co-purchased products (people who buy X also buy Y)
* **categories:** Location in product category hierarchy to which the product belongs (separated by |, category id in [])
* **reviews:** Product review information: time, user id, rating, total number of votes on the review, total number of helpfulness votes (how many people found the review to be helpful)

# Infrastructure

gremlin

**API layer**

Titan DB

**GraphDB layer**

**Storage layer**

Cassandra cluster ring

# GREMLIN code

1. **create schema**

gremlin> graph.makeKey('type').dataType(String.class).make()

==>type

gremlin> graph.makeKey('productno').dataType(String.class).indexed(Vertex.class).unique().make()

==>productno

gremlin> graph.makeKey('title').dataType(String.class).indexed('search', Vertex.class).make()

==>title

gremlin> graph.makeKey('salesrank').dataType(Integer.class).make()

==>salesrank

gremlin> graph.makeKey('group').dataType(String.class).make()

==>group

gremlin> graph.makeKey('status').dataType(String.class).make()

==>status

gremlin> graph.makeKey('reviews').dataType(Integer.class).make()

==>reviews

gremlin> graph.makeKey('avgrating').dataType(Float.class).make()

==>avgrating

gremlin> graph.makeKey('customerno').dataType(String.class).indexed(Vertex.class).unique().make()

==>customerno

gremlin> graph.makeKey('categoryno').dataType(Integer.class).indexed(Vertex.class).unique().make()

==>categoryno

gremlin> graph.makeKey('categoryname').dataType(String.class).indexed(Vertex.class).make()

==>categoryname

gremlin> date = graph.makeKey('date').dataType(Long.class).make()

==>date

gremlin> votes = graph.makeKey('votes').dataType(Integer.class).make()

==>votes

gremlin> rating = graph.makeKey('rating').dataType(Integer.class).make()

==>rating

gremlin> helpful = graph.makeKey('helpful').dataType(Integer.class).make()

==>helpful

gremlin> specificity = graph.makeKey('specificity').dataType(Float.class).make()

==>specificity

gremlin> graph.makeLabel('similar').make()

==>similar

gremlin> graph.makeLabel('review').sortKey(date).sortOrder(Order.DESC).signature(votes, rating, helpful).make()

==>review

gremlin> graph.makeLabel('parent').manyToOne().make()

==>parent

gremlin> graph.makeLabel('categorization').sortKey(specificity).make()

==>categorization

gremlin> reader = new GraphMLReader(graph)

==>com.tinkerpop.blueprints.util.io.graphml.GraphMLReader@dccf8c

gremlin> reader.inputGraph('../../productdb/productdb.graphml')

gremlin> graph.commit()

==>null

Check DB has been created successfully:

cma@griffinv1:~$ nodetool cfstats | awk -v RS="----------" '$0 ~ /titan/{print $0}' | egrep "Column Family|Space" | grep -v Bloom

Column Family: vertexindex

Space used (live): 181716333

Space used (total): 184812557

Column Family: edgeindex

Space used (live): 0

Space used (total): 0

Column Family: titan\_ids

Space used (live): 15447

Space used (total): 24027

Column Family: system\_properties

Space used (live): 4652

Space used (total): 4652

Column Family: vertexindex\_lock\_

Space used (live): 198333650

Space used (total): 201464852

Column Family: edgestore\_lock\_

Space used (live): 2845972

Space used (total): 2982054

Column Family: edgestore

Space used (live): 947508252

Space used (total): 951950068

1. **query sales title matching “enterprise data”**

gremlin> VAR2=graph.V.has('title',CONTAINS,'data').has('title',CONTAINS,'enterprise').next()

==>v[837392]

gremlin> VAR2.in('review').count()

==>6

gremlin> VAR2.in('review').

groovysh\_parse: 55: unexpected token: . @ line 55, column 18.

VAR2.in('review').

^

1 error

Display stack trace? [yN] VAR2.in('review')

gremlin> VAR2.in('review').dedup().out().dedup().count()

==>20

gremlin> VAR2.in('review').dedup().out().dedup().transform{it.outE.count()} <=find out customer who are interested in these topics

==>12

==>8

==>21

==>14

==>15

==>16

==>19

==>24

==>19

==>21

==>12

==>10

==>19

==>12

==>26

==>10

==>12

==>22

==>24

==>25

gremlin>

1. **extract customer information and relationship between customer and particular topic**

gremlin> G1=new TinkerGraph()

==>tinkergraph[vertices:0 edges:0]

gremlin> def addV(v,G1) {

gremlin> G1.addVertex(v.id,ElementHelper.getProperties(v))

gremlin> }

==>true

gremlin> def addL(e,G1) {

gremlin> outv = G1.getVertex(e.outV.next().id);

gremlin> if ( outv != null) {

gremlin> inv = G1.getVertex(e.inV.next().id);

gremlin> if ( inv != null) {

gremlin> G1.addEdge(e.id,outv,inv,e.label,ElementHelper.getProperties(e))

gremlin> }}}

==>true

gremlin> books = []

gremlin> VAR2.in('review').filter{it.outE.count() < 1000}.dedup().sideEffect{addV(it,G1)}.out().or(VAR2.both('similar')).dedup().sideEffect{addV(it,G1)}.store(books) <=query relationship and save it into book array

==>v[8530888]

==>v[7578148]

==>v[7099948]

==>v[6612600]

==>v[5804636]

==>v[5711116]

==>v[4134544]

==>v[3476340]

==>v[3148992]

==>v[3037500]

==>v[8407200]

==>v[6429904]

==>v[3109704]

==>v[6658216]

==>v[8072660]

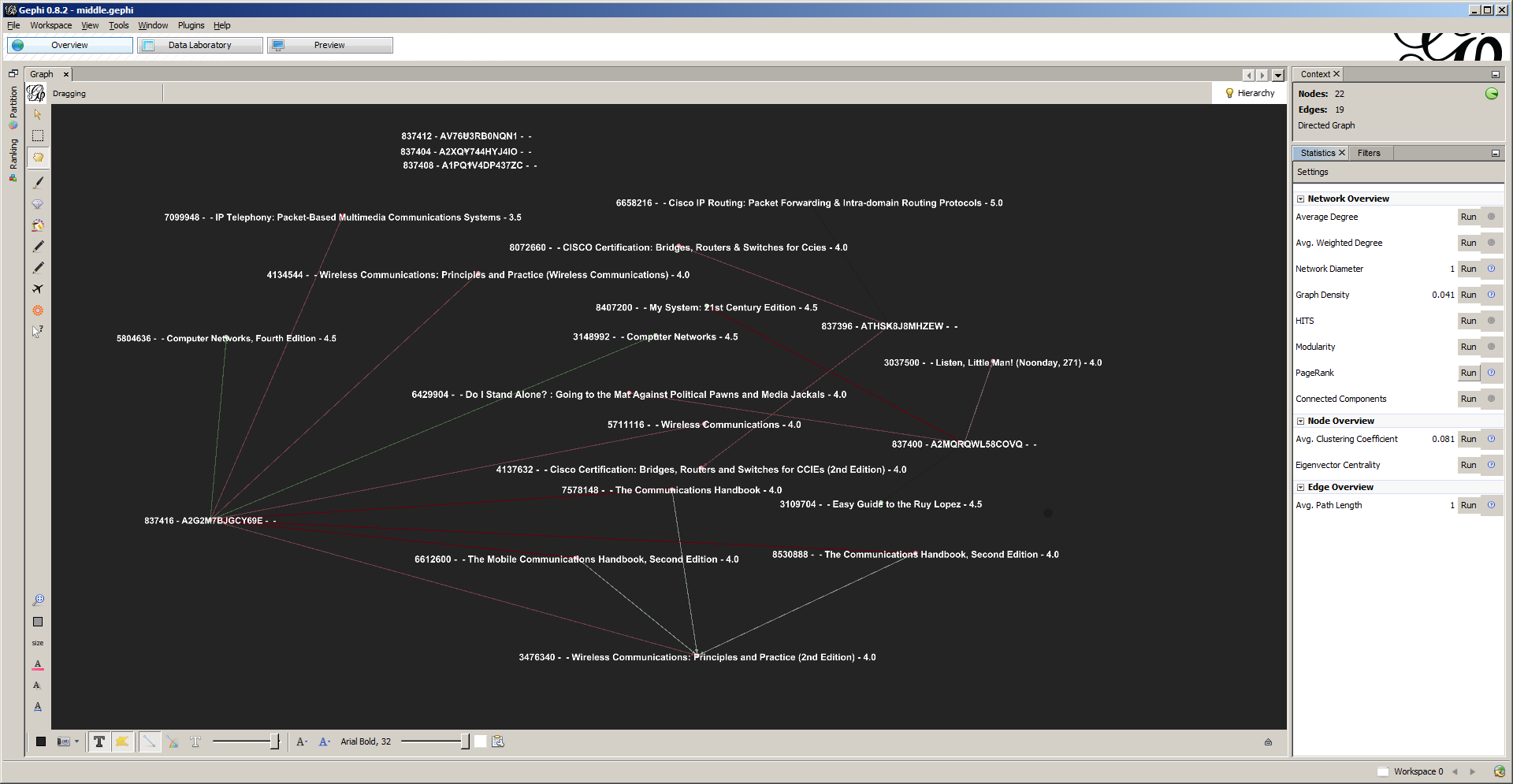
==>v[4137632]

gremlin> books.\_().inE.sideEffect{addL(it,G1)} <=add labels information to vertex points

gremlin> GraphMLWriter.outputGraph(G1,new FileOutputStream('output.graphml')) <=write output to output.graphml file

# Display topic related node and edge

Load data into gephi

****

# Conclusion

For “enterprise data” topic,

Customer 837416 is more interested in network communication, especially wireless communication

Book “wireless communications:principles and practice(2nd edition) is the most popular book in this area.