Review of GE2324

Top1 Introduction

1. Big data

> Volume: Data at rest (TB-EB of existing data to process) Velocity: Data in Motion

> Variaety: Data in many forms

> Veracity: Data in Doubt

Big data is not about the size of the data, it is about the value within the data

Hobbysit

> Byte : one grain of rice

> Kilobyte : cup of rice

Desktop

> Megabyte : 8 bags of rice

> Gigabyte : 3 Semi trucks

Internet

> Terabyte : 2 Container Ships

> Petabyte : Blankets Manhattan

Big Data

> Exabyte : Blankets west coast states

> Zettabyte : Fills the Pacific Ocean

2. The model of generating/consuming data has changed.

> old : Few companies are generating data.

> new: all of us are generationg data and all of us consuming data

3. Big data is not big

> big data -> smart data

4. Data Basics

- Data Definition

> Data are raw facts and figures that on their own have no meaning

> So Raw Data -> Context -> Information

> Information = Data + (Context + Meaning)(get by processing)

- Data Representation

> represented in binary

> change 1877 in dicimal to binary

```
1877 %2 = 938 Remainder 1
938 %2 = 469 Remainder 0
469 %2 = 234 Remainder 1
234 %2 = 117 Remainder 0
117 %2 = 58 Remainder 1
58 %2 = 29 Remainder 0
29 %2 = 14 Remainder 1
14 %2 = 7 Remainder 0
7 %2 = 3 Remainder 1
3 %2 = 1 Remainder 1
1 %2 = 0 Remainder 1
```

> 11101010101

- Data Transmission

> Newwork Transmission Model

> Text->digital->analog->digital->Text

> Data->Packet->Bit->Channel<-Bit->Packet->Data

- Data Storage

> Storage, also known as **mass media** or **auxiliary storage**, refers to the various media on which a computer system can store data.

> Storage devices hold programs and data in units called files.

> Memory is a temporary workplace where the computer transfers the contents of a file while it is being used.

Top2 Network Analysis

1. What is a Network?

> Any system of interconnected linear features

> Networks are sets of nodes connected by edges

> Networks == Graph

2. Network Components

> Actors: Nodes, Vertices Points
>
> Relations: Edges, Arcs, Lines, Ties

3. Division

> > Directed Network: single direction edges exist

> > Undirected Network: can go through it in either way no different
>
> ---
>
> > Weighted: have magnitude
>
> > Uweighted: don't have magnitude

4. properties

(1) Degree

- number of connecting edges

> **indegree**: how many directed edges (arcs) are incident on a node

> **outdegree**: how many directed edges (arcs) originate at a node

> **degree**: (in or out) number of edges on a node

> **Degree sequence**: An ordered list of the degree of each node (from large to small) e.g.

```
in: [2,2,2,1,1,1,1,0]
distribution :[(2,3)(1,4)(0,1)]->3 nodes with degree 2, 4 nodes with degree 1, 1 node with degree 0
out: [2,2,2,1,1,1,1,0]
distribution :[(2,3)(1,4)(0,1)]
degree: [4,3,3,3,3,2,1,1]
distribution :[(4,1)(3,4)(2,1)(1,2)]
```

!!! When calculating degree of directed graph, notice the two arrows

(2) Connected Components

> **Strongly connected components**: Each node within the component can be reached from each other node in the component by following links (正反连接)

> **Weakly connected components**: every node can be reached from every other node by following links in either direction （单项链接 包括单个元素）

> In undirected networks, one simply talks about “connected components"

(3) path

> In a path, each edge can be traveled one time
>
> path != Cycle (loop is a cycle which connects a vertex to itself)
>
> > In directed graph, a loop add 1 to both in-degree and out-degree
>
> > In undirected graph, a loop add 2 to degree

(4) centralization

1. degree centrality(dgree) & normalized degree centrality(degree/n-1) & Degree Centrailization(sum[delta(dmax-di)] / [(n-1)*(n-2)]
2. Betweeneness
3. closeness

(5)Comunity

1. cliques
    - all members know each other

   > maximum clique: largest possible size
   >
   > maximal clique: a clique cannot be extended by including one more adjacent vertex

   . (1) n-cliques
    - two actors in a subnet have a shortest path length at n (n does not have to be part of the subnet i.e. not have to be diameter)

   (2) n-class
    - two actors in a subnet have a shortest path length at n (n has to be part of the subnet i.e. has to be diameter)
2. k-cores
    - actors in subnet knows at least k others in the subnet
3. p-cores
    - def1: p = k/#actors in subnet
    - def2: p = k/#of all neibours of actor

[algorithm to find n-clique(& why we don't use n-club)]()

(6) cohesion in directed & weighted networks

> If propotion of double linked ties in graph is high "social cohesion is high"(#double linked tie*2/# of ties in the graph)

Top4 Cluster analysis

Def

- A grouping of data objects such that the objects within a group are similar(or related) to one another and different from(or unrelated to)the objects in other groups.
- Outliers are object don't belongs to any cluster

Methods

- k - means(k = #group) propose some rough centers (k random numbers)
  1. propose some rough centers
  2. for each item , attach to the nearest center

     (min(dis(i,c1),dis(i,c2)....)) distance = sqrt (delta x^2+delta y^2...)(maybe other defination)

     . refine the center by taking average of all members at that group

     . repeat the step 2 & 3 using the new centers until there's *no change* in the membership (after shift of center there may be new center)
  3. problems:

  > K- mean is sensitive to the center in the beginning (takes longer to run, or failing to go to a good solu)

  > need to specify K in the beginning (change in K need massice re- calculation(no reuse))

- HAC(Dierarchical / Dendrogram method)
  1. merge the two points/groups which are closest together, consider that as a new point(group)

     distance: Euclidean distance, maybe other definition
  2. update the distance from/to all others to/from the new group (how? could be ave (p67 of lec04), could be median, could be min) # DEPENDS ON QUESTION
  3. repeat step 1 & 2 until finally end up with 1 cluster
  4. Dendrogram: recording the sequence of merging(branch closer to bottom == being merged earlier)
  5. Advantage: Need diff num of cluster == cutting at diff point of dendrogram

---

Excel function

```
= IF(<condition>,yes_value,no_value)

= IF(<condition>,yes_value,IF(<condition>,yes_value,no_value))

= if(min(d1,d2,d3)==d1,"Group1",if(min(d1,d2,d3)==d2,"Group2","Group3"))
```

What is a good clustering?

- good is cluster with

  > High intra-class similarity
  >
  > Low iner-class similarity

- precise definition of clustering is difficult

  > subjective
  >
  > application-dependent

Data Correlation

Data Association Measures

- Pearson Correlation

  1: positive linear

  0: no relation

  -1: negative linear

- Spearman correlation
- Kendall's Rank Correlation
- Mutual Information

De-convolute the Data

Causality

Top 6 Association and Frequent Item Analysis

Market-Basket model

- Def
  - item: elements
  - itemset: set (elements' combinaton)
  - basket: sample set (resource)
  - market: the whole things
- Support
  - number of basket contain the itemset
  - frequent itemset: itemsets that support > threshold s

The Apriori Algorithm

- Naive algorithm
  - simply read nC2 pairs(n is number of documents)
  - Caculate support/number of baskets
  - times: nC0+...+nCn - 1 = 2^n -1
- Apriori Algorithm(from size 1 itemset to size n itemset)
  - AprioriPrinciple: if an itemsetis frequent, then all of its subsets must also be frequent.
  - From significant small item set, we can **merge** them and possibly get a bigger item set.
  - if the small small item set is already **not significant**, there is no need to continue eliminate impossible cases early to save time

---

Find interesting association

- confidence: for all recs with LHS, how many of tem (%) also get RHS

  number of LHS / number of (LHS and RHS)

- e.g.

  {b}=>m, {b,m}=>d

  A typical question: “find all association rules with support(support of LHS & RHS mensioned in the rules) ≥s and confidence ≥c.”

- how many? (2^n-2)

- interest: confidence - expectation (minus a penalty caused by natural occurrance)

  interest = confidence - expectation

  expectation = diaper(occurance of item behind the =>) / record(# of basket)

- Support: useful Interest: worth investigating

Top7 Finding similar items

Distance

- Hamming Distance

  > number of different items

- Euclidean Distance

  dist = sqrt(dx^2 + dy^2 + ...)

- Manhattan Distance (MD<=ED)

  dist = sum(abs(dx),abs(dy),....)

- Jaccard Similarity

  dj(a,b)=1 - (# a and b)/(# a or b)

- Distance Metric
  - non-negativity
  - identity (d = 0 iff a=b)
  - symmetry (d(a,b)=d(b,a))
  - triangle inequality (d(a,b)+d(b,c)>d(c,a))

Essential Techniques for finding similar documents

- Shingles(convert document)
- K-gram: a sequence of k characters that appears in the doc
  - E.g. doc = abcab, k = 2, then the set of all 2-gram is {ab,bc,ca}

  > set: cannot be duplicated
  >
  > bag: can be duplicated

- Minhashing/LSH(convert large sets to short signatures)

  Document C1 = {e1, e3, e4, e5}

  Document C2 = {e1, e4, e5}

  > use the table to calculate sim(a,b)=1 - (# a and b)/(# a or b)
  >
  > use the signature(**radom** permutation and calculate **steps to reach "1"**) to calculate sim(siga,sigb)

- Locality Sensitive Hashing(focus on pairs of signatures likely to be **similar**)

> Don't want to check all nC2(colum pairs)

- Possible step:
- Use minhash signatures+hashing t oarrange similar document into buckets. We can compare documents in the same bucket
- Use locality sensitive hasing

> turn a line through origin to meet the point(recording you operation by clockwise or not (0,1))
> compare the operation (use hamming distance)

```
  011100
  011101

> distance = 1
```