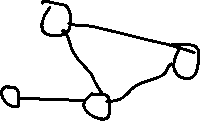
**Graphs & Networks**

Applications – Social Network – FB, Transportation - Logistics, Recommendations, Google Search (Page Rank), Bio field – relationships between Genes, Cybersecurity – map relationships between diff hosts , NLP - DeepWalk

Graph G = (N, E)



Nodes, Edges



Adjacency Matrix – Advantage use Eigen Vector to compute multiple properties – SVD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | N1 | N2 | N3 | N4 |
| N1 | 0 | 1 | 1 | 0 |
| N2 | 1 | 0 | 1 | 1 |
| N3 | 1 | 1 | 0 | 0 |
| N4 | 0 | 1 | 0 | 0 |

Node Degree, N1 -> 2 ; N2 -> 3 ; N3-> 2 ; N4 -> 1

Why find most important node ? -> Social Network -> Pass message -> Product recommend

Data sufficient to find most important node ? -> More things used to diff measure.

Types of Networks

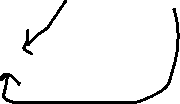
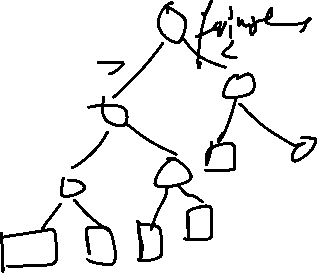
Directed Network – direction in edges – FB Message

Undirected Network – bi directional – FB Friends

Directed Acyclic Graph ex. Airflow DAGs , Spark -> No cycle. Vs Tree -> Go any direction

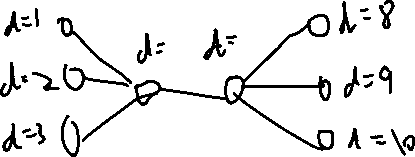
Diagram

Description automatically generated



**Centrality Measures in Network**

Degree Centrality –



Why called Local ?

Eigenvector Centrality – Global ?

Decomposes Adjacency Matrix as Ax = lambda \* x

Diagram

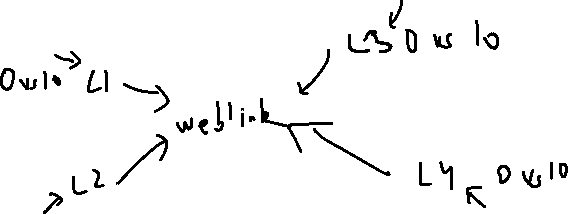
Description automatically generated

Where use ?

New node in network??

PageRank Centrality

What difference from Eigen Vector Centrality



Closeness Centrality – Avg Path Length

Diagram

Description automatically generated

Transportation

10 cities and you have transport from warehouse – Location of warehouse based on closeness centrality

Smaller Lij -> Larger Cc for node (Closeness centrality)

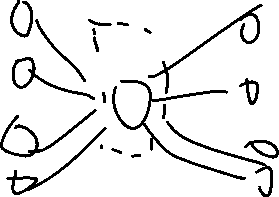
Road Network

Betweenness Centrality - the percentage of shortest paths where our studied node lies in them

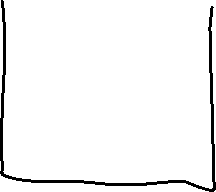
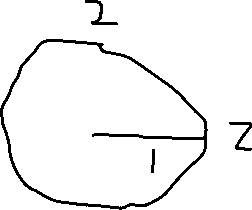
Information passing ?

* Bottleneck

Break Revolution example



**Monte Carlo Simulation**



Area of circle = pie(1)^2 = pie

Area of Sq = 4

Dart in circle = pie/4

N,cir = 10000000000, 0

For I in [1,N]

X= RAND(-1,1)

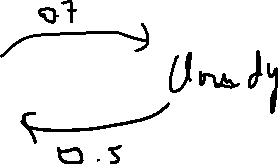
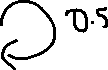
Y= RAND(-1,1)

If x^2 + y^2 <=1:

Cir = cir + 1

Calculate Circ / N

**Markov Chain** - ASS. - Markov Property – Today state dependent on previous day state



* P (Sunny state) -.58 P(Cloudy state) = .42 (Steady State) -> Markov Chain

State Space {Sunny, Cloudy}

Markov Ass – State on given day depends on state on previous day only

P(Wt | Wt-1 Wt-2 …) = P(Wt | Wt-1)

Transition Prob Matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Wt | |
|  |  | Sunny | Cloudy |
| Wt-1 | Sunny | 0.3 | 0.7 |
| Cloudy | 0.5 | 0.5 |

Qn – When achieve steady state – Prob converge to any one state

Assume W0

S - 1 C – 0

W1

S -0.3 (0.3\*1 + 0.5\*0) C – 0.7 (0.7\*1+0.5\*0)

W1

S - 0.44 C – 0.56

….

If assume steady state, Ps and Pc

Steady for Sunny –

0.3\*Ps + 0.5\*Ps = Ps => Ps = 0.42 & Pc = 0.52

Application

I eat Pizza

Pronoun Verb Noun

Find Prob(Verb | Noun) -> Using transition state probabilities

**Hidden Markov Model**

Sequence Observed

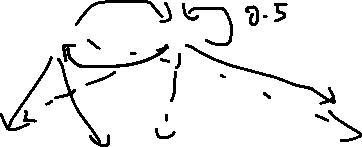
Red -> Green -> Blue

What is the sequence of hidden states -

1. Happy -> Happy -> Sad? b. Happy -> Sad -> Happy? c. So on



HAPPY SAD



RED GREEN BLUE

T

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Wt | |
|  |  | Happy | Sad |
| Wt-1 | Happy | 0.7 | 0.3 |
| Sad | 0.5 | 0.5 |

E

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Red | Green | Blue |  |
| Happy | 0.8 | 0.1 | 0.1 | 1 |
| Sad | 0.2 | 0.3 | 0.5 | 1 |

Start – No Prior

Observe -> Given

Day 1 G

Day 2 B

Day 3 R

Most Likely moods {M1, M2, M3}

P(A|B) = P(A and B) \* P(B) = P(B|A)\*P(B)/P(A)

Max(m1 m2 m3) P(C1=G C2=B C3=R | M1=m1,M2=m2,M3=m3)

= P(C3|C2,C1,M3,M2,M1)\*P(C2|C1,M2,M1)\*P(C1|M1) \*P(M3|M2,M1) \*P(M2|M1)\*P(M1) (Bayes Theorem)

Markov Ass – Mood today depends on yesterday only

+ Color shirt only depend on mood that day

= P(C3|M3)\*P(C2| M2)\*P(C1| M1) \*P(M3|M2) \*P(M2|M1)\*P(M1|Started)

E Matrix T Matrix

* Maximize prob of observed data with hidden states sequence using Hidden Markov Property assumption.
* Happy Happy Sad

**Applications of HMM**

**NLP**

I eat Pizza (Observed Variable)

Pronoun verb noun (Part of Speech)

I am teaching NLP -🡪 Max Prob of POS for this sentence Pronoun verb verb Noun

Assign POS to Observed Variable

Use HMM

Matrix

POS

POS Transition Prob

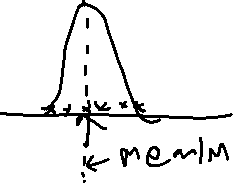
POS

Words Probability

**Kalman Filter**

Iterative Mathematical process that uses a set of equations and consecutive data inputs to quickly estimate true value, position, velocity, etc of object being measured, when measured values contain unpredicted or random error, uncertainty or variation

Track and estimate state of the system – Use case self driving cars



Text, letter

Description automatically generated

Text

Description automatically generated

1. State variable that contains the current value of the measurements being tracked (e.g. position, velocity)
2. the Process variable contains the predictive error of those measurements.
3. Each time step, the state transition matrix(A) moves the state and process matrix based on the current position and velocity, estimating a new position/velocity as well as new covariance.
4. From there, the Kalman Gain is calculated, along with the observed data.
5. The update process involves using the Kalman in conjuction with the previous estimate and new observed data to update the state variable towards a belief that’s somewhere between the prediction and measurement. The process covariance is also updated based on the Kalman gain.

6. These updates are then used for the next round of predictions.

Kalman Gain –

**K = Eest /(Eest + Emea)**

High – Emphasize current observation over past data and filter output highly sensitive (Emea small and E estimate high)

Low - Emphasize past data over current observation and filter output highly sensitive (Eest small and E measurement high)

Error in estimate vs Error in measurement

Xt = xt-1 +K(p - xt-1)

P – measurement xt – estimate at time t

[Understanding Kalman Filters with Python | by James Teow | Medium](https://medium.com/@jaems33/understanding-kalman-filters-with-python-2310e87b8f48)

**Application of using Neural Network in Graph**

[**https://towardsdatascience.com/deepwalk-its-behavior-and-how-to-implement-it-b5aac0290a15**](https://towardsdatascience.com/deepwalk-its-behavior-and-how-to-implement-it-b5aac0290a15)

Words in NLP presented in 3 ways -

I am teaching today today

1. Count Vectorizer (1, 1, 1, 2)
2. TF IDF Vectorizer
3. Word Embeddings

Latent Attributes

Royal Gender Food

King 0.9 0.7 0.1

Burger 0.1 0.2 0.9

Final Model – Much better predictions ( Add context meaning of words in embeddings)

**Deep Walk – Example of creating embedding from graph relationships**

Graph

Uses the relationships in the graph, embeddings

Product Graphs P1 -> P2 -> P3

Food Laundary Cricket

Bat 0.1 0.1 0.9

Maggie 0.9 0.1 0.1

**Day in Life of Data Scientist**

Skills expected as of today in order of highest to lowest –

1. Python
2. SQL
3. Cloud – AWS / GCP / Azure –

Data deployment project – Kinesis, s3, RedShift, Glue, Athena, Lambda, EMR – Spark, Kubernetes

ML project using Sagemaker on s3 or Redshift data

Deployment on AWS using lambda functions (Flask)

1. Machine learning (sklearn)
2. Deep Learning Framework – Recommend Pytorch ( Tensorflow +Keras)
3. Github (vvvvvv imp)
4. Distributed programming on Clusters -> Spark mllib, Kubernetes – DL framework with cuda or Dask or joblib
5. Knowledge of Dockers to deploy jobs on clusters
6. Airflow or CRON scheduler
7. CI / CD Deployment

Reference courses on Udacity or Udemy (Jose Portella, Lazy Programmer, Soledad Galli, Christopher Samiullah)

Reference – Medium / Towards Data Science

Projects on Kaggle