Lecture 6 - Link Prediction

Monday, January 31, 2022 8:00 AM

Today: link prediction

Thursday: collaborative filtering

Next week: centrality

Review:

Triadic closure 

strong vs. weak ties

Homophily

Preferential attachment

Link prediction

Inferring the growth process of a network → predicting which links are most likely to form

Facebook: who should you be friends

Amazon: what you will purchase

aka for several of the above

→ Recommender systems

→ a system that recommends thing

Facebook: recommend friends
Amazon: recommend products

Also: inferring "hidden links"

FB: friends in reality, but not on Facebook

Maybe, you're trying to hide the fact:
- Impropriety
- Terrorists "hiding in plain site"

Issue: intentional noise

Other examples: financial transaction networks + $$ laundering

→ problems generally reduce to eliminating noise and finding most probable links / reality

Our general approach for link prediction:

link prediction.

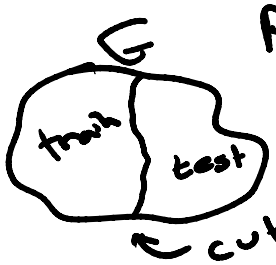
→ considering network at time t_0

→ use topological features (and/or) metadata to predict most probable links

Validate: test on time $t_0 + \Delta t$

→ can compare separate methods

OR: use train + test partitioning of graph



⇐ partitioning

↳ useful for "hidden links"

Unsupervised methods for link prediction

↳ not comparing/training to an explict ground truth

What are some methods:

Triadic closure

Triadic closure

Strong triadic closure

Common neighbors: $C(x,y) = |N(x) \cap N(y)|$

Preferential attachment: $P(x,y) = |N(x)| * |N(y)|$

$P(x,y) = d(x) * d(y)$

Jaccard Index: $J(x,y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$

Adamic-Adar: $\sum_{w \in N(x) \cap N(y)} \frac{1}{\log(|N(w)|)}$

considering common neighbors of x,y but minizing or biasing against large degree vertices

Personalized PageRank:

we'll discuss next week

---

code module

- consider DNC temporal network
- compare some of the above

- Compare some of the above methods in terms of prediction

Here: we're just considering topological properties of the network
Not training via ground truth
↪ unsupervised

Results: alright on ONC social comm. net.
→ ones directly relevant to social network growth did better

On other networks: you'll see on HW 1
→ limited predictive strength
→ triadic closure etc. isn't a driving growth process on a non-social network

---

Gettin' supervised

In general: supervised methods use some notion of "ground truth"

same notion of g...

Here: existing links and associated topology + metadata (and non-existing links)

↳ create "features" using this

↳ use these to "train" some algorithm

Today: we'll use matrix factorization

However: our features are not explicit

↳ for other approaches, use explicit features

↑ project idea

Note: difficult to get graph + metadata due to privacy concerns

---

Matrix factorization for link prediction

Consider:

- We have no metadata
- We don't have a good notion of what "explicit" features to use
- We want a general approach that is dataset-independent

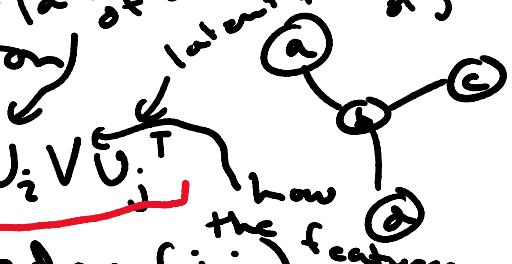
is dataset-independent

Matrix factorization in general → $X = UV$ (matrix, factors)
aka matrix decomposition

- Take some some matrix X, and "factorize" it into some matrix product
  - You might have seen:
    * Singular value decomp. → $X = U\Sigma V^*$
    * solving linear systems → $X = LV$
    * Cholesky, QR, Eigendecomps, etc

In the context of link prediction:

- we'll consider $A = UVU^T$ ← adjacency matrix
- prediction for $a_{ij} = U_i V U_j^T$
  - "latent features of vertex i"
  - latent features of j
  - how the features interact
  - edge (i,j)



	a	b	c	d
a	0	1	0	0
b	1	0	1	1
c	0	1	0	0
d	0	1	0	0

a_{ij} = nonzero $\in A$ ↳ there exists edge $(i,j) \in G$

- we're trying to minimize $\min_{U,V} \sum_{\text{nonzeros in } A} |a_{ij} - U_i V U_j^T|$
- Really $\min_{U,V} \sum (a_{ij} - U_i V U_j^T)^2$
- [illegible] optimize: gradient descent

~ U,V

→ How to optimize: gradient descent

Pros: don't need to explicitly construct features
scalable, easy to to train
can get high quality w/o much effort

Cons: weak generalization
need to retrain with novel data