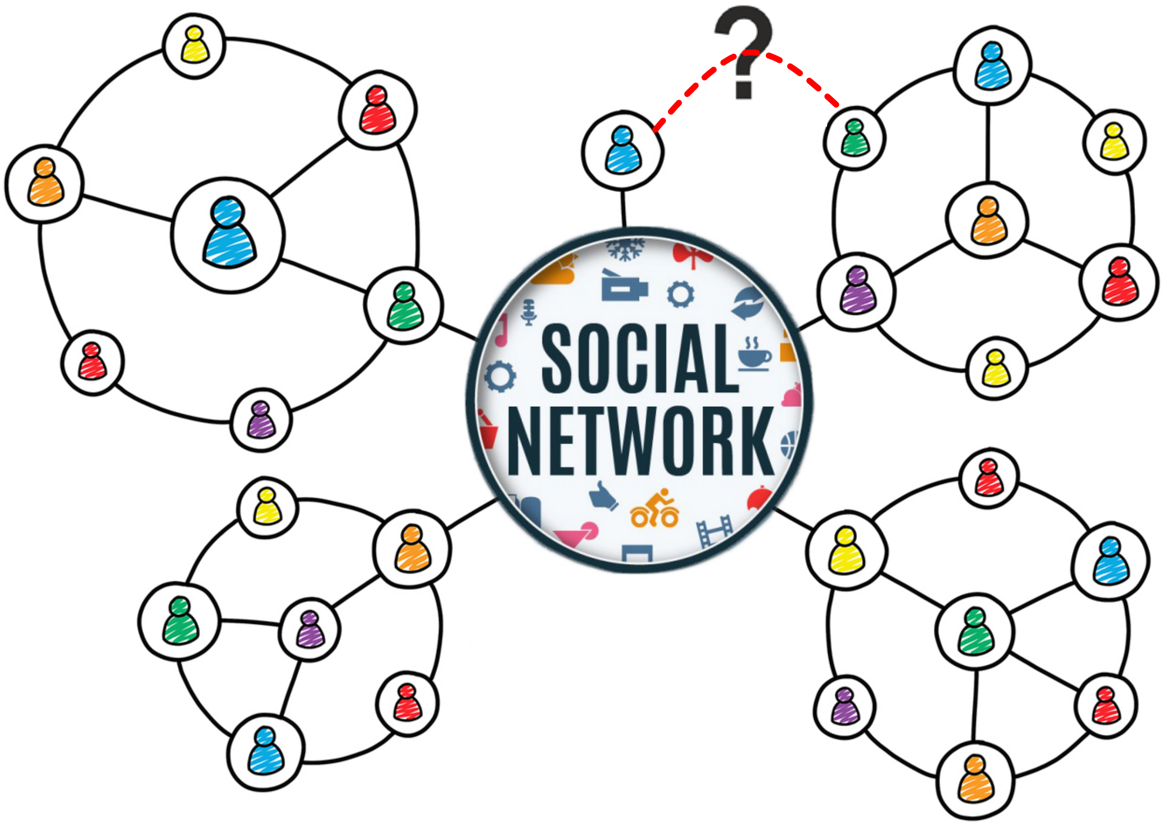
Leveraging Network Dynamics for Improved Link Prediction

Alireza Hajibagheri (University of Central Florida)
Gita Sukthankar (University of Central Florida)
Kiran Lakkaraju (Sandia National Labs)

Summary

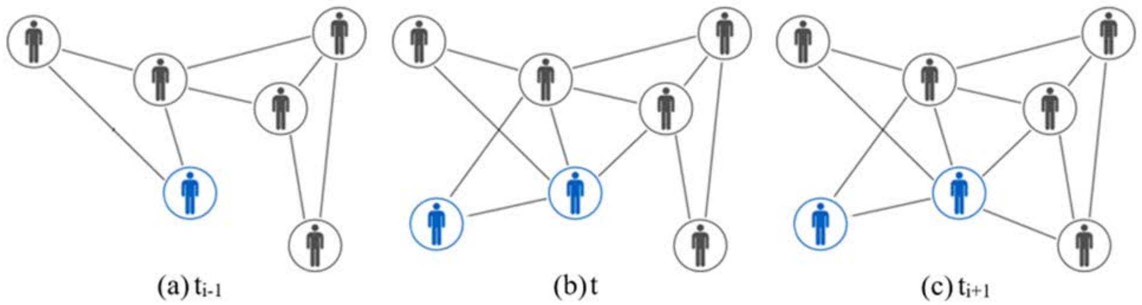
- A new link prediction model that improves prediction performance by learning link formation rates from data
- Demonstrated to work on a diverse group of datasets
- Spark implementation can handle a large set of training data

Introduction



Introduction

- Social networks are in flux



- Fully modeling the dynamics that drive the evolution of a network is a complex problem
- The **rate** of network change

Background

- Link Prediction Models
- Categories:
  - Unsupervised
  - Supervised

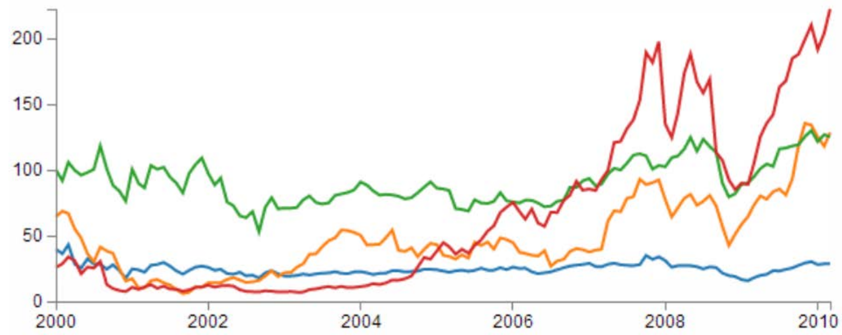
Background

- Unsupervised
  - Non-connected pairs are ranked based on similarity measure
  - Top *k* ranked are selected as future links
  - Simple and generalizes easily
  - But there are limitations
    - Cut off threshold
    - Links with highest scores are most likely to form the earliest, this is not true!
    - Does not consider earlier time slices

Background

- Supervised
  - Well suited for link prediction
  - Create labeled datasets of node pairs
  - Can simultaneously handle multiple structural patterns
  - Accurately fit model parameters based on training data
  - Limitations
    - Less generalizable
    - Unbalanced datasets
    - Does not consider earlier time slices

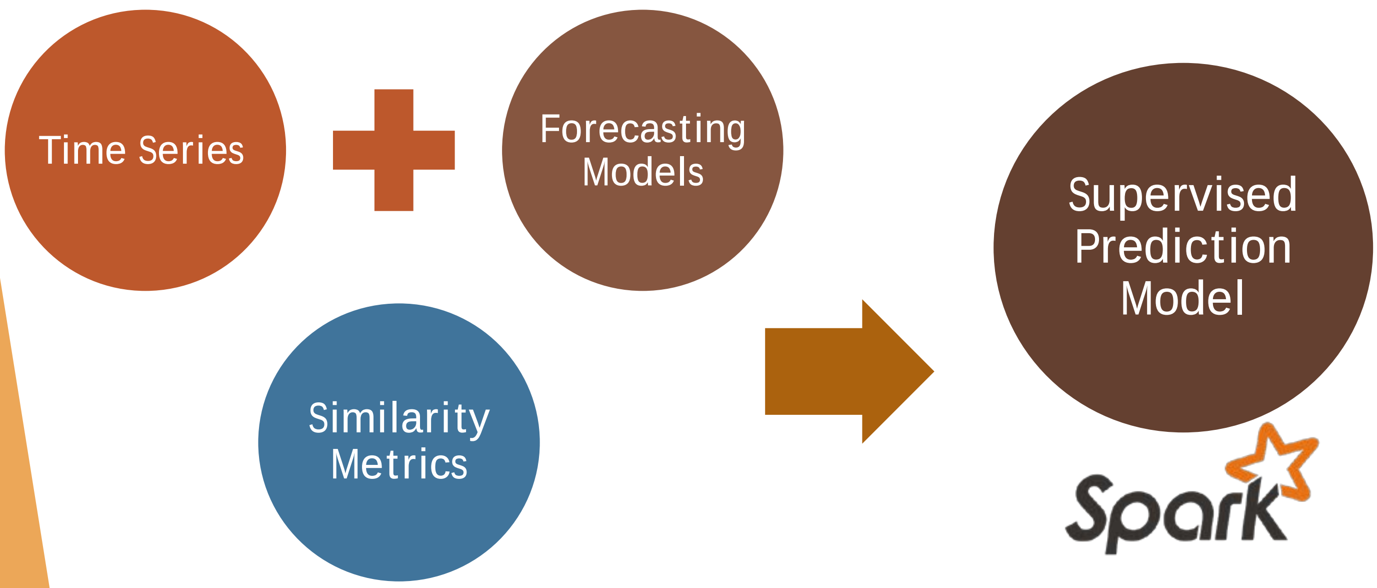
Background



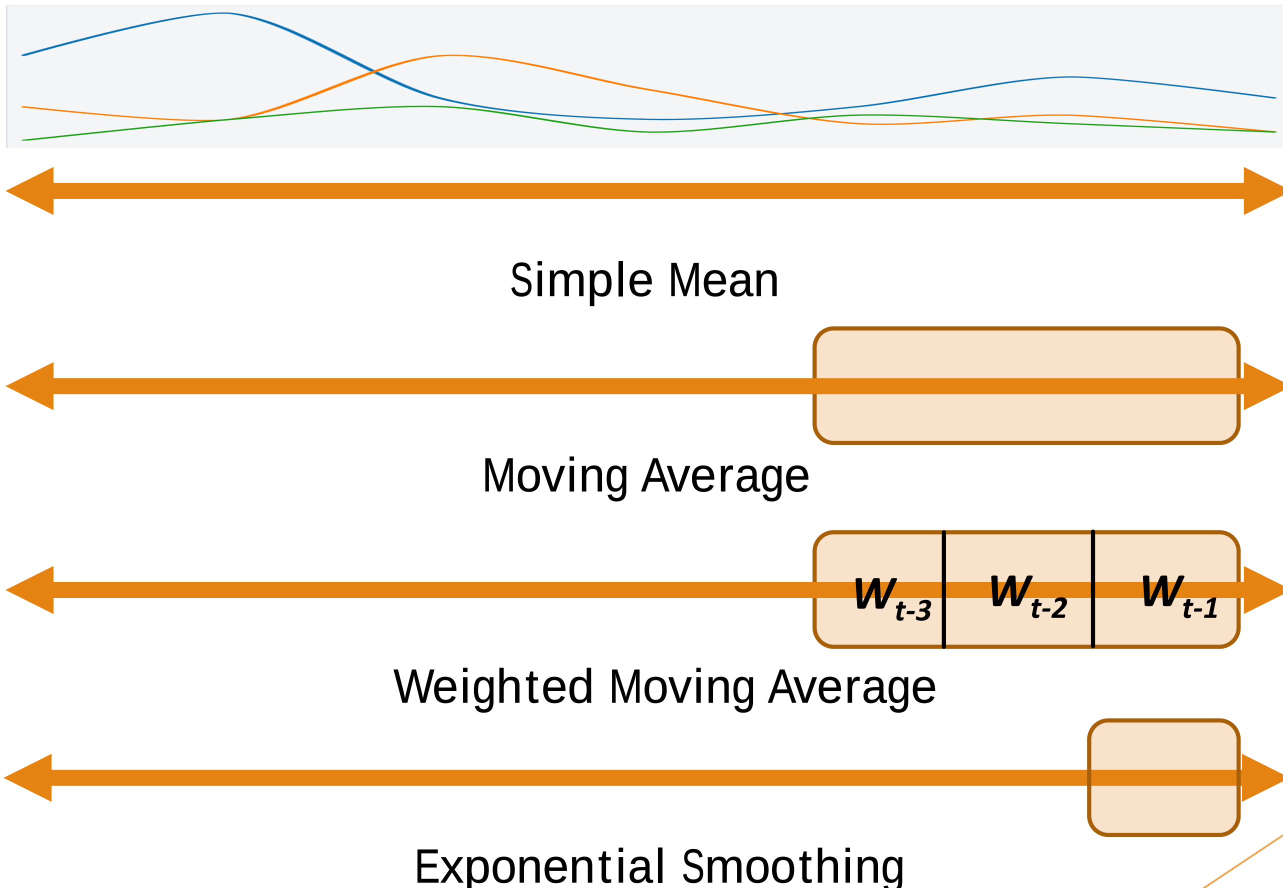
- Issue: current models are not able to address previous changes
- Network structure up to time *t* is used to predict time *t+1*
- Our solution?
  - Individuals' link formation rate
  - Rate is a user’s number of links at each time slice

Background

- RPM (**R**ate **P**rediction **M**odel)



Method (Time Series)



Method (Similarity Metrics)

- Common Neighbors

$$CN(x,y) = |N(x) \cap N(y)|$$

- Preferential Attachment

$$PA(x,y) = |N(x)| \times |N(y)|$$

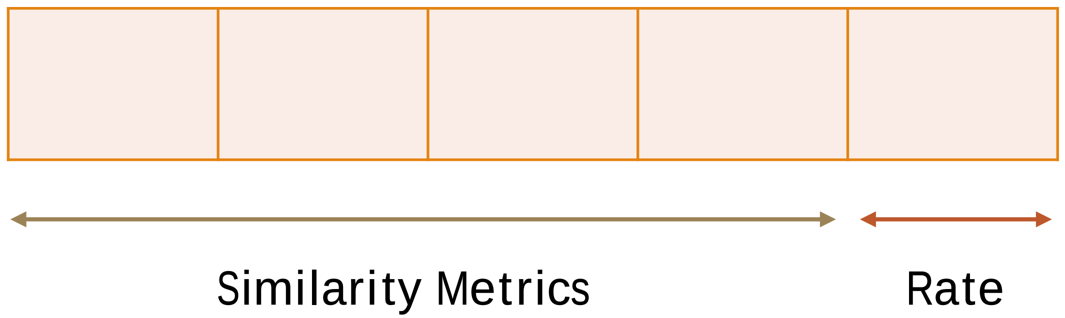
- Jaccard Coefficient

$$JC(x,y) = |N(x) \cap N(y)| / |N(x) \cup N(y)|$$

- Adamic/Adar

$$AA(x,y) = \sum_{z \in |N(x) \cup N(y)|} \frac{1}{\log(|N(z)|)}$$

Method (Supervised Model)



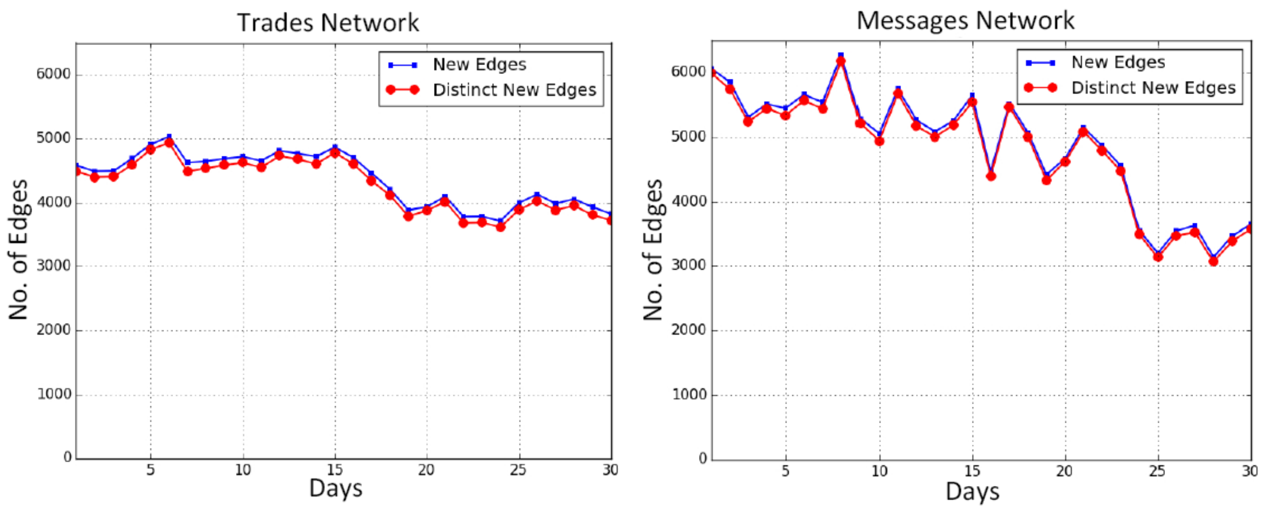
- Supervised Model:
  - SVM
  - Unbalanced data
  - Train at time ***t*** to predict time ***t+1***

Results (Datasets)

- Datasets selected from different networks
  - MMOG (Travian)
  - Communication (Enron)
  - Co-authorship (arXiv)

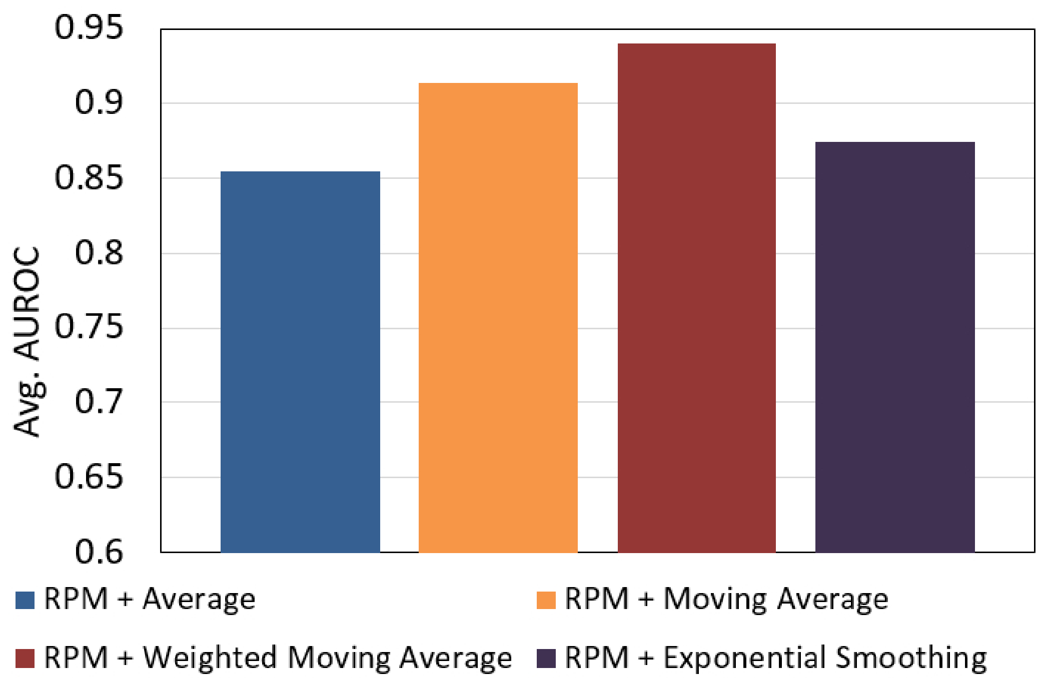
Data	Enron	Travian (Messages)	Travian (Trades)	hep-th
No. of nodes	150	2,809	2,466	17,917
Link (Class 1)	5,015	44,956	87,418	59,013
No Link (Class 0)	17,485	7,845,525	5,993,738	320,959,876
No. of snapshots	24	30	30	20

Results (Datasets)



New Edges (Blue) vs. Distinct New Edges (Red)

Results (Choice of Forecasting Model)



$C_{t-3}=0.2,\ C_{t-2}=0.3,\ C_{t-1}=0.5$

Results (Platform)

- Incorporating Spark

Spark

open source cluster computing framework
http://spark.apache.org/

Results (Methods and Performance Measure)

- Methods
  - Supervised-MA
  - Supervised
  - Four unsupervised models
- Performance measure
  - AUROC

Results (AUROC Performance)

Algorithms / Networks	Travian(Messages)	Travian(Trades)	Enron	hep-th
RPM	**0.8970**	**0.7859**	**0.9399**	**0.7834**
Supervised-MA	0.8002	0.6143	0.8920	0.7542
Supervised	0.7568	0.7603	0.8703	0.7051
Common Neighbors	0.4968	0.5002	0.7419	0.5943
Jaccard Coefficient	0.6482	0.4703	0.8369	0.5829
Preferential Attachment	0.5896	0.5441	0.8442	0.5165
Adamic/Adar	0.5233	0.4962	0.7430	0.6696

- **RPM** outperforms all other methods
- Supervised-MA is the next best option
- Supervised can get reasonable results

Conclusion

- RPM
  - Identifies the most active individuals
  - Improved link prediction performance
  - Spark reduces training time required
- Future Work
  - Multilayer networks
  - A generalizable model

Thank You ☺