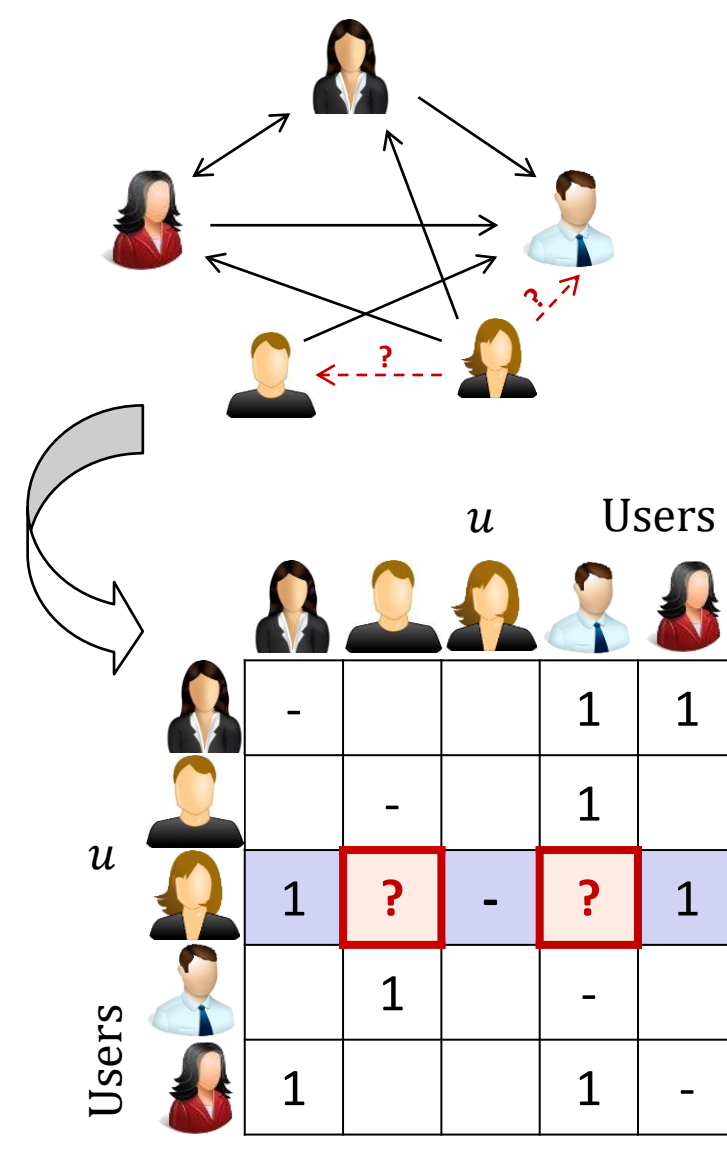


RELISON: A Framework for Link Recommendation in Social Networks

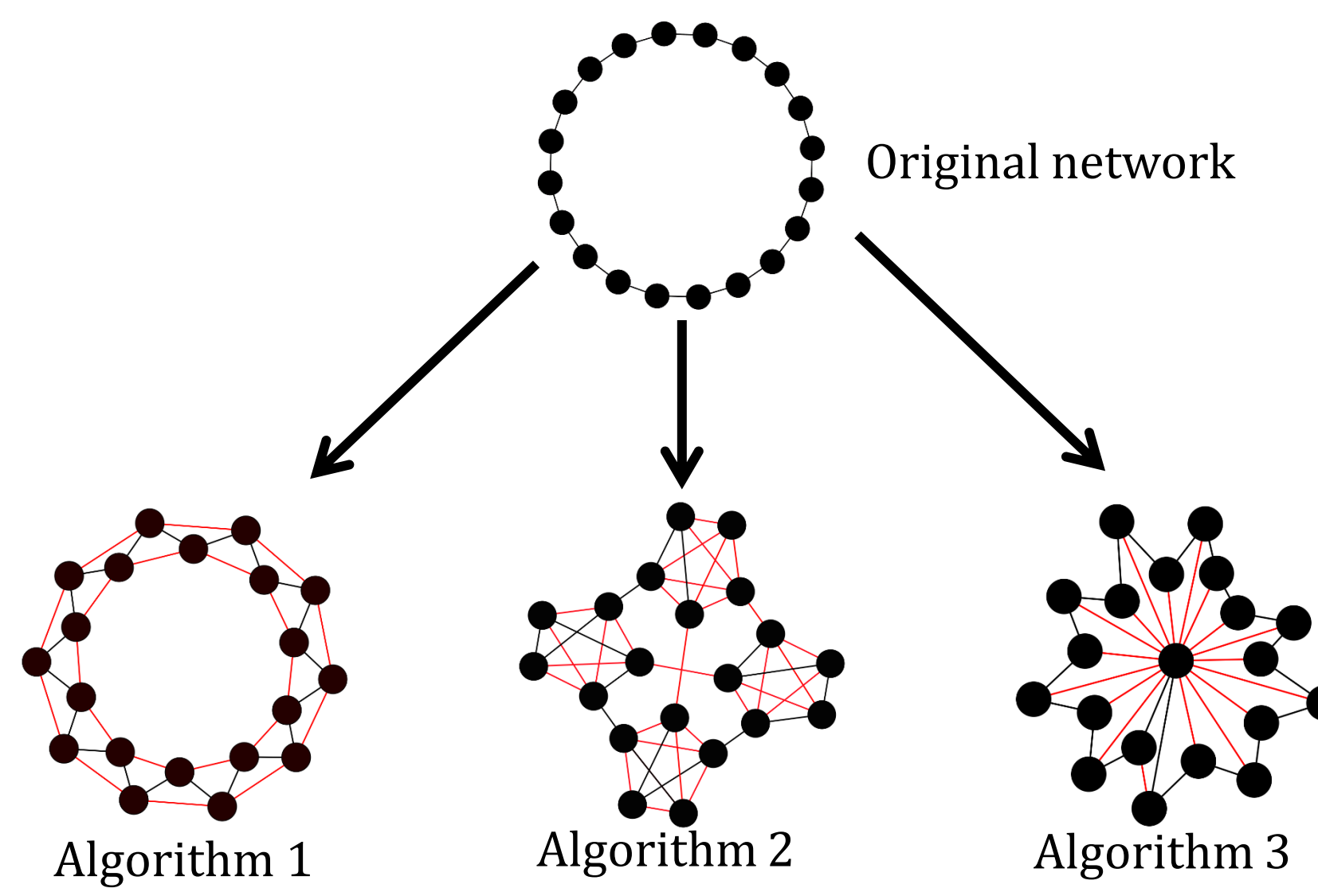
The contact recommendation task

- Given:
 - A social network $\mathcal{G} = \langle \mathcal{U}, E \rangle$
 - \mathcal{U} : Set of users
 - $E \subset \mathcal{U}^2 = \mathcal{U}^2 \setminus \{(u, u) | u \in \mathcal{U}\}$: Edges
 - Neighborhoods for each user $u \in \mathcal{U}$, $\Gamma(u)$
 - $\Gamma_{in}(u) = \{v \in \mathcal{U} | (v, u) \in E\}$
 - $\Gamma_{out}(u) = \{v \in \mathcal{U} | (u, v) \in E\}$
- For each $u \in \mathcal{U}$, predict k users which might be of interest
 - $\hat{\Gamma}_{out}(u) = \langle u_1, u_2, \dots, u_k \rangle$ $u_j \in \mathcal{U} \setminus (\{u\} \cup \Gamma_{out}(u))$
- Particularities w.r.t. classic recommendation
 - Recommended items belong to the user set
 - Users (and consequently, items) are not isolated
 - Recommendations affect the network structure



Motivation

Effects on network structure

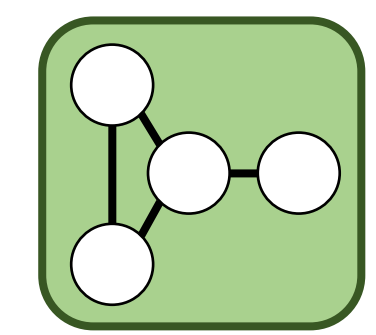


Reproducibility in recommender systems

- A challenge:
 - Different domains / data sources
 - Different evaluation metrics / tasks
- Efforts on open-source frameworks
 - Elliot
 - Lenskit
 - LibRec
 - MyMediaLite
 - RankSys
 - Beta-Recsys
 - ...

Is there a framework for contact recommendation?

RELISON



RELISON

- Java framework for **RE**commending **L**inks in **S**ocial **N**etworks
- Extensible
- Tools for experimentation on link recommendation
- Evaluating recommendation accuracy / novelty and diversity
- Analyzing the effect of recommendations on
 - Network structure
 - Information diffusion
- Network structure analysis and information diffusion functionalities can be used without recommendations

Experimentation

- Two ways
 - Integrating the library in another program
 - Pre-defined command line commands
- Command line programs
 - Link prediction / recommendation + evaluation
 - Social network analysis
 - Community detection
 - Information diffusion
 - Random graph generation
 - Inverted index creation for user-generated contents

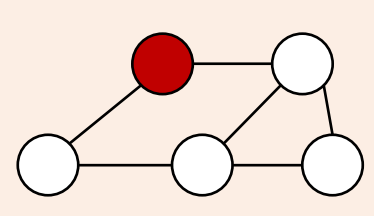
Overview of the framework

Core

- Basic functionalities
- Supported network types:
 - Simple
 - Multigraph
 - Weighted
 - Unweighted
 - Directed
 - Undirected
- Reading / writing networks
- Random network generation
 - Random attachment
 - Preferential attachment
 - Watts-Strogatz small-world networks

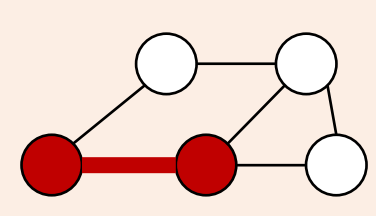
Social network analysis

- Understanding structural properties of social networks
- 50+ structural metrics, including
- Community detection and network partitions
 - Automatic detection of clusters of tightly connected users
 - Measuring partition quality: modularity
 - 8 algorithms



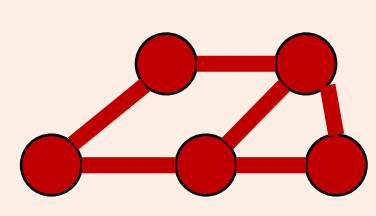
Node

- Betweenness
- Closeness
- Degree
- PageRank



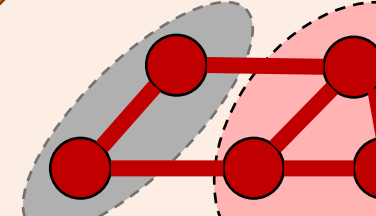
Edge/pair

- Betweenness
- Distance
- Embeddedness
- Reciprocity rate



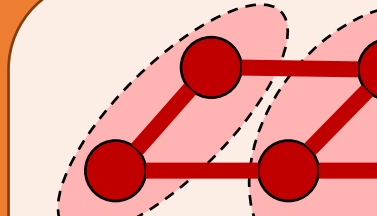
Graph

- Assortativity
- Clustering coef.
- Density
- Diameter



Community (individual)

- Degree
- Size
- Volume



Community (global)

- Edge Gini
- Modularity

Connectedness

- Strongly connected components
- Weakly connected components

Modularity

- Louvain
- FastGreedy
- Infomap

Other

- Girvan-Newman
- Spectral clustering
- Label propagation

Link recommendation

- Contact recommendation functionalities
- Built on top of RankSys
- 50+ algorithms for people / contact / link recommendation, including
- Global reranking (targeting structural properties)
 - Basic implementations (metric agnostic)
 - Optimized reranking (clustering coefficient, Gini-based community metrics)
- Also used for link prediction
 - Unique ranking for all the links
 - Link prediction evaluation measures

Collaborative filtering

- User-based kNN
- Item-based kNN
- Implicit matrix factorization

Common neighbors

- Adamic-Adar
- Cosine
- Jaccard
- IR models

Path-based algorithms

- Distance
- Katz
- Local path index

Random walks

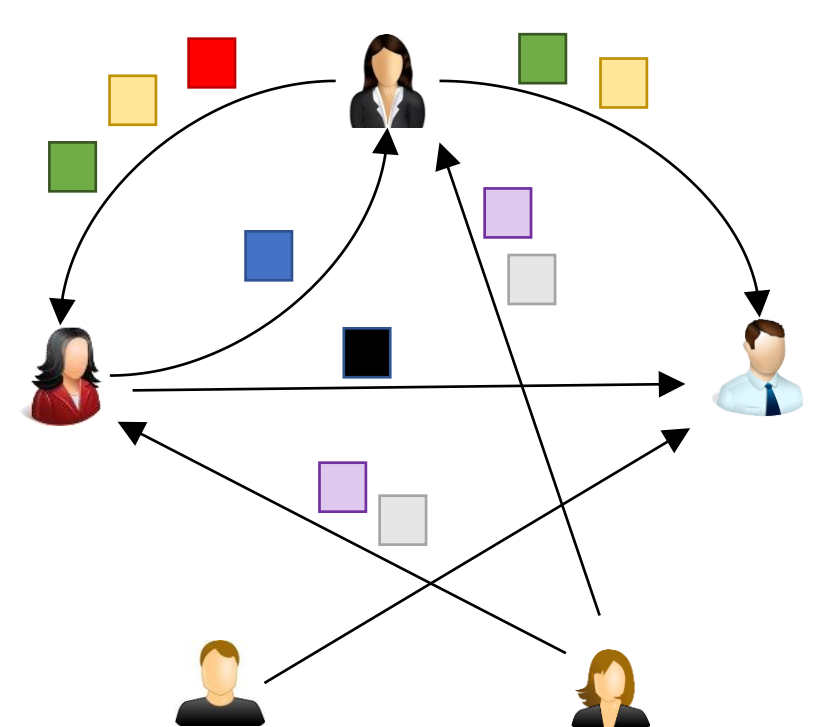
- PageRank
- HITS
- Money
- Hitting time

Other algorithms

- Twittomender
- Supervised classifiers
- LambdaMART

Information diffusion

- Simulation of the exchange of information in social networks
- Concurrent propagation of multiple user-generated contents
- Highly configurable
 - Which contents do users propagate?
 - Which users receive those contents?
 - Which contents draw users' attention?
 - Can users re-propagate information?
 - How do they decide to re-propagate?
- Pre-configured simulation models provided
- Metrics for measuring diffusion
 - Speed
 - Information novelty and diversity



Content

- Module for user-generated content analysis
- Support for content-based recommendation algorithms
- Building inverted indexes from the contents.

