Collective Graph Identification

Lise Getoor
University of California, Santa Cruz

Workshop on Incomplete Networked Data
March 23, 2016
Graph Analytics + Insights!
Graph Analytics + 1 + 1 = 3 WRONG!
Co-Author Graph

before

after
Motivation: ER and Network Analysis

- Measuring the topology of the internet ... using traceroute
Figure 2. The IP alias resolution problem. Paraphrasing Fig. 4 of [50], traceroute does not list routers (boxes) along paths but IP addresses of input interfaces (circles), and alias resolution refers to the correct mapping of interfaces to routers to reveal the actual topology. In the case where interfaces 1 and 2 are aliases, (b) depicts the actual topology while (a) yields an “inflated” topology with more routers and links than the real one.
Figure 3. The IP alias resolution problem in practice. This is reproduced from [48] and shows a comparison between the Abilene/Internet2 topology inferred by Rocketfuel (left) and the actual topology (top right). Rectangles represent routers with interior ovals denoting interfaces. The histograms of the corresponding node degrees are shown in the bottom right plot. © 2008 ACM.
IP Aliasing Problem  [Willinger et al. 2009]
Lesson: Make sure you are working on the right graph before performing analytics!

How do you get the right graph? Infer it from the data!
Graph Inference Patterns

- Entity Resolution
- Collective Classification
- Link Prediction
Entity Resolution: determining which nodes refer to same underlying entity
Collective Classification: inferring the labels of nodes in a graph
Collective Classification

Question: Democrat or Republican?
Collective Classification
Collective Classification
Common Graph Inference Patterns

- Collective Classification [✓]
- Link Prediction
- Entity Resolution
Link Prediction: inferring the existence of edges in a graph
Link Prediction

- **Entities**
  - People, Emails

- **Observed relationships**
  - communications, co-location

- **Predict relationships**
  - Supervisor, subordinate, colleague
Graph Inference Patterns

- ✔ Entity Resolution
- ✔ Collective Classification
- ✔ Link Prediction
Graph Identification

• Goal:
  – Given an input graph infer an output graph

• Three major components:
  – Entity Resolution (ER): Infer the set of nodes
  – Link Prediction (LP): Infer the set of edges
  – Collective Classification (CC): Infer the node labels

• Challenge: The components are intra and inter-dependent
Graph Identification

Input Graph: Email Communication Network

Output Graph: Social Network

Label: CEO Manager Assistant Programmer
Graph Identification

Input Graph: Email Communication Network

Output Graph: Social Network

• What's involved?
• What’s involved?
  • Entity Resolution (ER): Map input graph nodes to output graph nodes
Graph Identification

What’s involved?

• Entity Resolution (ER): Map input graph nodes to output graph nodes
• Link Prediction (LP): Predict existence of edges in output graph
Graph Identification

• What’s involved?
  • Entity Resolution (ER): Map input graph nodes to output graph nodes
  • Link Prediction (LP): Predict existence of edges in output graph
  • Node Labeling (NL): Infer the labels of nodes in the output graph
Graph Identification

• Goal:
  – Given an **input graph** infer an **output graph**

• Three major components:
  – **Entity Resolution (ER)**: Infer the set of nodes
  – **Link Prediction (LP)**: Infer the set of edges
  – **Collective Classification (CC)**: Infer the node labels

• Challenge: The components are intra and inter-dependent
Collective reasoning over richly structured complex graphs
Collective Classification
Collective Classification

Donates(A, 🦁) => Votes(A, 🦁): 5.0

Mentions(A, “Affordable Health”) => Votes(A, 🐴): 0.3
Collective Classification
Collective Classification

\[ \text{vote}(A, P) \land \text{friend}(B, A) \rightarrow \text{vote}(B, P) : 0.3 \]

\[ \text{vote}(A, P) \land \text{spouse}(B, A) \rightarrow \text{vote}(B, P) : 0.8 \]
Collective Classification
Link Prediction

- People, emails, words, communication, relations

- Use model to express dependencies
  - “If email content suggests type X, it is of type X”
  - “If A sends deadline emails to B, then A is the supervisor of B”
  - “If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues”
Link Prediction

- People, emails, words, communication, relations
- Use model to express dependencies
  - “If email content suggests type X, it is of type X”
  - “If A sends deadline emails to B, then A is the supervisor of B”
  - “If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues”

\[
\text{HasWord}(E, \text{“due”}) \Rightarrow \text{Type}(E, \text{deadline}) : 0.6
\]
Link Prediction

- People, emails, words, communication, relations
- Use model to express dependencies
  - “If email content suggests type X, it is of type X”
  - “If A sends deadline emails to B, then A is the supervisor of B”
  - “If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues”

\[
\text{Sends}(A,B,E) \land \text{Type}(E,\text{deadline}) \Rightarrow \text{Supervisor}(A,B) : 0.8
\]
Link Prediction

- People, emails, words, communication, relations
- Use model to express dependencies
  - “If email content suggests type X, it is of type X”
  - “If A sends deadline emails to B, then A is the supervisor of B”
  - “If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues”

\[ \text{Supervisor}(A, B) \land \text{Supervisor}(A, C) \implies \text{Colleague}(B, C) : \infty \]
Entity Resolution

- Entities
  - People References
- Attributes
  - Name
- Relationships
  - Friendship
- Goal: Identify references that denote the same person
Entity Resolution

- References, names, friendships
- Use model to express dependencies
  - “If two people have similar names, they are probably the same”
  - “If two people have similar friends, they are probably the same”
  - “If A=B and B=C, then A and C must also denote the same person”
Entity Resolution

- References, names, friendships
- Use model to express dependencies
  - ‘‘If two people have similar names, they are probably the same’’
  - ‘‘If two people have similar friends, they are probably the same’’
  - ‘‘If A=B and B=C, then A and C must also denote the same person’’

\[
A.\text{name} \approx_{\text{str\_sim}} B.\text{name} \Rightarrow A \approx B : 0.8
\]
Entity Resolution

- References, names, friendships
- Use model to express dependencies
  - “If two people have similar names, they are probably the same”
  - “If two people have similar friends, they are probably the same”
  - “If A=B and B=C, then A and C must also denote the same person”

\[ \{A.\text{friends}\} \approx \emptyset \{B.\text{friends}\} \Rightarrow A \approx B : 0.6 \]
Entity Resolution

- References, names, friendships
- Use model to express dependencies
  - “If two people have similar names, they are probably the same”
  - “If two people have similar friends, they are probably the same”
  - “If A=B and B=C, then A and C must also denote the same person”

\[
A \approx B \land B \approx C \Rightarrow A \approx C : \infty
\]
Challenges

- **Collective Classification**: labeling nodes in graph
  - irregular structure, not a chain, not a grid
  - Challenge: One large partially labeled graph

- **Link prediction**: predicting edges in graph
  - Dependencies among edges
  - Don’t want to reason about all possible edges
  - Challenge: scaling & extremely skewed probabilities

- **Entity resolution**: determine nodes that refer to the same entities in a graph
  - Dependencies between clusters
  - Challenge: enforcing constraints, e.g. transitive closure

*Key Idea: Predictions/Outputs depend on each other, joint reasoning is required!*
Probabilistic Soft Logic (PSL)

**Declarative language** based on logics to express collective probabilistic inference problems
- Predicate = relationship or property
- Atom = (continuous) random variable
- Rule = capture dependency or constraint
- Set = define aggregates

PSL Program = Rules + Input DB

Collective Classification

\[
\text{vote}(A,P) \land \text{friend}(B,A) \rightarrow \text{vote}(B,P) : 0.3
\]

\[
\text{vote}(A,P) \land \text{spouse}(B,A) \rightarrow \text{vote}(B,P) : 0.8
\]
PSL Foundations

- PSL makes large-scale reasoning scalable by mapping logical rules to convex functions

- Three principles justify this mapping:
  - LP programs for MAX SAT with approximation guarantees [Goemans and Williamson, ’94]
  - Pseudomarginal LP relaxations of Boolean Markov random fields [Wainwright, et al., ’02]
  - Łukasiewicz logic, a logic for reasoning about continuous values [Klir and Yuan, ‘95]
Hinge-loss Markov Random Fields

\[ P(Y \mid X) = \frac{1}{Z} \exp \left[ -\sum_{j=1}^{m} w_j \max\{\ell_j(Y, X), 0\}^{p_j} \right] \]

- Continuous variables in \([0, 1]\)
- Potentials are hinge-loss functions
- Subject to arbitrary linear constraints
- Log-concave!
PSL in a Slide

- MAP Inference in PSL translates into convex optimization problem -> **inference is really fast!**
- Inference further enhanced with state-of-the-art optimization and distributed processing paradigms such as ADMM & GraphLab -> **inference even faster!**
- **Outperforms discrete MRFs** in terms of speed, and (very) often accuracy
- **PSL is flexible:** Applied to image segmentation, activity recognition, stance-detection, sentiment analysis, document classification, drug target prediction, latent social groups and trust, engagement modeling, ontology alignment, and looking for more!
Discussion
Lesson: Make sure you are working on the right graph before performing analytics!

How do you get the right graph?

Use graph identification to infer it from the data!
Closing Comments

• Make sure you’re working on the right graph before performing analytics!
• Combining sampling (incomplete data) with inference
  – Active inference and surveying
  – Latent variable learning
• Important flipside to graph identification: **Privacy**
  – How to ensure that the graph can’t be re-identified?
• Research challenges and compelling applications abound!
Thank You!

Contact information:
getoor@ucsc.edu