JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Data Mining – Link Prediction

Andreas Karwath   Jörg Wicker

Johannes Gutenberg University Mainz

March 21, 2017

Outline

Acknowledgments

- Slides partially from
  - Lise Getoor
  - Ted Senator
  - Scott Macskassy
  - Foster Provost
  - Ramya Ramakrishnan
  - Jaideep Srivastava
  - Jean-Philippe Vert
  - George Karypis
  - Stefan Kramer

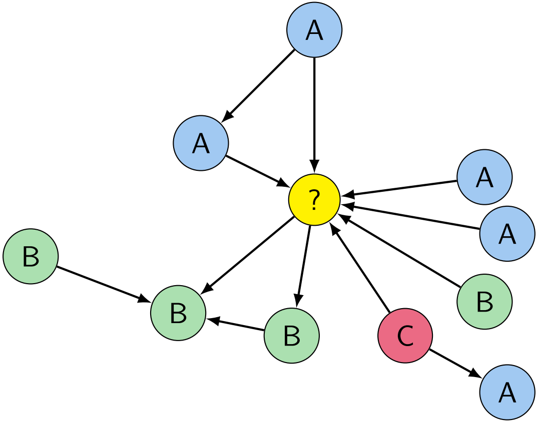
Section 1

Link-Based Object Classification

Link-Based Object Classification

- Predicting the category of an object based on its attributes and its links and attributes of linked objects
  - **web**: predict the category of a web page, based on words that occur on the page, links between pages, anchor text, html tags, etc.
  - **citation**: predict the topic of a paper, based on word occurrence, citations, co-citations
  - **epidemic**: predict disease type based on characteristics of the people

Link-Based Object Classification



Predicting the category of an object based on its attributes and its links and attributes of linked objects.

Link-Based Object Classification

- Variety of ways of describing link neighborhoods
  - mode, binary, count, in-links, out-links, ...
- Unlabeled data provide useful information
  - helps us infer object attribute distribution
  - *links between unlabeled data* allow us to make use of attributes of linked objects
  - *links between labeled data and unlabeled data* (training data and test data) help us make more accurate inferences
- Link-based classification challenges
  - feature construction
  - collective classification
  - use of labeled and unlabeled data

Major Distinctions

- Within-network classification
  - training entities are connected directly to entities whose classifications (labels) are to be estimated
- Across-network classification
  - learning from one network and applying the learned models to a separate, presumably similar network
- Iterative classification
  - feature set enhanced by features derived from the node's neighborhood
- Collective classification
  - treats the problem as a global optimization problem

Definition Within-Network Collective Inference

- **Given**: graph $G = (V, E, X, Y)$, where x_i is an attribute vector and y_i is a label variable for vertex v_i in V, and known values of y_i for a (training) subset of vertices V_1
- **Find**: (simultaneously) the values of y_i for the remaining (test) vertices, $V_2 = V \setminus V_1$, or a probability distribution over those values.

Iterative Classification

- Features describing vertex plus features derived from the neighborhood of the vertex (e.g., existence of neighbor of particular class)
- Only the *labeled data* is employed in learning, but the *structure of the network* is employed
- Use any propositional learning algorithm, apply classifier iteratively to the unlabeled vertices
- First approach due to (Chakrabarti *et al.*, 1998):
  - local *relaxation labeling algorithm* applied to subgraph around the vertex to be classified
  - combine local features and the labels of neighbors using a Naïve Bayes model (local features of neighbors harm performance)

Methods: Relaxation Labeling

- Technique from computer vision, also used for solving non-linear equations
- Given: set of $O = o_1, ..., o_n$ object features belonging to an object and a set of labels $L = l_1, ..., l_m$
- Iterative adjustment of probabilities
  - labeling process starts with an initial, and perhaps arbitrary, assignment of probabilities for each label for each feature
  - probabilities are transformed step by step according to some *relaxation schedule*: update of the probabilities taking into account the probabilities of labels for neighbouring features
  - repeated until method converges or stabilises (little or no change between successive steps): convergence not generally guaranteed

Relational Neighbor (RN)[1]

- Completes partially labeled graph by simply computing the weighted counts of each class among the neighbors
  - node priors set to relative class frequency in the training data
  - relational classification by a **weighted** average of the class probability estimates of the nodes neighbor
  - weights on edges set in a domain-dependent fashion
  - select class with maximum weighted count
- Relaxation labeling method similar to Chakrabarti *et al.*
- Simple model performing surprisingly well in practice

[1] Macskassy and Provost, 2007

Transductive Inference

- Introduced by Vapnik in the 1990s

- Reasoning from observed, specific (training) cases to *specific* (test) cases
  - in contrast, induction is reasoning from observed training cases to general rules, i.e., *a more general problem than transduction*
  - interesting in cases where the predictions of the transductive model are not achievable by any inductive model, due to transductive inference on different test sets producing mutually inconsistent predictions

- Hints about the distribution of instances
  - consider the case of two large clusters corresponding to two classes that cannot be clearly detected in a training set
  - closely related: semi-supervised learning, but different motivation

Section 2

Link Prediction

Link Prediction

- Predict whether a link exists between two entities, based on attributes and other observed links

- Application
  - **web**: predict if there will be a link between two page
  - **citation**: predicting if a paper will cite another paper
  - **epidemics**: predicting who a patients contacts are

- Methods
  - often viewed as a binary classification problem
  - local conditional probability model, based on structural and attribute features
  - difficulty: sparseness of existing links
  - collective prediction, e.g., Markov random field model

- *There exist different versions of link prediction*: ...

Different Versions of Link Prediction

1. Given a social network at time t_i predict the social link between actors at time $t_i + 1$
2. Given a social network with an incomplete set of social links between a complete set of actors, predict the unobserved social links
3. Given information about actors, predict the social link between them

- Main approaches: fit the social network on a model and then use the model for prediction
- Other approaches specifically target the link prediction problem

Within-Network vs. Across-Network

- **Within-network link prediction**
  - the links to be predicted are from the same graph (network) as the training data (edges among vertices)
- **Across-network link prediction**
  - as before, learning from one network and applying the learned models to a separate, presumably similar network

Predicting Link Existence

- Predicting whether a link exists between two objects
  - **web**: predict whether there will be a link between two pages
  - **citation**: predicting whether a paper will cite another paper
  - **epidemics**: predicting who a patients contacts are

Link Cardinality Estimation

- Predicting the number of links to an object
  - **web**: predict the authoritativeness of a page based on the number of in-links; identifying hubs based on the number of out-links
  - **citation**: predicting the impact of a paper based on the number of citations
  - **epidemics**: predicting the infectiousness of a disease based on the number of people diagnosed

Link Type

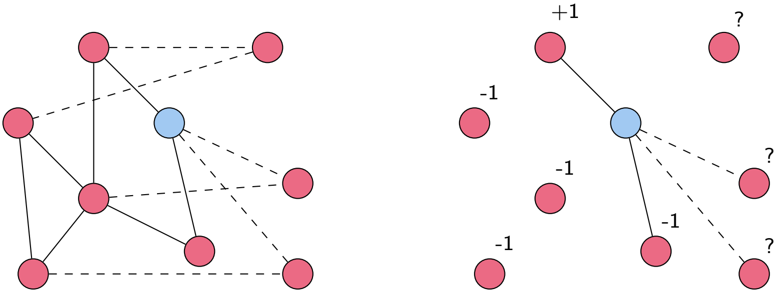
- Predicting type or purpose of link
  - **web**: predict advertising link or navigational link
  - **citation**: predicting whether co-author is also an advisor; predict an advisor-advisee relationship
  - **epidemics**: predicting whether contact is familial, co-worker or acquaintance

Link Prediction: Pairwise SVM [2]

- Classifying pairs of genes as "interacting" or "not interacting", using the training graph as a set of training edges with known labels
- Here: labeling of edges, not of nodes
- Definition of a kernel between edges using the kernel between individual genes as plug-in
- Training of all pairs of distinct vertices in the training set, thus $n(n-1)/2$ training points (local approach: n)
  - due to (roughly quadratic) SVM complexity and memory requirements, scales at least in $O(n^4)$
  - practically: random subsampling of n negative pairs to obtain a balanced training set of size $2n$

[2] Ben-Hur and Noble, 2005

(Within Network) Link Prediction as Local Classification



Within-network link prediction task from the left-hand side is transformed into a "local" classification task on the right-hand side.

Local Classification for Link Prediction

- Problem considered: predict edges between a vertex in training set V_1 and a vertex in test set $V_2 = V \setminus V_1$
- Addresses tasks like finding new genes regulated by a transcription factor or finding missing enzymes in a metabolic pathway
- Solution: train one local classifier for each vertex v in the training set V_1
  - assumption: all edges between two vertices in V_1 are observed
  - take every other vertex u in V_1 and assign label $+1$ if there is an edge from v to u and -1 otherwise
  - apply any machine learning algorithm for classification to learn a function assigning $+1$ or -1 to any new vertex (in test set V_2)

Section 3

Summary

Summary Graph Mining

- Analysis of networks and sets of graphs
- Clustering commonly used to detect sub-groups (sub-networks)
- Pattern mining for sets of graphs
- Object classification and link prediction

Outlook: Relational Learning

- Can everything be encoded in a graph?
- What about the following artificial problem?