Network Inference & Link Prediction

Network Application Diagnostics
B2M32DSA

Radek Mařík

Czech Technical University
Faculty of Electrical Engineering
Department of Telecommunication Engineering
Prague CZ

October 31, 2019

Outline

Outline

Classification Predictions [?]

- **The expectation:** the terms *positive* and *negative* refer to the classifier's prediction.
- **The observation:** the terms *true* and *false* refer to whether that prediction corresponds to the external judgment.
- The confusion matrix (CZ kontigenční tabulka)

		Predicted / Classified	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

- **TN / True Negative**
  - the real case is negative
  - and predicted negative
- **FP / False Positive**
  - the real case is negative
  - but predicted positive
  - Type I error
- **TP / True Positive**
  - the real case is positive
  - and predicted as positive
- **FN / False Negative**
  - the real case is positive
  - but predicted negative
  - Type II error

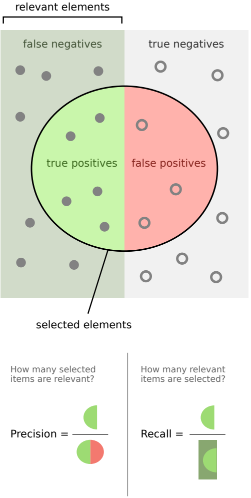
Precision and Recall [?, ?]

- **Precision**
  - the probability that a (randomly selected) retrieved document is relevant.
  - the probability that a (randomly selected) object is correctly classified.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**
  - the probability that a (randomly selected) relevant document is retrieved in a search.
  - the probability that a (randomly selected) class object is correctly classified.

$$\text{Recall} = \frac{TP}{TP + FN}$$

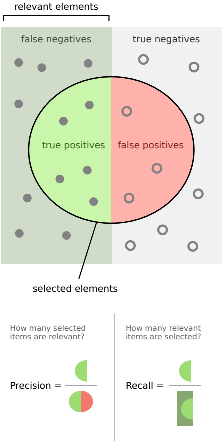


Accuracy and F1-Measure [?, ?]

- **Accuracy**
  - the proportion of true results (both true positives and true negatives) among the total number of cases examined.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

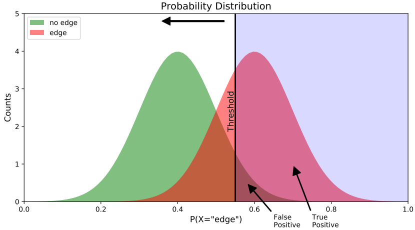
- **$F1$-Measure**
  - the harmonic mean of precision and recall.
  - an $F1$ score reaches its best value at 1 (perfect precision and recall) and worst at 0.

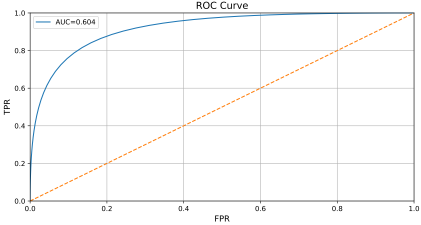


$$\text{F1} = \left(\frac{\text{Precision}^{-1} + \text{Recall}^{-1}}{2}\right)^{-1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

ROC Curves [?]

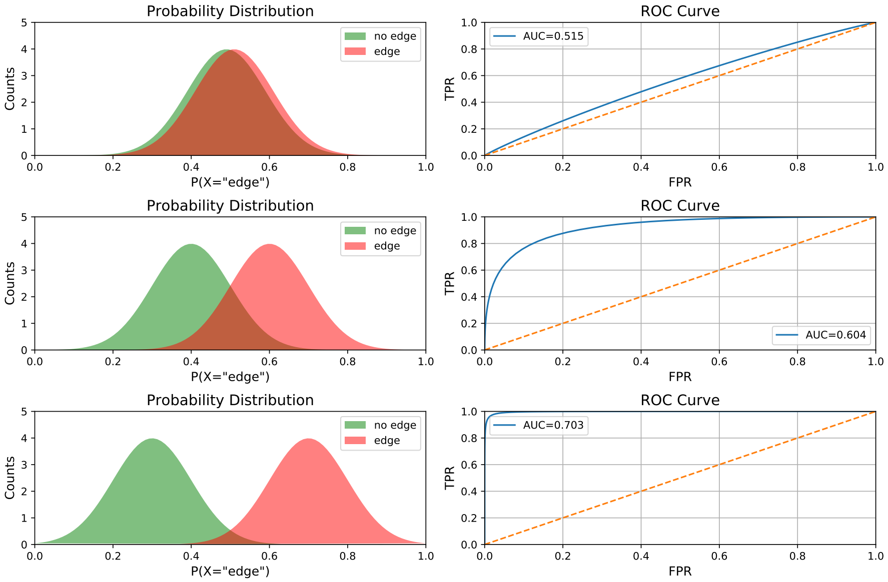
- Plotting the true positive rate (TPR) against the false positive rate (FPR).
- Dealing with heavy class imbalance.
- The model performance is measured by the area under the ROC curve (AUC).
- The best possible AUC is 1.
- The worst AUC is 0.5 (the 45 degrees random line).
- If the AUC is below 0.5, do the exact opposite of what the model recommends.





$$\text{TPR} = \frac{TP}{TP+FN}, \quad \text{FPR} = \frac{FP}{FP+TN}$$

ROC Performance Assessment [?]



Outline

Graph Notation [?]

- Let $G(V, E)$ be an undirected random network graph.
- $V^{(2)}$ is the set of distinct unordered pairs of vertices.
- E is the set of edges in G.
- $N_v = |V|$ is the number of vertices.
- $N_e = |E|$ is the number of edges.
- $V^{(2)} \setminus E$ is the set of non-edges in G.
- $V^{(2)}_{obs}$ is the observed presence or absence of edges.
- $V^{(2)}_{miss} = V^{(2)} \setminus V^{(2)}_{obs}$ is the set of edges for which observations are missing.
- Sparse graph: $|E| \ll |V|^2$
- The set $\mathcal{N}(u)$ of neighbors of $u \in V$ in $G(V, E)$:

$$\mathcal{N}(u) = \{v | v \in V, e = uv \in E\}$$

Outline

Network Topology Inference [?]

- What should constitute a vertex and an edge is determined by user-specified decisions and rules.
- Such a network graph construction lacks an element of validation.
  - if the network representation is "accurate",
    - i.e. capturing some well-defined but unobservable relational structure.
  - What accuracy can be expected given the available measurements?
  - Are there other similar representations with about the same accuracy?
  - How is the representation robust to changes in the measurements?
  - How is the representation usefull for other purposes?

Network Topology Inference Problem

- Given a set of measurements from a system of interest, e.g.
  - vertex attributes $\boldsymbol{x} = (x_1, \ldots, x_{N_\mathcal{V}})$
  - binary indicators $\boldsymbol{y} = [y_{ij}]$ of certain edges.
- and given a collection $\mathscr{G}$ of potential graphs G,
- select an appropriate member of $\mathscr{G}$ that best captures the underlying state of the system.

Network Topology Inference [?]

- What should constitute a vertex and an edge is determined by user-specified decisions and rules.
- Such a network graph construction lacks an element of validation.
  - if the network representation is "accurate",
    - i.e. capturing some well-defined but unobservable relational structure.
  - What accuracy can be expected given the available measurements?
  - Are there other similar representations with about the same accuracy?
  - How is the representation robust to changes in the measurements?
  - How is the representation usefull for other purposes?

Network Topology Inference Problem

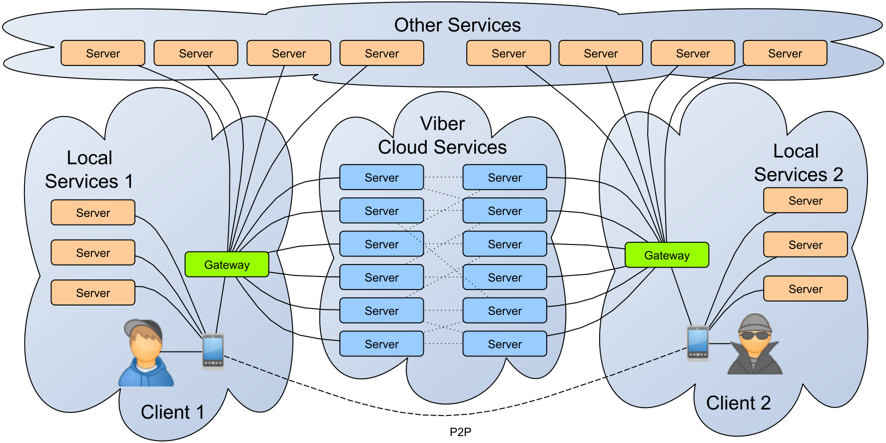
- Given a set of measurements from a system of interest, e.g.
  - vertex attributes $\boldsymbol{x} = (x_1, \ldots, x_{N_\mathcal{V}})$
  - binary indicators $\boldsymbol{y} = [y_{ij}]$ of certain edges.
- and given a collection $\mathscr{G}$ of potential graphs G,
- select an appropriate member of $\mathscr{G}$ that best captures the underlying state of the system.

Network Inference Problems [?]

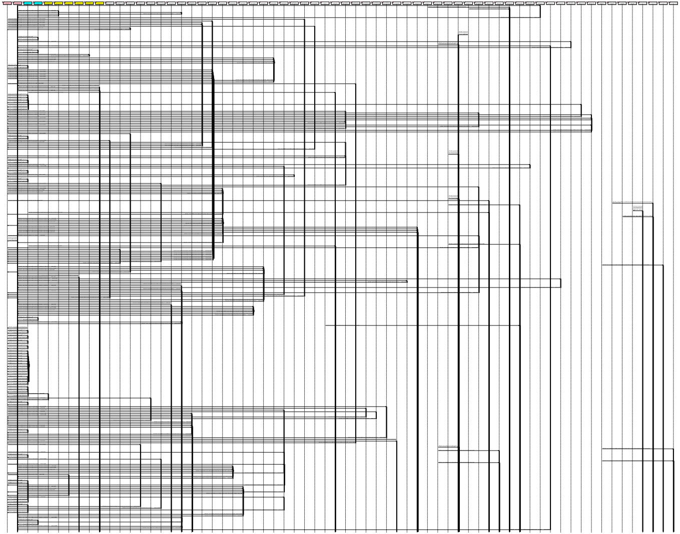
- **Link Prediction** . . . inferring whether or not a pair of vertices does or does not have an edge between them
  - using measurements that include a subset of vertex pairs whose edge/non-edge status is already observed.
  - knowledge of all of the vertices.
  - the status of some of the edges/non-edges
- **Association Graph Inference** . . . the relation defining edges is itself unobserved and must be inferred from measurements reflecting these characteristics.
  - no knowledge of edge status anywhere in the network graph,
  - relevant measurements at all of the vertices are assumed.
- **Tomographic Network Inference** . . . the measurements are available only at vertices that are somehow at the perimeter of the network.
  - measurements at only a particular subset of vertices are known.

Outline

Exemplar (Viber) Environment [?]



Example Capture Characteristics [?]

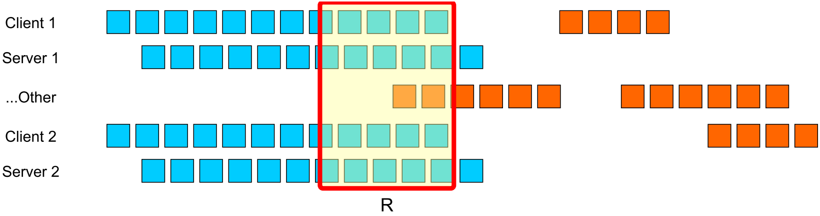


- 138882 PCAP blocks
- 1788 transport sessions
- 2 clients
- 22 viber.com servers
- 150 peers of 2 clients
- 5660 possible concurrent sessions
- How to analyze?

Concurrent Communication Detection [?]

Selection of IP nodes

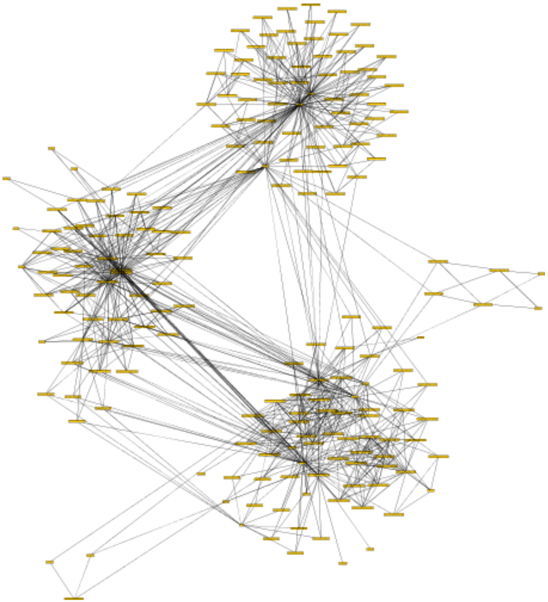
- *viber.com* servers → viber clients → other Viber servers
- classified based on entropy based characteristics of TCP/IP distributions



$$s(a,b) = \frac{\sum_{\forall i,j: t_a[i]-t_b[j]<R} R/(t_a[i]-t_b[j])}{\sum_{\forall i,j: t_a[i]-t_b[j]<R} 1}$$

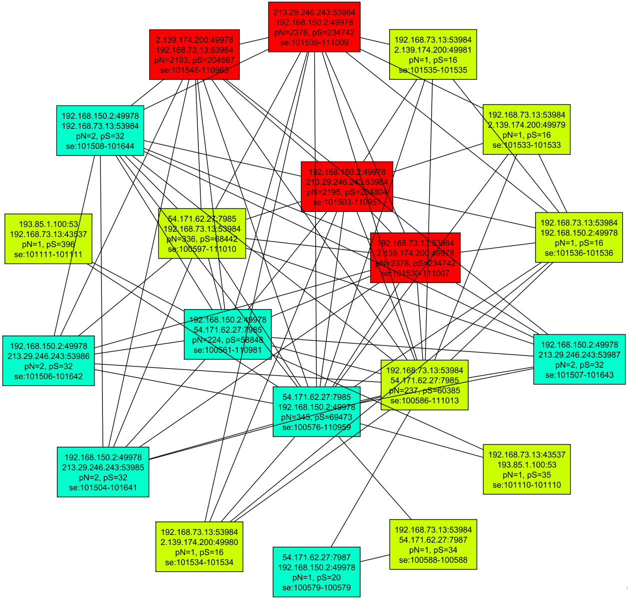
In our experiments: $R = 50ms, \quad s(a,b) > 0.001$

UDP packet sequence concurrency as a complex network [?]



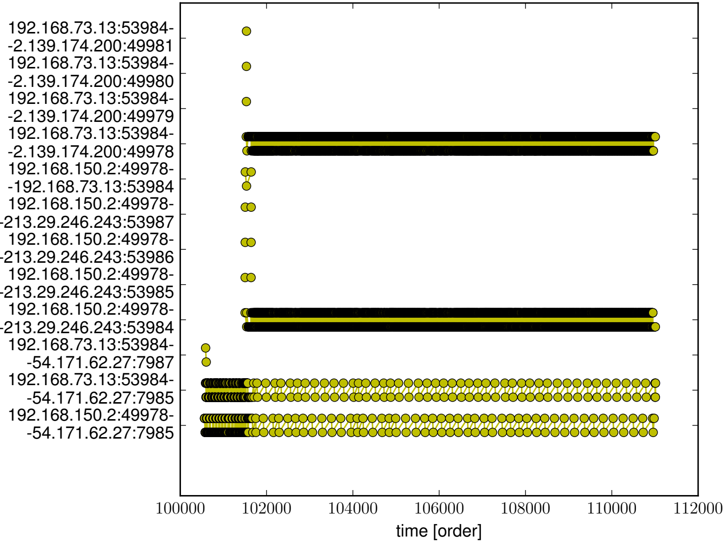
- captures with two clients
- *"communities"* of concurrent sessions
- some clusters related to only one client
- interesting clusters consist of nodes of both clients

UDP packet sequence concurrency network component [?]



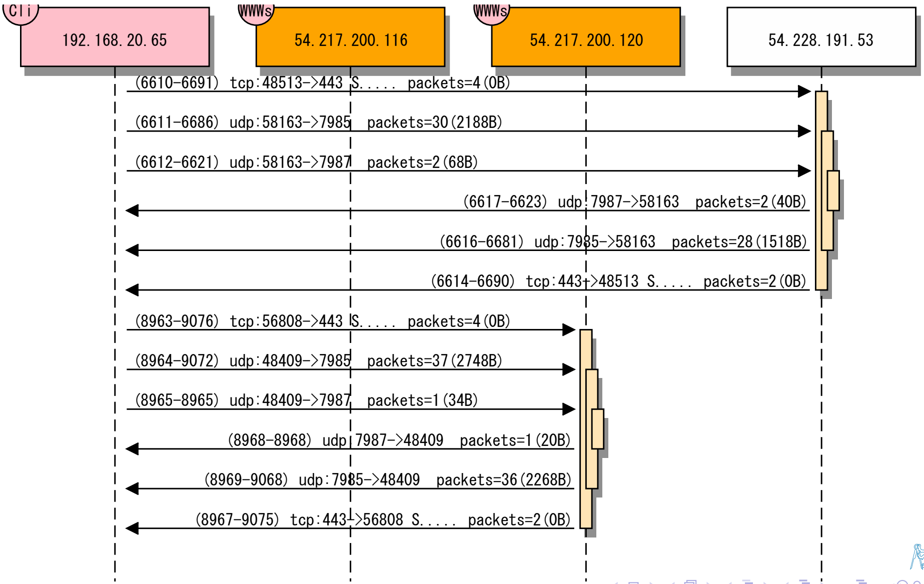
- restricted on one of the components
- two *Viber* clients
- 192.168.73.13
- 192.168.150.2

UDP packet sequence concurrency - packet timing [?]

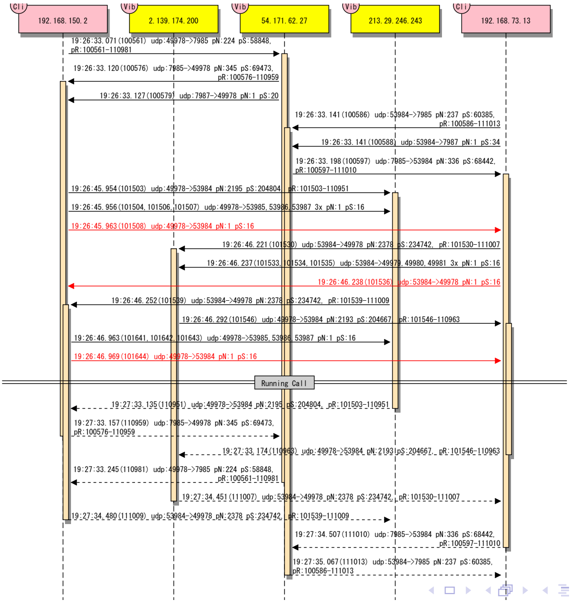


- signals
- calls
- keep-alive packets
- direct client to client packets

Message Sending [?]

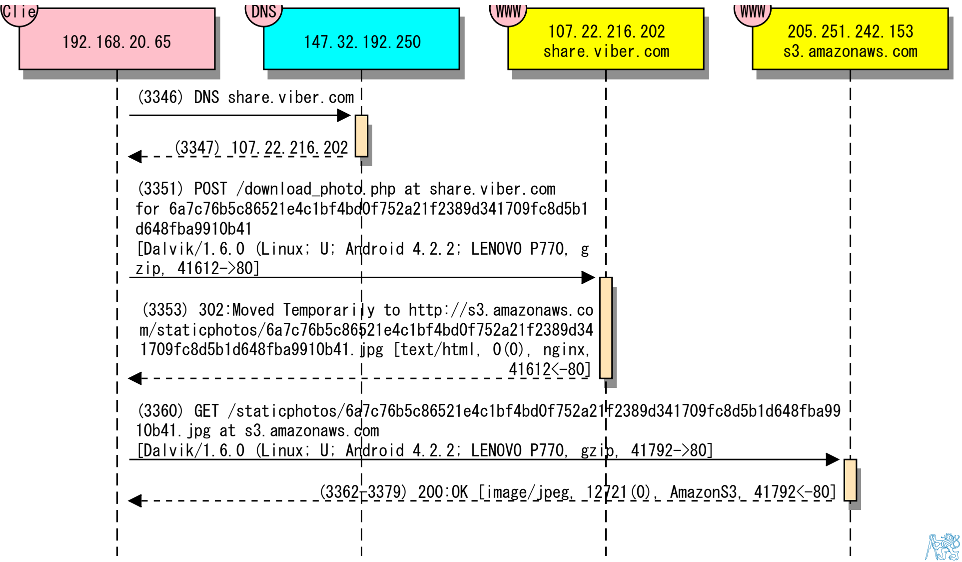


Voice Call [?]



Cli
192. 168. 150. 2
Vib
2. 139. 174. 200
Vib
54. 171. 62. 27
Vib
213. 29. 246. 243
Cli
192. 168. 73. 13
19:26:33. 071 (100561) udp:49978->7985 pN:224 pS:58848, pR:100561-110981
19:26:33. 120 (100576) udp:7985->49978 pN:345 pS:69473, pR:100576-110959
19:26:33. 127 (100579) udp:7987->49978 pN:1 pS:20
19:26:33. 141 (100586) udp:53984->7985 pN:237 pS:60385, pR:100586-111013
19:26:33. 141 (100588) udp:53984->7987 pN:1 pS:34
19:26:33. 198 (100597) udp:7985->53984 pN:336 pS:68442, pR:100597-111010
19:26:45. 954 (101503) udp:49978->53984 pN:2195 pS:204804, pR:101503-110951
19:26:45. 956 (101504, 101506, 101507) udp:49978->53985, 53986, 53987 3x pN:1 pS:16
19:26:45. 963 (101508) udp:49978->53984 pN:1 pS:16
19:26:46. 221 (101530) udp:53984->49978 pN:2378 pS:234742, pR:101530-111007
19:26:46. 237 (101533, 101534, 101535) udp:53984->49979, 49980, 49981 3x pN:1 pS:16
19:26:46. 238 (101536) udp:53984->49978 pN:1 pS:16
19:26:46. 252 (101539) udp:53984->49978 pN:2378 pS:234742, pR:101539-111009
19:26:46. 292 (101546) udp:49978->53984 pN:2193 pS:204667, pR:101546-110963
19:26:46. 963 (101641, 101642, 101643) udp:49978->53985, 53986, 53987 pN:1 pS:16
19:26:46. 969 (101644) udp:49978->53984 pN:1 pS:16
Running Call
19:27:33. 135 (110951) udp:49978->53984 pN:2195 pS:204804, pR:101503-110951
19:27:33. 157 (110959) udp:7985->49978 pN:345 pS:69473, pR:100576-110959
19:27:33. 174 (110963) udp:49978->53984 pN:2193 pS:204667, pR:101546-110963
19:27:33. 245 (110981) udp:49978->7985 pN:224 pS:58848, pR:100561-110981
19:27:34. 451 (111007) udp:53984->49978 pN:2378 pS:234742, pR:101530-111007
19:27:34. 480 (111009) udp:53984->49978 pN:2378 pS:234742, pR:101539-111009
19:27:34. 507 (111010) udp:7985->53984 pN:336 pS:68442, pR:100597-111010
19:27:35. 067 (111013) udp:53984->7985 pN:237 pS:60385, pR:100586-111013

Security/Privacy Assessment - Contact Picture Transfer



Outline

1. Preliminary Tools
   - Classification Evaluation
   - Network Terminology
2. Network Inference
   - Introduction
   - Network Inference Example - Viber
3. Link Prediction
   - Introduction
   - Scoring Functions

Link Prediction Motivation [?]

- Networks are highly dynamic objects
  - they grow and change . . . e.g. by adding new edges
- Network evolution models
  - To what extent can the evolution of a network be modeled using features *intrinsic to the network itself*?
  - A number of proximity measures lead to predictions that outperform a random prediction by factors 40 to 50.
  - The network topology might contain latent information from which one can infer future edges (interactions).
- Recovery of a hidden/latent informal network by observing the official observable network part.

Link Prediction Definition [?, ?]

- **Link Prediction**
  - A network is changing over time.
  - Given a snapshot of a network at time t,
  - predict edges added in the interval (t, t')
- **Link Completion** (missing links identification).
  - Given a network,
  - infer links that are consistent with the structure, but missing.
  - *Find unobserved edges*
- **Link Reliability**
  - Estimate the reliability of given links in the graph.

What can be predicted?

- Link existence,
- Link weight,
- Link type,
- Link cardinality.

Link Prediction [?, ?]

- Given a graph $G(V, E)$
- The number of *missing edges*: $|V|(|V|-1)/2-|E|$
- Probability of a correct random guess $O(\frac{1}{|v|^2})$
  - in sparse graphs ($|E| \ll |V|^2$)
- Each edge $e \in E$ represents an interaction between its endpoints at a particular time $t(e)$.
- Multiple interactions are represented by parallel edges with different time-stamps.
- $G[t, t']$ is the subgraph of G restricted to edges with time-stamps between t and t', $t < t'$.

Link prediction phases

1. **Learning:** *training* interval $[t_0, t'_0] \ldots G[t_0, t'_0]$
2. **Prediction:** *testing* interval $[t_1, t'_1] \ldots G[t_1, t'_1]$

Link Prediction [?, ?]

- Given a graph $G(V, E)$
- The number of *missing edges*: $|V|(|V|-1)/2 - |E|$
- Probability of a correct random guess $O(\frac{1}{|v|^2})$
  - in sparse graphs ($|E| \ll |V|^2$)
- Each edge $e \in E$ represents an interaction between its endpoints at a particular time $t(e)$.
- Multiple interactions are represented by parallel edges with different time-stamps.
- $G[t, t']$ is the subgraph of G restricted to edges with time-stamps between t and t', $t < t'$.

Link prediction phases

1. **Learning:** *training* interval $[t_0, t'_0] \ldots G[t_0, t'_0]$
2. **Prediction:** *testing* interval $[t_1, t'_1] \ldots G[t_1, t'_1]$

Scoring Algorithm [?, ?]

- Proximity/Similarity score $c(v_1, v_2)$ ... it is assumed that the higher the score the higher the probability that the vertexes v_1 and v_2 interact and they are linked by the edge.

Link prediction by proximity scoring

1. Compute proximity/similarity score $c(v_1, v_2)$ for each pair of nodes.
2. Sort all pairs by the decreasing score.
3. Select top pairs as new links
   - n pairs
   - pairs above a treshold.

Outline

Scoring Functions - Neighborhood Based [?, ?]

Local neighborhoods of v_i and v_j

- Number of **common neighbors**:
  - based on the idea that links are formed between nodes who share many common neighbors

$$c^{CN}(v_i, v_j) = |\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|$$

- **Jaccard's coefficient**:
  - a measure of the likelihood that a neighbor of v_i is a neighbor of v_j and vice versa.

$$c^{JA}(v_i, v_j) = \frac{|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|}{|\mathcal{N}(v_i) \cup \mathcal{N}(v_j)|}$$

- **Adamic/Adar**:
  - The larger weight is assigned to common neighbors v of v_i and v_j which themselves have few neighbors $\log|\mathcal{N}(v)|$,
  - i.e. v_i and v_j are "related" because of the rarer neighbor v.

$$c^{AA}(v_i, v_j) = \sum_{v \in \mathcal{N}(v_i) \cap \mathcal{N}(v_j)} \frac{1}{\log|\mathcal{N}(v)|}$$

Scoring Functions - Neighborhood Based [?]

Preferential attachment:

- A new node is attached to a network node u that has a higher probability of fitness expressed as the size of its neighborhood $|\mathcal{N}(u)$.

$$c(v_i, v_j) = |\mathcal{N}(v_i)||\mathcal{N}(v_j)|$$

Scoring Functions - Path Based [?]

Paths and ensembles of paths between v_i and v_j

- Shortest path:

$$-\min_s\{path_{ij}^s > 0\}$$

- Katz score:

$$\sum_{\ell=1}^{\infty} \beta^\ell |paths^{(\ell)}(v_i, v_j)| = \sum_{\ell=1}^{\infty} (\beta \boldsymbol{A})_{ij}^\ell = (\boldsymbol{I} - \beta \boldsymbol{A})^{-1} - \boldsymbol{I}$$

- Personalized (rooted) PageRank:

$$PR = \alpha(\boldsymbol{D}^{-1}\boldsymbol{A})^T PR + (1 - \alpha)$$

Scoring Functions - Path Based [?]

Local neighborhoods of v_i and v_j

- Number of common neighbors

$$|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|$$

- Jaccard's coefficient

$$\frac{|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|}{\mathcal{N}(v_i) \cup \mathcal{N}(v_j)}$$

- Adamic/Adar:

$$\sum_{v \in \mathcal{N}(v_i) \cap \mathcal{N}(v_j)} \frac{1}{\log |\mathcal{N}(v)|}$$

Summary

- Network inference problem
- Network inference case study
- Link prediction problem
- Link prediction scoring functions

Competencies

- Define precision, recall, accuracy, and $F1$-measure used in classification evaluation.
- How ROC curves are used in classication problems?
- Define the network inference problem and its subproblems.
- How is it possible to detect packet sequence concurrency?
- Define the link prediction problem is its subproblems.
- Define typical scoring functions used in the link prediction problem.

References I

[LK03] David Liben-Nowell and Jon M. Kleinberg. The link prediction problem for social networks. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003*, pages 556–559. ACM, 2003.

[MBKK15] Radek Mařík, Pavel Bezpalec, Jan Kučerák, and Lukáš Kencl. Revealing viber communication patterns to assess protocol vulnerability. In *2015 International Conference on Computing and Network Communications (CoCoNet). Leonia, NJ 07605: EDAS Conference Services*, pages 502–510, 2015.

[Wik19a] Wikipedia. Precision and recall, 2019.

[Wik19b] Wikipedia. Receiver operating characteristic, 2019.