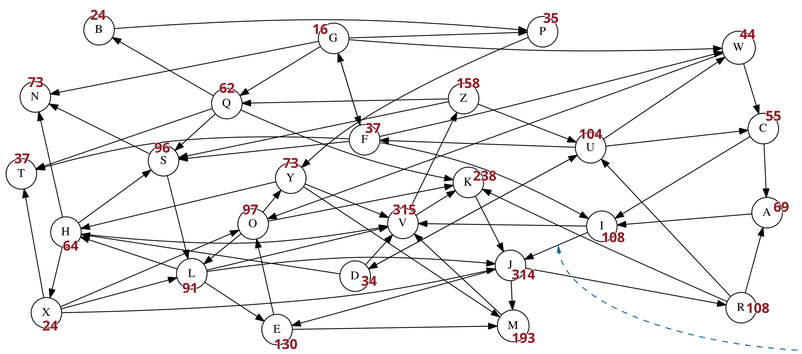
School of Computer and Communication Sciences
Laboratory for Communications and Applications 4

Lucas Maystre and Matthias Grossglauser

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ChoiceRank: Identifying Preferences from Node Traffic in Networks

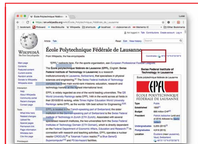


Suppose that we observe

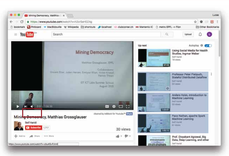
Network structure + **Marginal traffic data**

Is it possible to understand **how users navigate along the edges of the network?**

"how many users went along edge I to J ?"



Given a **hyperlink graph** and **page traffic**: How often is each link clicked?



Given **recommendations** and **view counts**: How useful is each recommendation?

Given a **bikeshare network** and **# bikes**: Where do people go?

Formulation as a Maximum-Likelihood Problem

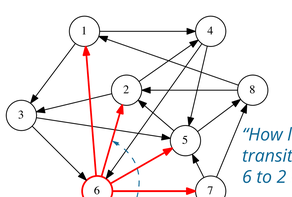
number of nodes: n
number of edges: $O(n^2)$ → problem is underdetermined

Assumption on the transition probabilities:

$$p_{ij} \propto \underbrace{\text{similarity}(i,j)}_{w_{ij}} \cdot \underbrace{\text{score}(j)}_{\lambda_j}$$

In this poster: $w_{ij} = 1$

$$p_{ij} = \frac{\lambda_j}{\sum_{k \in N_i^+} \lambda_k}$$



"How likely is a transition from 6 to 2 ?"

$$p_{62} = \frac{\lambda_2}{\lambda_1 + \lambda_2 + \lambda_5 + \lambda_7}$$

Goal: find "good" parameters $\boldsymbol{\lambda}$.

The likelihood of parameters $\boldsymbol{\lambda}$ is given by

$$\ell(\boldsymbol{\lambda}) = \sum_{(i,j) \in E} c_{ij} \log \frac{\lambda_j}{\sum_{k \in N_i^+} \lambda_k}$$

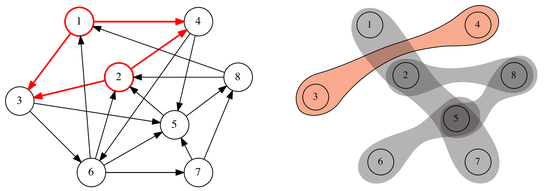
$$= \sum_{i=1}^{n} \left[c_i^- \log \lambda_i - c_i^+ \log \sum_{k \in N_i^+} \lambda_k \right]$$

incoming traffic: $\sum_{j \in N_i^-} c_{ji}$

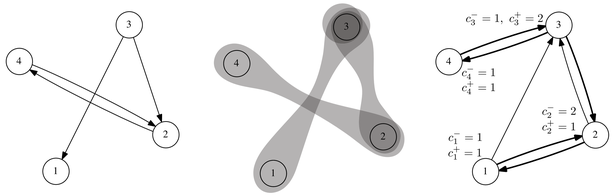
outgoing traffic: $\sum_{j \in N_i^+} c_{ij}$

Marginal node traffic is a **sufficient statistic** for the parameters.

Maximum-Likelihood Estimate: Potential Issues



For some network structures, the model is **not identifiable**.



If the traffic is sparse, the likelihood function might have **no maximum**.

The ChoiceRank Algorithm

Regularizer: ensures that problem is **well-posed**

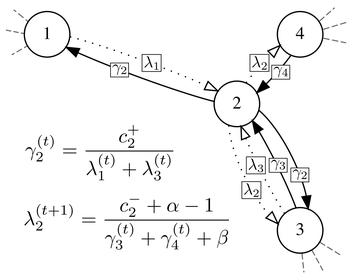
$$\log p(\boldsymbol{\lambda} \mid \mathcal{D}) = \sum_{i=1}^{n} \left[c_i^- \log \lambda_i - c_i^+ \log \sum_{k \in N_i^+} \lambda_k \right] + \sum_{i=1}^{n} (\alpha - 1) \log \lambda_i - \beta \lambda_i$$

≈ independent Gamma(α, β) prior

A **simple**, iterative update leads to the max:

$$\lambda_i^{(t+1)} = \frac{c_i^- + \alpha - 1}{\sum_{j \in N_i^-} \gamma_j^{(t)} + \beta}, \quad \gamma_j^{(t)} = \frac{c_j^+}{\sum_{k \in N_j^+} \lambda_k^{(t)}}$$

Each iteration requires only **2 passes over the edges**.



$$\gamma_2^{(t)} = \frac{c_2^+}{\lambda_1^{(t)} + \lambda_3^{(t)}}$$

$$\lambda_2^{(t+1)} = \frac{c_2^- + \alpha - 1}{\gamma_3^{(t)} + \gamma_4^{(t)} + \beta}$$

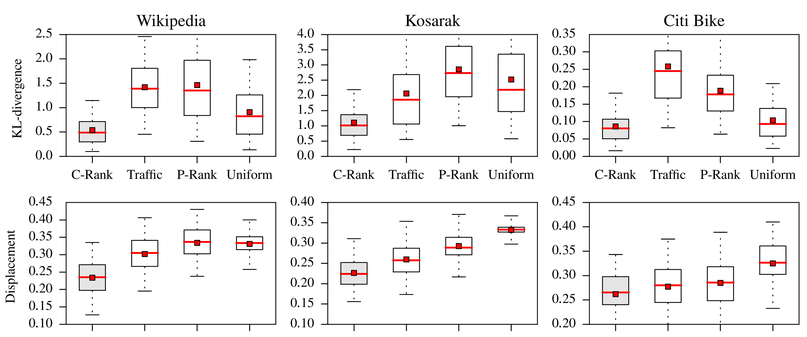
Experimental Results

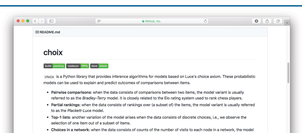
Network: hyperlink graph
Data: clickstream

Network: NYC bikeshare network
Data: trips between stations

$$D_{\mathrm{KL}}(\boldsymbol{p}_i^\star, \boldsymbol{q}_i) = \sum_{j \in N_i^+} p_{ij}^\star \log \frac{p_{ij}^\star}{q_{ij}}$$

$$D_{\mathrm{FR}}(\boldsymbol{p}_i^\star, \boldsymbol{q}_i) = \frac{1}{|N_i^+|^2} \sum_{j \in N_i^+} |\sigma_i^\star(j) - \hat{\sigma}_i(j)|$$



choix

Python implementation and tutorial **notebook** available at `github.com/lucasmaystre/choix`