ChoiceRank

Identifying Preferences from Node Traffic in Networks

Lucas Maystre, Matthias Grossglauser
School of Computer and Communication Sciences, EPFL

ICML — August 8th, 2017
Motivating Example
Problem Statement

Explain how users navigate along edges... ...given network structure and marginal traffic.
Choice Model

Underconstrained problem
→ “low-rank” parametrization of $p_{ij}$.

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

Consistent with Luce's choice axiom.
Probability of choosing $i$ over $j$ does not depend on the other alternatives.

[Luce 1959]
Prior Work

Inverting a Steady-State
[Kumar et al. WSDM 2015]

Random-walk framework

Given:
- directed graph $G = (V, E)$
- model for transitions
- stationary distribution $\pi$

Find matrix $P$ such that
- $\pi = \pi P$
- $p_{ij} = 0$ if no edge

Our work

We merely assume discrete choices on a network.

works with:
- **finite** traffic
- **arbitrary** network structure
Marginal Traffic is Sufficient

Given network structure + marginal traffic, find “good” parameters $\lambda$.

Pretend that we can observe all transitions $\mathcal{D} = \{c_{ij} \mid (i, j) \in E\}$

$$
\ell(\lambda; \mathcal{D}) = \sum_{(i,j)\in E} c_{ij} \left[ \log \lambda_j - \log \sum_{k \in N_i^+} \lambda_k \right]
$$

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

Marginal traffic $\{(c_i^+, c_i^-) \mid i \in V\}$ is a minimally sufficient statistic
Robust Inference

**ML estimate is often ill-defined** because of graph structure or data sparsity. → embed in a *Bayesian setting* by postulating a prior on $\lambda_i$.

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

$$+ \sum_{i=1}^{n} \left[ (\alpha - 1) \log \lambda_i - \beta \lambda_i \right]$$

**Theorem:** if $\alpha > 1$ and $\beta > 0$, there is always a unique maximum
ChoiceRank Algorithm

We maximize the log-posterior using the **MM algorithm**. [Hunter 2004]

\[
\lambda_i^{(t+1)} = \frac{c_i^-}{\sum_{j \in N_i^-} \gamma_j^{(t)}}, \quad \text{where} \quad \gamma_j^{(t)} = \frac{c_j^+}{\sum_{k \in N_j^+} \lambda_k^{(t)}}
\]

**Scales well to large graphs.** Tested on Common Crawl hyperlink graph:

- **3.5 B** nodes, **128 B** edges
- Takes **20 min / iteration** on a recent machine

One iteration requires two passes over the edges
Experimental Results

**English Wikipedia traffic** — 2 M nodes, 13 M edges, 1.2 B transitions.

How well do we recover the transition probabilities?
Using ChoiceRank to understand network traffic

1. Generating sample data

First, we will generate sample data. This includes:
1. generating a network,
2. generating a parameter for each node of the network,
3. generating samples of choices in the network.

```python
# 1. Generate a network.
graph = nx.random_regular_graph(p_edge, n)  # p_edge = 0.3

# 2. Generate a parameter for each node.
params = -np.random.rand(n)  # Each node is assigned a private parameter

# 3. Generate samples of choices in the network.
choices = [np.random.choice(graph.neighbors(node), size=n_samples, p=softmax(params), replace=True)]
```

The network looks as follows:

![Network Diagram]

Code & Examples

[github.com/lucasmaystre/choix](https://github.com/lucasmaystre/choix)
ChoiceRank vs. PageRank

ChoiceRank

• Given a **network** and **marginal traffic**, find **transition probabilities**.

• Assumption: transitions follow **Luce's choice axiom**.

• ChoiceRank score corresponds to a page's **utility**.

PageRank

• Given a **network**, find **steady-state traffic**.

• Assumption: transitions are **uniformly random** over neighbors.

• PageRank score corresponds to a page's **popularity**.
Issues with ML estimate 1
Issues with ML estimate 2

c_3^- = 1, \ c_3^+ = 2

c_4^- = 1
\ c_4^+ = 1

c_2^- = 2
\ c_2^+ = 1

c_1^- = 1
\ c_1^+ = 1
NYC Bike Sharing Data

Applications beyond clickstream data — e.g., mobility networks.