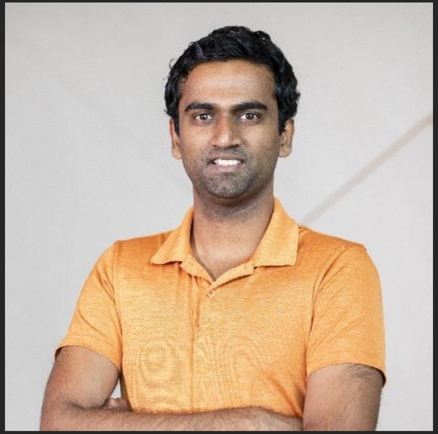
Statistical Estimation from Dependent Data

Yuval Dagan
MIT



Vardis Kandiros
(MIT)

Nishanth Dikkala
(Google Research)

Surbhi Goel
(Microsoft Research)

Constantinos Daskalakis
(MIT)

Data is dependent!

- Social networks:
  - Friends influence each other
  - Indirect effects can cause global dependencies
- Temporal data
  - Past influences future
- Many other examples!

criminal activity [Glaeser et al'96]
welfare participation [Bertrand et al'00]
school achievement [Sacerdote'01]
participation in Retirement Plans [Duflo-Saez'03]
obesity [Trogdon et al'08, Christakis-Fowler'13]

Econometrics: Disentagling individual from network effects [Manski'93],[Bramoulle-Djebbari-Fortin'09]
Microeconomics: Behavior/Opinion dynamics [Montanari-Saberi'10]
Meteorological and geographical data

Only one joint sample is available!

Data: $(x_1, y_1) \cdots (x_n, y_n)$

- $x_i \in X$: feature vector e.g. of student i
- $y_i \in \{-1,1\}$: label e.g. do they drink?

Claim: This is one big dependent sample!

- Rather than n i.i.d. samples

Abstract model

Distribution:

Individual effects

Network effects

$$\Pr_J[y_1 \cdots y_n \mid x_1 \cdots x_n; \theta, \beta] \propto \prod_{i=1}^{n} P_\theta(y_i \mid x_i) \cdot e^{\beta \cdot \sum_{i,j} J_{ij} y_i y_j}$$

Learnable parameters:

- $P_\theta(y_i \mid x_i)$: Individual effect (likelihood of label ignoring dependencies)
- $\beta \geq 0$: network effect (strength of dependencies)

Known parameter:

- $J \in \mathbb{R}^{n \times n}$: Weighted adjacency matrix

Result: efficient learning algorithm

Reminder: $\Pr_J[y_1 \cdots y_n | x_1 \cdots x_n; \theta, \beta] \propto \prod_{i=1}^n P_\theta(y_i \mid x_i) \cdot e^{\beta \cdot \sum_{i,j} J_{ij} y_i y_j}$

Logistic regression with network effect:

$$P_\theta(y_i|x_i) = \sigma(y_i\langle\theta, x_i\rangle) = \frac{e^{y_i\langle\theta,x_i\rangle}}{e^{\langle\theta,x_i\rangle} + e^{-\langle\theta,x_i\rangle}}$$

Error bound: (under some assumptions)

$$\left\|\hat{\theta} - \theta^*\right\|_2 \le \tilde{O}\left(\sqrt{d/n}\right), \qquad \left|\hat{\beta} - \beta^*\right| \le \tilde{O}\left(\sqrt{d/\|J\|_\infty}\right)$$

Special case studied by [Daskalakis, Dikkala, Panageas '19]

Abstract function-classes (e.g. Neural networks)

Proof Idea – via the Ising model (MRF)

Substituting $P_{\vec{h}}(\, y_i \mid x_i \,) \propto e^{h_i y_i}$, gives an **Ising model** [DDP19]*

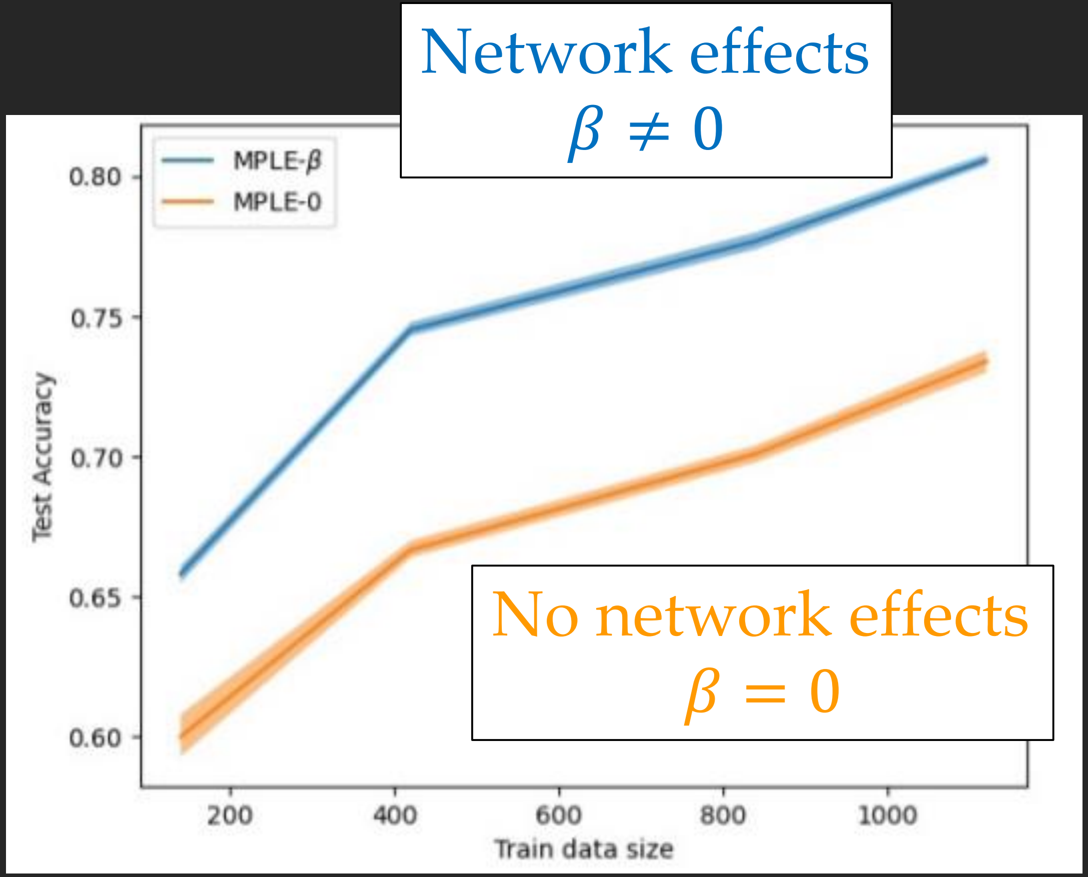
$$\Pr_{J}\left[y_1 \cdots y_n \mid x_1 \cdots x_n; \vec{h}, \beta \right] \propto e^{\sum_{i=1}^{n} h_i y_i + \beta \cdot \sum_{i,j=1}^{n} J_{ij} y_i y_j}$$

$\vec{h} \in \mathbb{R}^n$ external field ; J Interaction matrix ; β inverse temperature

- Proof
  - A new theorem on learning Ising models from one sample
  - Algorithm: maximum pseudo-likelihood estimator [Besag'75,…,Chatterjee'07]
  - We provide **a new analysis**, via the naïve mean-field equation
  - Significant challenges over prior work
    - [D, Daskalakis, Dikkala, Kandiros '20] No external field (no identification issues)
    - [Ghosal and Mukherjee '18] *[Daskalakis, Dikkala, Panageas '19] Restricted settings

Experiments:
It helps to utilize dependencies!

- Citation dataset: Cora
  - Nodes = publications
  - Edges = citation links
  - Labels = areas of publication
- Classification with
  Network and individual effects vs.
  Only individual effects (i.i.d.)
  - Plot: Test accuracy as a function of n
  - Implementation: using 2-layer Neural networks for individual effect



Summary

- Learning from dependent observations
- Efficient gradient based algorithm
- Proof: a general theorem on learning Ising models from one sample

- Furure work:
  - Even more complex dependency models (e.g. higher order)
  - Learning from partial information