Department of Statistics

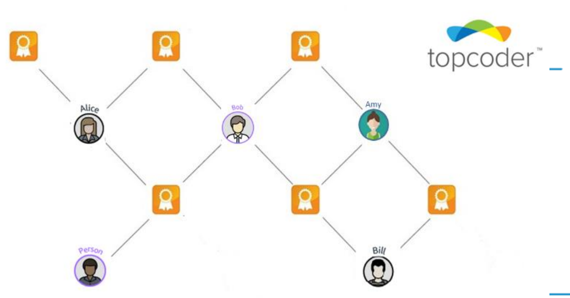
Representation Learning for relational data

Student: Yiliu Wang

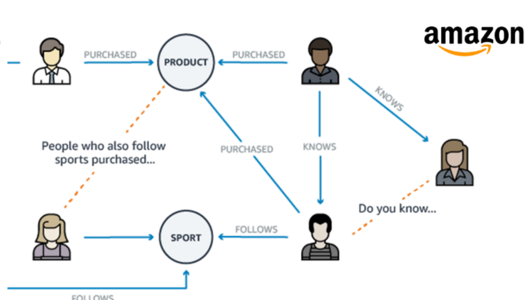
Supervisor: Milan Vojnovic

Motivations

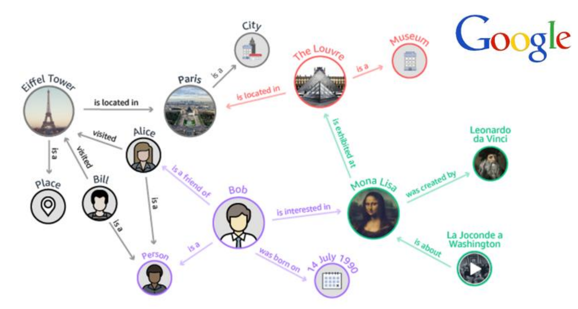
Relational data is widely used in our lives.



Online labor platform



Recommender system



Knowledge graph

Many relational data can be expressed using hypergraphs:

- **Nodes** represent items or individuals.
- **Hyperedges (subsets of nodes)** represent entities grouped by relationships

The challenge is to learn individual representations that explain observed data at a hyperedge level.

The Basic Model

Given ß = ($ß_1$, …, $ß_n$), the **generalized beta model** constructs a random hypergraph by putting a hyperedge independently for a group of nodes S with probability:

$$p(y_S = 1) = \frac{1}{1 + e^{-\sum_{i \in S} ß_i}}$$

It can be interpreted as modelling **group success** as a single dimensional **"OR"** of node parameter values.

The model has connections with:

- **Beta model for random graphs**: a special case of the generalized beta model when the cardinality for every group S is 2
- **Group testing problem:** a noisy version of group testing where each individual is according to a double-exponential distribution.

Open Research Questions and Initial findings

Key Questions

- **Existence and Uniqueness of MLE**: Under what specific conditions the maximum likelihood parameter estimate exists and is unique, or exists but may not be unique?
- **Accuracy and its relation to inputs:** In terms of the mean squared error of MLE, does this accuracy depends on key properties of the input data? If true, is it possible to accelerate learning by manipulating the input data?
- **Inference questions beyond MSE**: How can we test if some items are significantly better than others? If so, how can we identify the best or a set of best items with a minimum sampling complexity?

Main results

MLE **uniquely exists** if and only if the following conditions are satisfied:

- The design matrix formed by input data points is of **full rank**
- The input data points **overlap, or equivalently,** the scaled degree vector lies in the **polytope of degree sequences**

We also bridged the gap between uniqueness and existence conditions.

A key property for bounding the mean squared error of MLE is the **smallest eigenvalue** of the Gram matrix of the design matrix. This eigenvalue is strictly positive if and only if a certain graph-theoretic measure of **non-bipartiteness** is strictly positive.

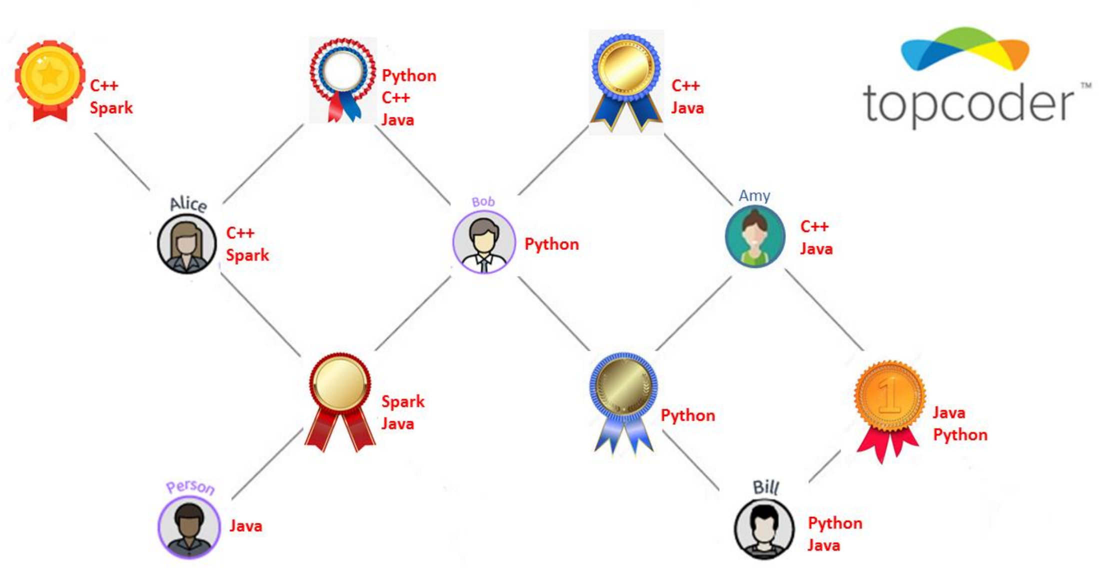
- Under the setting of **random design**, we can accelerate learning by adjusting group size
- The **optimal design** problem is equivalent to an ellipsoid inclusion problem solvable using semi-definite programing

The estimation problem has been studied extensively for related models.

- For **pairwise beta models**, asymptotically power hypothesis tests have been proposed.
- For **beta model with singletons**, a median elimination approach is optimal for finding the best item.
- For **group testing**, a near-optimal information theoretic approach is used to localize significant items.
- For **comparison models**, the scoring procedure is optimal to rank the item sets with respect to minimizing an expected risk.

Challenges

- Relationships might be explained by features in **multiple dimensions**
- Intricate **interactions** might exist between entities and features



An example of **online labor platform**

- each user has a particular skillset
- each project requires a particular combination of skills
- skills might be correlated

General models

- Use vector $ß_i$ to represent **latent features** of ith entity
- Use vector $γ_j$ to represent **task requirements** for jth relation

A natural extension:

$$p(y_S = 1) = \frac{1}{1 + \exp(-γ_j{}^T \sum_{i \in S} ß_i)}$$

It can be interpreted as modelling **group success** as a multi-dimensional **"OR"** of latent feature values.

- Feature correlations can be captured using a correlation matrix Θ
- Individual interactions can be characterized by a feature vector $ß_S$

A general model:

$$p(y_S = 1) = σ\left(γ_j{}^T Θ ß_S\right) \quad \text{where} \quad ß_S = f(ß_i, i \in S)$$

where σ denotes the logistic function and f is any vector-valued function acts component-wisely on individual features.