Network Learning

Steve Poulson

STG

28 November 2012

Outline

Motivation
- How to simulate the growth of power law graph
- New Link / Node prediction
- Implementation

Outline

Algorithm

Data: Directed Graph A
Result: Digraph B with n new vertices, m new edges
while *not n vertices* **do**
    pick random edge (v_1, v_2) from A where $v_1 \in$ B
    add v_2 add (v_1, v_2) to graph B
end
while *not m edges added* **do**
    pick random edge (v_1, v_2) from A where $v_1, v_2 \in$ B
    add (v_1, v_2) to graph B
end

Algorithm 1: Grow

Outline

Supervised learning problem

- $\mathbf{Y} \leftarrow \mathbf{A^{t+1}} - \mathbf{A^t}$
- $\min(L(f(\phi(\mathbf{A^t})), \mathbf{Y}))$

Supervised learning problem

- $\mathbf{Y} \leftarrow \mathbf{A^{t+1}} - \mathbf{A^t}$
- $\min(L(f(\phi(\mathbf{A^t})), \mathbf{Y}))$

Supervised learning problem

- $\mathbf{Y} \leftarrow \mathbf{A}^{\mathbf{t}+\mathbf{1}} - \mathbf{A}^{\mathbf{t}}$
- $\min(L(f(\phi(\mathbf{A}^{\mathbf{t}})), \mathbf{Y}))$

Feature Function ϕ

Maps $\mathbf{A}_{i,j} \rightarrow (x_1 \ldots x_n)$

- Random Walk between nodes
- Modularity
- Cosine Distance of neighbour list
- Betweeness Centrality

Feature Function ϕ

Maps $\mathbf{A}_{i,j} \rightarrow (x_1 \ldots x_n)$

- Random Walk between nodes
- Modularity
- Cosine Distance of neighbour list
- Betweeness Centrality

Feature Function ϕ

Maps $\mathbf{A}_{i,j} \rightarrow (x_1 \dots x_n)$

- Random Walk between nodes
- Modularity
- Cosine Distance of neighbour list
- Betweeness Centrality

Feature Function ϕ

Maps $\mathbf{A}_{i,j} \rightarrow (x_1 \ldots x_n)$

- Random Walk between nodes
- Modularity
- Cosine Distance of neighbour list
- Betweeness Centrality

Outline

Local vs Global

- Global features from a 1 million node graph unfeasible
- Power Law graphs scale invariant so local features should characterize graph
- Can use Hadoop Map calculates subset of local features, reduce assembles training set which classifier runs on

Local vs Global

- Global features from a 1 million node graph unfeasible
- Power Law graphs scale invariant so local features should characterize graph
- Can use Hadoop Map calculates subset of local features, reduce assembles training set which classifier runs on

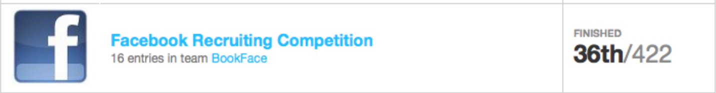
Local vs Global

- Global features from a 1 million node graph unfeasible
- Power Law graphs scale invariant so local features should characterize graph
- Can use Hadoop Map calculates subset of local features, reduce assembles training set which classifier runs on

Let's try it on a Kaggle competition

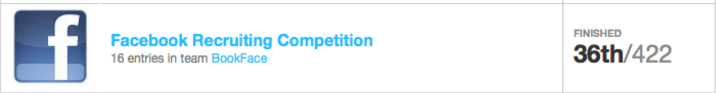
- Prototyped on sklearn / networkx
- Hadoop cluster on AWS
- Java: Weka / Hadoop / cern.colt.matrix / Jung
- Random Forrest classifer beat rest

Kaggle Evaluation



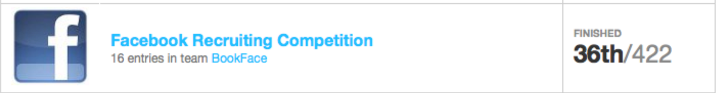
- A bit of fun :)
- Mean Average Precision @ 10 = 0.71371
- Within 2% of winner

Kaggle Evaluation



- A bit of fun :)
- Mean Average Precision @ 10 = 0.71371
- Within 2% of winner

Kaggle Evaluation



- A bit of fun :)
- Mean Average Precision @ 10 = 0.71371
- Within 2% of winner

Summary

- Power law Sampling gives a time varying dataset
- Random Forrest best
- Pretty good results