Using Local Spectral Methods to Robustify Graph-Based Learning (and why it matters to incomplete networks)

Code  www.cs.purdue.edu/homes/dgleich/codes/robust-diffusions

Joint work with Michael Mahoney @ Berkeley supported by NSF CAREER CCF-1149756

David F. Gleich
Purdue University

David Gleich · Purdue
The graph-based data analysis pipeline

Raw data
- Relationships
- Images
- Text records
- Etc.

Convert to a graph
- Nearest neighs
- Kernels
- 2-mode to 1-mode
- Etc.

Algorithm/Learning
- Important nodes
- Infer features
- Clustering
- Etc.
Semi-supervised graph-based learning

Given a graph, and a few labeled nodes, predict the labels on the rest of the graph.

Algorithm

1. Run a diffusion for each label (possibly with neg. info from other classes)
2. Assign new labels based on the value of each diffusion
Semi-supervised graph-based learning

Given a graph, and a few labeled nodes, predict the labels on the rest of the graph.

Algorithm

1. Run a diffusion for each label (possibly with neg. info from other classes)
2. Assign new labels based on the value of each diffusion
Semi-supervised graph-based learning

Given a graph, and a few labeled nodes, predict the labels on the rest of the graph.

Algorithm

1. Run a diffusion for each label (possibly with neg. info from other classes)
2. Assign new labels based on the value of each diffusion
The graph-based data analysis pipeline

1 0 0 0 1 0 0 1
0 1 0 1 0 0 1 1
0 1 0 1 0 0 0 1
1 0 0 0 0 0 1 1
1 1 0 1 1 1 0 1
1 0 1 1 0 0 1 1
1 0 1 1 1 0 1 0
1 1 1 1 0 1 0 0
1 1 1 0 0 1 1 1
1 1 0 1 1 1 1 1

Raw data
• Relationships
• Images
• Text records
• Etc.

Convert to a graph

Algorithm/Learning

The potential downstream signal is determined by this step

Most algorithmic and statistical research happens here

David Gleich · Purdue
**“Incompleteness” in the initial data modeling decisions**

<table>
<thead>
<tr>
<th>Explicit graphs</th>
<th>Constructed graphs</th>
<th>Labeled graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>are those that are given to a data analyst. (“n=all”)</td>
<td>are built based on some other primary data.</td>
<td>occur in information diffusion/propagation</td>
</tr>
<tr>
<td><strong>“A social network”</strong></td>
<td><strong>“nearest neighbor graphs”</strong></td>
<td><strong>“function prediction”</strong></td>
</tr>
<tr>
<td>- Friends not joined?</td>
<td>- K-NN or ε-NN</td>
<td>- Labeled nodes</td>
</tr>
<tr>
<td>- Users not logged in for a year?</td>
<td>- Thresholding correlations to zero</td>
<td>- Labeled edges</td>
</tr>
<tr>
<td>- Etc.</td>
<td></td>
<td>- Some are wrong</td>
</tr>
</tbody>
</table>

**Incomplete** due to the network sampling

**Incomplete** due to signal information / truncation

**Incomplete** due to unknown functions and roles
“Fast diffusion” arises “naturally” in algorithms on incomplete networks.

Given a graph $G$ that we expect to be missing some edges, along with some labels $s$ find an algorithm that

- produces smooth functions over the graph edges and the missing edges
- has the maximum amount of information

\[
\text{minimize } \text{wiggles}(\text{ALG},s) - \lambda \cdot \text{information}(\text{ALG})
\]

Optimization over the space of ALGORITHMS!

New stuff, possible small tech. issues

“Implicit” perspective in Orecchia and Mahoney, 2011
“Fast diffusion” arises “naturally” in algorithms on incomplete networks.

Given a graph G that we expect to be missing edges, along with some labels s find an algorithm that

- produces smooth functions over the graph edges and the missing edges
- has the maximum amount of information

\[
\text{Our model: } \text{prob. } \alpha \text{ s where } s_i \text{ iid}
\]

\[
\text{Our input: } f = Xs
\]

\[
\text{Our algorithm: } \sum (f_i - f_j)^2 \text{ log det } X^T X
\]

\[
\text{minimize } \mathbb{E} \left[ \sum (f_i - f_j)^2 \right] - \lambda \cdot \text{log det } X^T X
\]

Optimization over the space of LINEAR ALGORITHMS!
“Fast diffusion” arises “naturally” in algorithms on incomplete networks.

\[
\text{minimize } \quad E[\sum (f_i - f_j)^2] - \lambda \cdot \log \det X^T X
\]

\[
E[\sum (f_i - f_j)^2] = E[s^T X^T (L_G + \alpha L_{all}) X s] = \text{trace}(X^T (L_G + \alpha L_{all}) X)
\]

Super-simple, more insight here (and maybe a few more technical assumptions)!

\[
\text{minimize } \quad \text{trace}((L_G + \alpha L_{all}) Y) - \log \det Y
\]

subject to \( Y = Y^T, Y \succeq 0 \)

The Bottom Line

\[
Y = (L_G + \alpha L_{all})^{-1}
\]

\[
X = (L_G + \alpha L_{all})^{-1/2}
\]

An algorithm “closely” related to diffusion/PageRank-like results from a model on incomplete graphs
Semi-supervised graph-based learning

Given a graph, and a few labeled nodes, predict the labels on the rest of the graph.

Algorithm

1. Run a diffusion for each label (possibly with neg. info from other classes)
2. Assign new labels based on the value of each diffusion
Some important observations

1. We illustrate a common mincut framework for a variety of SSL diffusions
2. Provide a more robust SSL labeling method
3. Show how to “localize” one (and make it scalable / robust!)
The diffusions proposed for semi-supervised learning are \( s,t \)-cut minorants

In the unweighted case, solve via max-flow.

In the weighted case, solve via network simplex or industrial LP.

\[
\begin{align*}
\text{minimize} & \quad \sum_{ij \in E} C_{i,j} |x_i - x_j| \\
\text{subject to} & \quad x_s = 1, \quad x_t = 0.
\end{align*}
\]

MINCUT LP

\[
\begin{align*}
\text{minimize} & \quad \sqrt{\sum_{ij \in E} C_{i,j} |x_i - x_j|^2} \\
\text{subject to} & \quad x_s = 1, \quad x_t = 0.
\end{align*}
\]

Spectral minorant – lin. sys.
Representative cut problems

Zhou et al. NIPS 2003; Zhu et al., ICML 2003; Andersen Lang, SODA 2008; Joachims, ICML 2003
Value based rounding doesn’t work for all diffusions

(VALUE-BASED rounding fails for most of these diffusions)

BUT

There is still a signal there!

Adding more labels doesn’t help either, see the paper for those details
Rank-based rounding is far more robust.

NEW IDEA
Look at the RANK of the item in each diffusion instead of its VALUE.

JUSTIFICATION
Based on the idea of sweep-cut rounding in spectral methods (use the order induced by the eigenvector, not its values)

IMPACT
Much more robust rounding to labels
Implicit regularization views on the Zhou et al. diffusion

The Mahoney-Orecchia-Vishnoi (MOV) vector is a localized/regularized/robustified variation on the Fiedler vector to find a small conductance set nearby a seed.

minimize $\sqrt{\sum_{ij \in E} C_{i,j}|x_i - x_j|^2}$
subject to $x_S = 1, x_t = 0$.

RESULT
The spectral minorant of Zhou is equivalent to the weakly-local MOV solution.

PROOF
The two linear systems are the same (after working out a few equivalences).

IMPORTANCE
We’d expect Zhou to be “more robust”
We find this in synthetic and real-world studies.

Real-world scenario: sparse graph, high error

Sparse graph, low error
A scalable, localized algorithm for Zhou et al’s diffusion.

\[
\text{minimize } \sqrt{\sum_{ij \in E} C_{i,j}|x_i - x_j|^2} \\
\text{subject to } x_s = 1, x_t = 0.
\]

RESULT
We can use a variation on coordinate descent methods related to the Andersen-Chung-Lang PUSH procedure to solve Zhou’s diffusion in a scalable manner.

PROOF. See Gleich-Mahoney ICML ‘14
IMPORANCE (1)
We should be able to make Zhou et al. scale.

IMPORANCE (2)
Using this algorithm adds another implicit regularization term that should further improve robustness!

\[
\begin{align*}
\text{minimize } & \sum_{ij \in E} C_{i,j}|x_i - x_j|^2 + \tau \sum_{i \in V} d_i x_i \\
\text{subject to } & x_s = 1, x_t = 0, x_i \geq 0.
\end{align*}
\]
Varying density in an SSL construction.

We use the digits experiment from Zhou et al. 2003. 10 digits and a few label errors.

We vary density either by the num. of nearest neighbors or by the kernel width.

\[ A_{i,j} = \exp \left( -\frac{\|d_i - d_j\|^2_2}{2\sigma^2} \right) \]
Then we find that using 1-norm regularization improves accuracy.
Recap, discussion, future work

- Faster algorithms are **better** for incomplete data!
- “Fast diffusion” arises “naturally” in algorithms on incomplete networks.
- TODO Find algorithms that are relevant for real-world “missing” models.

Given a graph $G$ that we expect to be missing edges, along with some labels $s$ find an algorithm that

- produces smooth functions over the graph edges and the missing edges
- has the maximum amount of information

Our algorithm

\[
\text{Our input} \quad f = Xs
\]

\[
\text{Our model} \quad \alpha \text{ prob. where } s_i \text{ iid}
\]

\[
\text{Our algorithm} \quad \sum (f_i - f_j)^2
\]

\[
\log \det X^T X
\]