Scalable Learning & Inference
Over Graphs

Partha Pratim Talukdar
Indian Institute of Science
ppt@cds.iisc.ac.in

February 17, 2015
Supervised Learning

Labeled Data → Learning Algorithm → Model
Supervised Learning

Examples:
- Decision Trees
- Support Vector Machine (SVM)
- Maximum Entropy (MaxEnt)
Semi-Supervised Learning (SSL)

Labeled Data → Learning Algorithm → Model

A Lot of Unlabeled Data
Semi-Supervised Learning (SSL)

Labeled Data

Learning Algorithm

Model

A Lot of Unlabeled Data

Examples: Self-Training Co-Training
Why SSL?

How can unlabeled data be helpful?
Why SSL?

How can unlabeled data be helpful?

Without Unlabeled Data
Why SSL?

How can unlabeled data be helpful?

Labeled Instances

Without Unlabeled Data
Why SSL?

How can unlabeled data be helpful?
Why SSL?

How can unlabeled data be helpful?

Labeled Instances

Decision Boundary

Without Unlabeled Data

With Unlabeled Data
Why SSL?

How can unlabeled data be helpful?

With Unlabeled Data

Without Unlabeled Data

Labeled Instances

Decision Boundary

Unlabeled Instances
Why SSL?

How can unlabeled data be helpful?

Example from [Belkin et al., JMLR 2006]
Inductive vs Transductive
Inductive vs Transductive

Supervised
(Labeled)

Semi-supervised
(Labeled + Unlabeled)
Inductive vs Transductive

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Inductive</th>
<th>Transductive</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Labeled)</td>
<td>(Generalize to Unseen Data)</td>
<td>(Doesn’t Generalize to Unseen Data)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semi-supervised</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(Labeled + Unlabeled)</td>
<td></td>
</tr>
</tbody>
</table>
Inductive vs Transductive

Inductive (Generalize to Unseen Data)

Transductive (Doesn’t Generalize to Unseen Data)

Supervised (Labeled)

SVM, Maximum Entropy

Semi-supervised (Labeled + Unlabeled)
<table>
<thead>
<tr>
<th></th>
<th>Inductive (Generalize to Unseen Data)</th>
<th>Transductive (Doesn’t Generalize to Unseen Data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>SVM, Maximum Entropy</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Semi-supervised (Labeled + Unlabeled)</td>
<td></td>
</tr>
</tbody>
</table>
Inductive vs Transductive

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Inductive</th>
<th>Transductive</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Labeled)</td>
<td>SVM, Maximum Entropy</td>
<td>(Doesn’t Generalize to Unseen Data)</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>Manifold Regularization</td>
<td>X</td>
</tr>
<tr>
<td>(Labeled + Unlabeled)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised (Labeled)</td>
<td>Inductive (Generalize to Unseen Data)</td>
<td>Transductive (Doesn’t Generalize to Unseen Data)</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>SVM, Maximum Entropy</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Semi-supervised (Labeled + Unlabeled)</td>
<td>Manifold Regularization</td>
<td>Label Propagation (LP), MAD, MP, ...</td>
</tr>
</tbody>
</table>
## Inductive vs Transductive

<table>
<thead>
<tr>
<th>Supervised (Labeled)</th>
<th>Inductive (Generalize to Unseen Data)</th>
<th>Transductive (Doesn’t Generalize to Unseen Data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM, Maximum Entropy</td>
<td>X</td>
<td>Label Propagation (LP), MAD, MP, ...</td>
</tr>
<tr>
<td>Manifold Regularization</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Most Graph SSL algorithms are non-parametric** (i.e., # parameters grows with data size)
Inductive vs Transductive

<table>
<thead>
<tr>
<th>Supervised (Labeled)</th>
<th>Inductive (Generalize to Unseen Data)</th>
<th>Transductive (Doesn’t Generalize to Unseen Data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM, Maximum Entropy</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Semi-supervised (Labeled + Unlabeled)</td>
<td>Manifold Regularization</td>
<td>Label Propagation (LP), MAD, MP, ...</td>
</tr>
</tbody>
</table>

Most Graph SSL algorithms are non-parametric (i.e., # parameters grows with data size)
Inductive vs Transductive

Inductive (Generalize to Unseen Data)

SVM, Maximum Entropy

Transductive (Doesn’t Generalize to Unseen Data)

X

Supervised (Labeled)

Semi-supervised (Labeled + Unlabeled)

Manifold Regularization

Label Propagation (LP), MAD, MP, ...

Most Graph SSL algorithms are non-parametric (i.e., # parameters grows with data size)

Two Popular SSL Algorithms

- Self Training
Two Popular SSL Algorithms

• Self Training

• Co-Training

Given:

• a set $L$ of labeled training examples
• a set $U$ of unlabeled examples

Create a pool $U'$ of examples by choosing $u$ examples at random from $U$

Loop for $k$ iterations:

Use $L$ to train a classifier $h_1$ that considers only the $x_1$ portion of $x$
Use $L$ to train a classifier $h_2$ that considers only the $x_2$ portion of $x$

Allow $h_1$ to label $p$ positive and $n$ negative examples from $U'$
Allow $h_2$ to label $p$ positive and $n$ negative examples from $U'$
Add these self-labeled examples to $L$
Randomly choose $2p + 2n$ examples from $U$ to replenish $U'$
Why Graph-based SSL?
Why Graph-based SSL?

• Some datasets are naturally represented by a graph
  • web, citation network, social network, ...
Why Graph-based SSL?

• Some datasets are naturally represented by a graph
  • web, citation network, social network, ...

• Uniform representation for heterogeneous data
Why Graph-based SSL?

- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...
- Uniform representation for heterogeneous data
- Easily parallelizable, scalable to large data
Why Graph-based SSL?

- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...
- Uniform representation for heterogeneous data
- Easily parallelizable, scalable to large data
- Effective in practice
Why Graph-based SSL?

- Some datasets are naturally represented by a graph
  - web, citation network, social network, ...
- Uniform representation for heterogeneous data
- Easily parallelizable, scalable to large data
- Effective in practice

![Text Classification Graph SSL Non-Graph SSL Supervised](image)
Graph-based SSL
Graph-based SSL
Graph-based SSL

![Diagram representing graph-based SSL with similarity scores of 0.8, 0.2, and 0.6 between documents.](image)
Graph-based SSL

"business" 0.8 0.2 0.6 "politics"

Similarity
Graph-based SSL

"business"  Similarity  "politics"

0.8  0.2  0.6

"business"  "politics"
Graph-based SSL
Smoothness Assumption
If two instances are similar according to the graph, then output labels should be similar.
Graph-based SSL

Smoothness Assumption
If two instances are similar according to the graph, then output labels should be similar.
Graph-based SSL

Smoothness Assumption
If two instances are similar according to the graph, then output labels should be similar.

- Two stages
  - Graph construction (if not already present)
  - Label Inference
Outline

• Motivation
• Graph Construction
• Inference Methods
• Scalability
• Applications
• Conclusion & Future Work
Outline

• Motivation

• Graph Construction

• Inference Methods

• Scalability

• Applications

• Conclusion & Future Work
Graph Construction

• Neighborhood Methods
  • k-NN Graph Construction (k-NNG)
  • e-Neighborhood Method

• Metric Learning

• Other approaches
Neighborhood Methods
Neighborhood Methods

• k-Nearest Neighbor Graph (k-NNG)
  • add edges between an instance and its k-nearest neighbors
Neighborhood Methods

- k-Nearest Neighbor Graph (k-NNG)
  - add edges between an instance and its k-nearest neighbors

![Diagram of k-NNG graph with k = 3]
Neighborhood Methods

• **k-Nearest Neighbor Graph (k-NNG)**
  • add edges between an instance and its $k$-nearest neighbors

• **e-Neighborhood**
  • add edges to all instances inside a ball of radius $e$
Neighborhood Methods

• **k-Nearest Neighbor Graph (k-NNG)**
  • add edges between an instance and its k-nearest neighbors

• **e-Neighborhood**
  • add edges to all instances inside a ball of radius $e$
Issues with k-NNG
Issues with k-NNG

• Not scalable (quadratic)
Issues with k-NNG

• Not scalable (quadratic)
• Results in an asymmetric graph
Issues with k-NNG

- Not scalable (quadratic)
- Results in an asymmetric graph
  - b is the closest neighbor of a, but not the other way
Issues with k-NNG

• Not scalable (quadratic)
• Results in an asymmetric graph
  • b is the closest neighbor of a, but not the other way

• Results in irregular graphs
  • some nodes may end up with higher degree than other nodes
Issues with k-NNG

• Not scalable (quadratic)
• Results in an asymmetric graph
  • \( b \) is the closest neighbor of \( a \), but not the other way

• Results in **irregular graphs**
  • some nodes may end up with higher degree than other nodes

Node of degree 4 in the k-NNG (\( k = 1 \))
Issues with $e$-Neighborhood
Issues with $\epsilon$-Neighborhood

• Not scalable
Issues with $\epsilon$-Neighborhood

- Not scalable
- **Sensitive to value of $\epsilon$**: not invariant to scaling
Issues with $\epsilon$-Neighborhood

• Not scalable

• **Sensitive to value of $\epsilon$**: not invariant to scaling

• **Fragmented Graph**: disconnected components
Issues with $\epsilon$-Neighborhood

• Not scalable

• **Sensitive to value of $\epsilon$**: not invariant to scaling

• **Fragmented Graph**: disconnected components

Figure from [Jebara et al., ICML 2009]
Graph Construction using Metric Learning
Graph Construction using Metric Learning

\[ w_{ij} \propto \exp(-D_A(x_i, x_j)) \]
Graph Construction using Metric Learning

\[ w_{ij} \propto \exp(-D_A(x_i, x_j)) \]

\[ D_A(x_i, x_j) = (x_i - x_j)^T A(x_i - x_j) \]

Estimated using Mahalanobis metric learning algorithms
Graph Construction using Metric Learning

- **Supervised Metric Learning**
  - ITML [Kulis et al., ICML 2007]
  - LMNN [Weinberger and Saul, JMLR 2009]

- **Semi-supervised Metric Learning**
  - IDML [Dhillon et al., UPenn TR 2010]

\[
D_A(x_i, x_j) = (x_i - x_j)^T A(x_i - x_j)
\]

\[w_{ij} \propto \exp(-D_A(x_i, x_j))\]
Benefits of Metric Learning for Graph Construction
Benefits of Metric Learning for Graph Construction

100 seed and 1400 test instances, all inferences using LP
Benefits of Metric Learning for Graph Construction

Graph constructed using supervised metric learning

Error

Amazon  | Newsgroups  | Reuters  | Enron A  | Text

100 seed and 1400 test instances, all inferences using LP
Benefits of Metric Learning for Graph Construction

Graph constructed using supervised metric learning

Graph constructed using semi-supervised metric learning

[Dhillon et al., UPenn TR 2010]

Error

Amazon  Newsgroups  Reuters  Enron A  Text

100 seed and 1400 test instances, all inferences using LP

[Dhillon et al., UPenn TR 2010]
Benefits of Metric Learning for Graph Construction

Careful graph construction is critical!

Graph constructed using supervised metric learning [Dhillon et al., UPenn TR 2010]

Graph constructed using semi-supervised metric learning [Dhillon et al., 2010]

100 seed and 1400 test instances, all inferences using LP
Other Graph Construction Approaches

• Local Reconstruction
  • Linear Neighborhood [Wang and Zhang, ICML 2005]
  • Regular Graph: b-matching [Jebara et al., ICML 2008]
  • Fitting Graph to Vector Data [Daitch et al., ICML 2009]

• Graph Kernels
  • [Zhu et al., NIPS 2005]
Outline

• Motivation
• Graph Construction
• Inference Methods
• Scalability
• Applications
• Conclusion & Future Work
Graph Laplacian
Graph Laplacian

• Laplacian (un-normalized) of a graph:

\[ L = D - W, \text{ where } D_{ii} = \sum_j W_{ij}, \quad D_{ij}(\neq i) = 0 \]
Graph Laplacian

• Laplacian (un-normalized) of a graph:

\[ L = D - W, \text{ where } D_{ii} = \sum_j W_{ij}, \quad D_{ij}(\neq i) = 0 \]
Graph Laplacian

- Laplacian (un-normalized) of a graph:

\[ L = D - W, \text{ where } D_{ii} = \sum_j W_{ij}, \ D_{ij}(\neq i) = 0 \]

\[
\begin{pmatrix}
  a & b & c & d \\
  a & 3 & -1 & -2 & 0 \\
b & -1 & 4 & -3 & 0 \\
c & -2 & -3 & 6 & -1 \\
d & 0 & 0 & -1 & 1 \\
\end{pmatrix}
\]
Graph Laplacian (contd.)

• \( L \) is positive semi-definite (assuming non-negative weights)
• Smoothness of prediction \( f \) over the graph in terms of the Laplacian:
Graph Laplacian (contd.)

- $L$ is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction $f$ over the graph in terms of the Laplacian:

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$
Graph Laplacian (contd.)

- $L$ is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction $f$ over the graph in terms of the Laplacian:

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$
Graph Laplacian (contd.)

• $L$ is positive semi-definite (assuming non-negative weights)
• Smoothness of prediction $f$ over the graph in terms of the Laplacian:

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$

Vector of scores for single label on nodes

Measure of Non-Smoothness
Graph Laplacian (contd.)

- $L$ is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction $f$ over the graph in terms of the Laplacian:

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$

Vector of scores for single label on nodes

Measure of Non-Smoothness

$$f^T = [1 \ 10 \ 5 \ 25]$$
Graph Laplacian (contd.)

- $L$ is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction $f$ over the graph in terms of the Laplacian:

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$

Vector of scores for single label on nodes

Measure of Non-Smoothness

$$f^T = [1 \ 10 \ 5 \ 25]$$

$$f^T L f = 588$$

Not Smooth
Graph Laplacian (contd.)

- \( L \) is positive semi-definite (assuming non-negative weights)
- Smoothness of prediction \( f \) over the graph in terms of the Laplacian:

\[
f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2
\]

Vector of scores for single label on nodes

- Measure of Non-Smoothness

```
f^T = [1 1 3]
f^T L f = 4
```

```
f^T = [1 10 5 25]
f^T L f = 588
```

Not Smooth
Graph Laplacian (contd.)

• $L$ is positive semi-definite (assuming non-negative weights)
• Smoothness of prediction $f$ over the graph in terms of the Laplacian:

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$

Vector of scores for single label on nodes

Measure of Non-Smoothness

**Example**: Let $f^T = [1 1 3 25]$. Then

$$f^T L f = \sum_{i,j} W_{ij} (f_i - f_j)^2$$

For the graph on the left:

- $f^T L f = 4$
- Smooth

For the graph on the right:

- $f^T = [1 10 5 25]$
- $f^T L f = 588$
- Not Smooth
Outline

• Motivation
• Graph Construction
• Inference Methods
• Scalability
• Applications
• Conclusion & Future Work

Label Propagation
- Modified Adsorption
- Measure Propagation
- Sparse Label Propagation
- Manifold Regularization
Notations

\( Y_{v,l} \) : score of seed label \( l \) on node \( v \)

\( \hat{Y}_{v,l} \) : score of estimated label \( l \) on node \( v \)

\( R_{v,l} \) : regularization target for label \( l \) on node \( v \)

\( S \) : seed node indicator (diagonal matrix)

\( W_{uv} \) : weight of edge \((u, v)\) in the graph
LP-ZGL [Zhu et al., ICML 2003]

$$\arg \min_{\hat{Y}} \sum_{l=1}^{m} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l$$

such that $Y_{ul} = \hat{Y}_{ul}$, $\forall S_{uu} = 1$
LP-ZGL [Zhu et al., ICML 2003]

\[
\operatorname{arg\ min}_{\hat{Y}} \sum_{l=1}^{m} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l
\]

such that \( Y_{ul} = \hat{Y}_{ul}, \forall S_{uu} = 1 \)
LP-ZGL [Zhu et al., ICML 2003]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l
\]

such that \( Y_{ul} = \hat{Y}_{ul}, \forall S_{uu} = 1 \)

Smooth Match Seeds (hard)

Graph Laplacian
LP-ZGL [Zhu et al., ICML 2003]

$$\arg\min_{\hat{Y}} \sum_{l=1}^{m} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 = \sum_{l=1}^{m} \hat{Y}_l^T L \hat{Y}_l$$

such that

$$Y_{ul} = \hat{Y}_{ul}, \; \forall S_{uu} = 1$$

Smooth

Match Seeds (hard)

- Smoothness
  - two nodes connected by an edge with high weight should be assigned similar labels

Graph Laplacian
LP-ZGL [Zhu et al., ICML 2003]

arg min \( \hat{Y} \) \[
\sum_{l=1}^{m} W_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2
\]

such that \( Y_{ul} = \hat{Y}_{ul}, \forall S_{uu} = 1 \)

Smooth

Match Seeds (hard)

Smoothness
- two nodes connected by an edge with high weight should be assigned similar labels

Solution satisfies harmonic property

Smoothness
Outline

- Motivation
- Graph Construction
- Inference Methods
- Scalability
- Applications
- Conclusion & Future Work

Inference Methods:
- Label Propagation
- Modified Adsorption
- Manifold Regularization
- Spectral Graph Transduction
- Measure Propagation
Modified Adsorption (MAD)
[Talukdar and Crammer, ECML 2009]
Modified Adsorption (MAD)

[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left[ \| S \hat{Y}_l - SY_l \|^2 + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \| \hat{Y}_l - R_l \|^2 \right]
\]

- \( m \) labels, +1 dummy label
- \( M = W^\top + W' \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)

Seed Scores
Estimated Scores
Label Priors

25
Modified Adsorption (MAD)
[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left( \left\| S \hat{Y}_l - SY_l \right\|^2 + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \left\| \hat{Y}_l - R_l \right\|^2 \right)
\]

- \( m \) labels, +1 dummy label
- \( M = W^\top + W' \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)
Modified Adsorption (MAD)  
[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left( \| S \hat{Y}_l - SY_l \|^2 \right) + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \| \hat{Y}_l - R_l \|^2
\]

- \( m \) labels, +1 dummy label
- \( M = W' + W' \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)
Modified Adsorption (MAD)
[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left\| S \hat{Y}_l - SY_l \right\|^2 + \mu_1 \sum_{u,v} M_{uv} (\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \left\| \hat{Y}_l - R_l \right\|^2
\]

- \( m \) labels, +1 dummy label
- \( M = W^T + W' \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)
Modified Adsorption (MAD)  
[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left[ \| S\hat{Y}_l - S\hat{Y} \|_2^2 \right] + \mu_1 \sum_{u,v} M_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \| \hat{Y}_l - R_l \|_2^2
\]

- \( m \) labels, +1 dummy label
- \( M = \) for none-of-the-above label
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)
Modified Adsorption (MAD)
[Talukdar and Crammer, ECML 2009]

$$\arg \min_{\mathbf{Y}} \sum_{l=1}^{m+1} \left( \| \mathbf{S} \mathbf{\hat{Y}}_l - \mathbf{S} \mathbf{Y}_l \|^2 \right) + \mu_1 \sum_{u,v} \mathbf{M}_{uv} (\mathbf{\hat{Y}}_{ul} - \mathbf{\hat{Y}}_{vl})^2 + \mu_2 \| \mathbf{\hat{Y}}_l - \mathbf{R}_l \|^2$$

- \(m\) labels, +1 dummy label
- \(\mathbf{M} = \mathbf{W} + \mathbf{W}\) is the symmetrized weight matrix
- \(\mathbf{\hat{Y}}_{vl}\): weight of label \(l\) on node \(v\)
- \(\mathbf{Y}_{vl}\): seed weight for label \(l\) on node \(v\)
- \(\mathbf{S}\): diagonal matrix, nonzero for seed nodes
- \(\mathbf{R}_{vl}\): regularization target for label \(l\) on node \(v\)

**Match Seeds (soft)**

**Smooth**

**Match Priors (Regularizer)**

MAD has extra regularization compared to LP-ZGL
[Zhu et al, ICML 03]; similar to QC [Bengio et al, 2006]
Modified Adsorption (MAD)
[Talukdar and Crammer, ECML 2009]

\[
\arg\min_{\hat{Y}} \sum_{l=1}^{m+1} \left( \| S\hat{Y}_l - SY_l \|^2 \right) + \mu_1 \sum_{u,v} M_{uv}(\hat{Y}_{ul} - \hat{Y}_{vl})^2 + \mu_2 \| \hat{Y}_l - R_l \|^2
\]

- \( m \) labels, +1 dummy label
- \( M = W + W^T \) is the symmetrized weight matrix
- \( \hat{Y}_{vl} \): weight of label \( l \) on node \( v \)
- \( Y_{vl} \): seed weight for label \( l \) on node \( v \)
- \( S \): diagonal matrix, nonzero for seed nodes
- \( R_{vl} \): regularization target for label \( l \) on node \( v \)

MAD has extra regularization compared to LP-ZGL
[Zhu et al, ICML 03]; similar to QC [Bengio et al, 2006]
Solving MAD Objective
Solving MAD Objective

• Can be solved using matrix inversion (like in LP)
  • but matrix inversion is expensive
Solving MAD Objective

• Can be solved using matrix inversion (like in LP)
  • but matrix inversion is expensive

• Instead solved exactly using a system of linear equations ($Ax = b$)
  • solved using Jacobi iterations
  • results in iterative updates
  • guaranteed convergence

• see [Bengio et al., 2006] and [Talukdar and Crammer, ECML 2009] for details
Solving MAD using Iterative Updates

Inputs $\mathbf{Y}, \mathbf{R} : |V| \times (|L| + 1)$, $\mathbf{W} : |V| \times |V|$, $\mathbf{S} : |V| \times |V|$ diagonal

$\hat{\mathbf{Y}} \leftarrow \mathbf{Y}$

$\mathbf{M} = \mathbf{W}' + \mathbf{W}^\dagger$

$Z_v \leftarrow S_{vv} + \mu_1 \sum_{u \neq v} M_{vu} + \mu_2 \quad \forall v \in V$

repeat

for all $v \in V$ do

$\hat{Y}_v \leftarrow \frac{1}{Z_v} \left( (SY)_v + \mu_1 M_v \hat{Y} + \mu_2 R_v \right)$

end for

until convergence
Solving MAD using Iterative Updates

Inputs $\mathbf{Y}, \mathbf{R} : |V| \times (|L| + 1)$, $\mathbf{W} : |V| \times |V|$, $\mathbf{S} : |V| \times |V|$ diagonal

$\hat{\mathbf{Y}} \leftarrow \mathbf{Y}$

$\mathbf{M} = \mathbf{W}' + \mathbf{W}^\dagger$

$Z_v \leftarrow S_{vv} + \mu_1 \sum_{u \neq v} M_{vu} + \mu_2 \quad \forall v \in V$

repeat

for all $v \in V$ do

$\hat{\mathbf{Y}}_v \leftarrow \frac{1}{Z_v} \left( (S \mathbf{Y})_v + \mu_1 M_v \hat{\mathbf{Y}} + \mu_2 R_v \right)$

end for

until convergence
Solving MAD using Iterative Updates

Inputs $Y, R: |V| \times (|L| + 1), W: |V| \times |V|, S: |V| \times |V|$ diagonal

$\hat{Y} \leftarrow Y$

$M = W' + W^\dagger$

$Z_v \leftarrow S_{vv} + \mu_1 \sum_{u \neq v} M_{vu} + \mu_2 \quad \forall v \in V$

repeat

for all $v \in V$ do

$\hat{Y}_v \leftarrow \frac{1}{Z_v} \left( (SY)_v + \mu_1 M_v \cdot \hat{Y} + \mu_2 R_v \right)$

end for

until convergence

• Importance of a node can be discounted
• Easily Parallelizable: Scalable (more later)
Other Graph-based SSL Methods

- **TACO** [Orbach and Crammer, ECML 2012]
- **SSL on Directed Graphs**
  - [Zhou et al, NIPS 2005], [Zhou et al., ICML 2005]
- **Spectral Graph Transduction** [Joachims, ICML 2003]
- **Graph-SSL for Ordering**
  - [Talukdar et al., CIKM 2012]
- **Learning with dissimilarity edges**
  - [Goldberg et al., AISTATS 2007]
Outline

• Motivation
• Graph Construction
• Inference Methods
• **Scalability**
  - Scalability Issues
  - Node reordering
  - MapReduce Parallelization
• Applications
• Conclusion & Future Work
More (Unlabeled) Data is Better Data
More (Unlabeled) Data is Better Data

[Graph with 120m vertices]

[Subramanya & Bilmes, JMLR 2011]
More (Unlabeled) Data is Better Data

Challenges with large unlabeled data:
• Constructing graph from large data
• Scalable inference over large graphs

Graph with 120m vertices

Subramanya & Bilmes, JMLR 2011
Outline

- Motivation
- Graph Construction
- Inference Methods
- Scalability
- Applications
- Conclusion & Future Work

Scalability Issues
- Node reordering
  [Subramanya & Bilmes, JMLR 2011; Bilmes & Subramanya, 2011]
- MapReduce Parallelization
Label Update using Message Passing

Processor 1
Processor 2
Processor k
Processor 1

SMP with k Processors

Graph nodes (neighbors not shown)

Current label estimate on a

Processor 1
Processor 2
Processor k
Processor 1

SMP with k Processors

Graph nodes (neighbors not shown)
Label Update using Message Passing

Processor 1
Processor 2
Processor k
Processor 1

SMP with k Processors

Graph nodes (neighbors not shown)

New label estimate on v

Seed
Prior

Processor 1
Processor 2
Processor k
Processor 1

0.60
0.75
0.05
0.05

v

a

b

c
Node Reordering Algorithm: Intuition

![Diagram showing nodes k, a, b, and c with connections]

k → a → b → c
Node Reordering Algorithm: Intuition

Which node should be processed along with k: the one with highest intersection of neighborhood with k.
Which node should be processed along with k: the one with highest intersection of neighborhood with k
Node Reordering Algorithm: Intuition

Which node should be processed along with $k$: the one with highest intersection of neighborhood with $k$
Node Reordering Algorithm: Intuition

Which node should be processed along with k: the one with highest intersection of neighborhood with k
Node Reordering Algorithm: Intuition

Cardinality of Intersection

| $N(k) \cap N(a)$ | = 1

Which node should be processed along with k: the one with highest intersection of neighborhood with k
Node Reordering Algorithm: Intuition

Which node should be processed along with k: the one with highest intersection of neighborhood with k

Cardinality of Intersection

| $|N(k) \cap N(a)| = 1$ |
|-------------------------|
| $|N(k) \cap N(b)| = 2$ |
| $|N(k) \cap N(c)| = 0$ |
Node Reordering Algorithm: Intuition

Cardinality of Intersection

\[ |N(k) \cap N(a)| = 1 \]

\[ |N(k) \cap N(b)| = 2 \]

\[ |N(k) \cap N(c)| = 0 \]

Which node should be processed along with k: the one with highest intersection of neighborhood with k
Speed-up on SMP after Node Ordering

[Subramanya & Bilmes, JMLR, 2011]
Outline

- Motivation
- Graph Construction
- Inference Methods
- Scalability
- Applications
- Conclusion & Future Work

Scalability Issues
- Node reordering
- MapReduce Parallelization
MapReduce Implementation of MAD

Current label estimate on b

Seed 0.60
Prior 0.75

v

0.05

36
MapReduce Implementation of MAD

- **Map**
  - Each node sends its current label assignments to its neighbors
MapReduce Implementation of MAD

• Map
  • Each node send its current label assignments to its neighbors
MapReduce Implementation of MAD

• **Map**
  - Each node sends its current label assignments to its neighbors

• **Reduce**
  - Each node updates its own label assignment using messages received from neighbors, and its own information (e.g., seed labels, reg. penalties etc.)

• Repeat until convergence
MapReduce Implementation of MAD

- **Map**
  - Each node sends its current label assignments to its neighbors

- **Reduce**
  - Each node updates its own label assignment using messages received from neighbors, and its own information (e.g., seed labels, reg. penalties etc.)

- Repeat until convergence
MapReduce Implementation of MAD

- **Map**
  - Each node sends its current label assignments to its neighbors

- **Reduce**
  - Each node updates its own label assignment using messages received from neighbors, and its own information (e.g., seed labels, regular penalties, etc.)

- **Repeat until convergence**

**Code in Junto Label Propagation Toolkit**
(includes Hadoop-based implementation)

[https://github.com/parthatalukdar/junto](https://github.com/parthatalukdar/junto)
MapReduce Implementation of MAD

• Map
  - Each node sends its current label assignments to its neighbors

• Reduce
  - Each node updates its own label assignment using messages received from neighbors, and its own information (e.g., seed labels, reg. penalties etc.)

• Repeat until convergence

Graph-based algorithms are amenable to distributed processing

Code in Junto Label Propagation Toolkit (includes Hadoop-based implementation)
https://github.com/parthatalukdar/junto
When to use Graph-based SSL and which method?
When to use Graph-based SSL and which method?

- When input data itself is a graph (relational data)
- or, when the data is expected to lie on a manifold
When to use Graph-based SSL and which method?

• When input data itself is a graph (relational data)
  • or, when the data is expected to lie on a manifold

• MAD, Quadratic Criteria (QC)
  • when labels are not mutually exclusive
  • MADDL: when label similarities are known
When to use Graph-based SSL and which method?

- When input data itself is a graph (relational data)
  - or, when the data is expected to lie on a manifold

- MAD, Quadratic Criteria (QC)
  - when labels are not mutually exclusive
  - MADDL: when label similarities are known

- Measure Propagation (MP)
  - for probabilistic interpretation
When to use Graph-based SSL and which method?

- When input data itself is a graph (relational data)
  - or, when the data is expected to lie on a manifold
- MAD, Quadratic Criteria (QC)
  - when labels are not mutually exclusive
  - MADDL: when label similarities are known
- Measure Propagation (MP)
  - for probabilistic interpretation
- Manifold Regularization
  - for generalization to unseen data (induction)
Graph-based SSL: Summary
Graph-based SSL: Summary

• Provide flexible representation
  • for both IID and relational data
Graph-based SSL: Summary

• Provide flexible representation
  • for both IID and relational data
• Graph construction can be key
Graph-based SSL: Summary

- Provide flexible representation
  - for both IID and relational data
- Graph construction can be key
- Scalable: Node Reordering and MapReduce
Graph-based SSL: Summary

• Provide flexible representation
  • for both IID and relational data
• Graph construction can be key
• Scalable: Node Reordering and MapReduce
• Can handle labeled as well as unlabeled data
Graph-based SSL: Summary

- Provide flexible representation
  - for both IID and relational data
- Graph construction can be key
- Scalable: Node Reordering and MapReduce
- Can handle labeled as well as unlabeled data
- Can handle multi class, multi label settings
Graph-based SSL: Summary

• Provide flexible representation
  • for both IID and relational data
• Graph construction can be key
• Scalable: Node Reordering and MapReduce
• Can handle labeled as well as unlabeled data
• Can handle multi class, multi label settings
• Effective in practice
Open Challenges
Open Challenges

• Graph-based SSL for Structured Prediction
  • Algorithms: Combining Inductive and graph-based methods
  • Applications: Constituency and dependency parsing, Coreference
Open Challenges

• Graph-based SSL for Structured Prediction
  • Algorithms: Combining Inductive and graph-based methods
  • Applications: Constituency and dependency parsing, Coreference

• Scalable graph construction, especially with multi-modal data
Open Challenges

• Graph-based SSL for Structured Prediction
  • Algorithms: Combining Inductive and graph-based methods
  • Applications: Constituency and dependency parsing, Coreference

• Scalable graph construction, especially with multi-modal data

• Extensions with other loss functions, sparsity, etc.
References (I)


References (II)


Thanks!

Web: http://graph-ssl.wikidot.com/