Collective classification in network data

Seminar on graphs, UCSB 2009
1 Problem

2 Methods
   - Local methods
   - Global methods

3 Experiments
1 Problem

2 Methods
   - Local methods
   - Global methods

3 Experiments
Example
Correlations

1. Correlation between label and attributes (classic IR hypothesis)
2. Correlation between label and labels and attributes of known neighbors
3. Correlation between labels of unknown neighbors
**Definition**

**CC**: Combined classification of inter-linked objects using label-attribute correlations and label-label neighbor correlations.

A major difference to general classification is that inference for all unknown instances is simultaneous.
Inference

Definition

Given a joint distribution of the unknown labels, compute the marginal distribution for a single node’s label.

1. Exact inference is intractable for arbitrary networks.
3. Most research is focused on approximate inference.
A more formal view on the problem

1. The network structure is modeled as a graph \( G=(V,E) \).
2. Each node is a variable defined over a given domain.
3. \( V \) contains two types of variables: \( X \) and \( Y \)
4. Goal: Label the nodes in \( Y \)
1 Problem

2 Methods
   - Local methods
   - Global methods

3 Experiments
Local and global

- Collaborative classification
  - Local
    - ICA
    - Gibbs sampling
  - Global
    - Loopy belief propagation
    - Mean field
1 Problem

2 Methods
   - Local methods
   - Global methods

3 Experiments
Iterative classification algorithm (ICA)
ICA mechanics

1. Classify a node $Y_i$ based on its neighbors $N_i$
2. Use a local classifier $f(N_i)$ to compute the best value of $y_i$
3. Iteratively apply to all $Y_i$ using the best estimates of unknowns in $N_i$
4. Use the labeling that stabilizes over time
Gibbs sampling (GS)
Gibbs sampling - basic idea

1. Sample from a multivariate joint distribution (unknown explicitly)
2. Generates a series of samples based on conditional distributions of each variable
3. Example: Sample values from $f(X, Y)$
   - Start with initial $X = x_0$
   - Sample $y_0 = p(Y|X = x_0)$
   - Sample $x_1 = p(X|Y = y_0)$...
   - $(x_0, y_0), (x_1, y_1)...$ are samples from $p(X, Y)$ if $p(\ast|\ast)$ are the true conditionals
4. Simpler to sample from conditional distributions than to integrate over a joint (especially if the latter is unavailable)
Gibbs sampling for CC

1. The joint distribution is \( p(Y_1, Y_2, \ldots, Y_n) \)
2. Assume that we know the conditionals
   \( p(Y_k | Y_1 = y_1, \ldots, Y_{k-1} = y_{k-1}, Y_{k+1} = y_{k+1} \ldots) \)
3. Perform GS and estimate the marginals
   \( p(Y_i), Y_i \in Y \) based on the samples
Assume we know the conditionals?

1. Assume we can **estimate** the conditional $p(Y_i|N_i)$ using a local classifier

2. Assume independence of indirect neighbors $p(Y_i|N_i) = p(Y_i|Y)$

3. No guarantee that the estimated conditionals are the true conditionals
The mechanics of GS for CC

1. Initialize assignments of $Y_i$
2. Perform a "burn-in" number of sample steps
3. Sample and count label assignments
4. Estimate marginals based on counts. Decide on labels.
Feature construction for local classifiers

1. Classifiers normally require fixed-length FVs
2. Choice of aggregation - max, count, exists, etc.

Local classifiers (Decision trees, Log. Regression, SVM, etc.). Training.

3. Nodes ordering - robust to simple random, based on label diversity etc.

4. Performance (running time)
Feature construction

Aggregation: count, avg, exists, proportion, graph based, etc.
## Local classifiers

<table>
<thead>
<tr>
<th>Reference</th>
<th>local classifier used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neville &amp; Jensen [44]</td>
<td>naïve Bayes</td>
</tr>
<tr>
<td>Lu &amp; Getoor [35]</td>
<td>logistic regression</td>
</tr>
<tr>
<td>Jensen, Neville, &amp; Gallagher [25]</td>
<td>naïve Bayes, decision trees</td>
</tr>
<tr>
<td>Macskassy &amp; Provost [36]</td>
<td>naïve Bayes, logistic regression, weighted-vote relational neighbor, class distribution relational neighbor</td>
</tr>
<tr>
<td>McDowell, Gupta, &amp; Aha [39]</td>
<td>naïve Bayes, k-nearest neighbors</td>
</tr>
</tbody>
</table>
1 Problem

2 Methods
   • Local methods
   • Global methods

3 Experiments
Global methods

Collaborative classification

Local
- ICA
- Gibbs sampling

Global
- Loopy belief propagation
- Mean field
Additional notation

1. $L$ is the set of labels, $G(V, E)$ is the network of objects

2. Three types of clique potentials(distributions)

3. $\psi_i$ for each $Y_i \in Y$ is a mapping $\psi_i : L \rightarrow R^+$

4. $\psi_{ij}$ for each $(Y_i, X_j) \in E$ is a mapping $\psi_{ij} : L \rightarrow R^+$

5. $\eta_{ij}$ for each $(Y_i, Y_j) \in E$ is a mapping $\eta_{ij} : L \times L \rightarrow R^+$
Back to our example
"Known" potential of a label $y_i$

$\phi_i(y_i) = \psi_i(y_i) \sum_{(Y_i, X_j) \in E} \psi_{ij}(y_i)$

It is computed without considering "unknown" neighbors
Back to our example

\[ y_1 \quad \psi_{13} \quad y_1 \quad \psi_{15} \quad y_1 \quad \psi_{18} \]

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SH</td>
<td>0.6</td>
<td>SH</td>
<td>0.1</td>
<td>SH</td>
</tr>
<tr>
<td>CH</td>
<td>0.4</td>
<td>CH</td>
<td>0.9</td>
<td>CH</td>
</tr>
</tbody>
</table>

\[ \Phi_1 = \psi_1 \ast \psi_{13} \ast \psi_{14} \ast \psi_{16} \ast \psi_{18} = \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y_1</td>
<td>\Phi_1</td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>0.0096</td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0.0216</td>
<td></td>
</tr>
</tbody>
</table>

\[ \Phi_2 = \psi_2 \ast \psi_{25} \ast \psi_{26} = \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y_2</td>
<td>\Phi_2</td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0.405</td>
<td></td>
</tr>
</tbody>
</table>

\[ y_1 \quad y_2 \quad \eta_{12} \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SH</td>
<td>SH</td>
<td>0.9</td>
</tr>
<tr>
<td>SH</td>
<td>CH</td>
<td>0.1</td>
</tr>
<tr>
<td>CH</td>
<td>SH</td>
<td>0.1</td>
</tr>
<tr>
<td>CH</td>
<td>CH</td>
<td>0.9</td>
</tr>
</tbody>
</table>

\[ y_2 \quad \psi_2 \quad y_2 \quad \psi_{26} \quad y_2 \quad \psi_{25} \]

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y_2</td>
<td>\psi_2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y_2</td>
<td>\psi_{25}</td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>
A pairwise MRF is given by the pair $\langle G(V, E), \Psi \rangle$, $G$ is a graph, $\Psi$ is a set of potentials $\psi, \eta, \phi$.
For an assignment $y$ of all $Y$ the MRF is associated with

$$P(y|x) = \alpha \prod_{y_i \in Y} \phi_i(y_i) \prod_{(y_i, y_j) \in E} \eta_{ij}(y_i, y_j)$$
The MRF defines a joint p.d.f. of all "unknown" labels.

Each $P(y|x)$ is the probability of a given world $y$.

Same as before obtaining the marginal for $P(Y_i = y_i)$ would require summing over exponential number of terms.

$\#P$ problem $\rightarrow$ approximation.
Instead of working with the actual distribution defined by the MRF, work with an approximate "trial" distribution.

The "trial" distribution should be simpler (to compute/store).

It should be easier to extract marginals from the "trial" distribution.

The "trial" should be fitted to the actual distribution.
Loopy belief propagation (LBP)
Loopy belief propagation is defined on a pairwise MRF. It is a discrete time message passing algorithm. At each step a message $m_{i \rightarrow j}(y_j)$ is passed from unknown node $Y_i$ to $Y_j$.

$$m_{i \rightarrow j}(y_j) = \alpha \sum_{y_i \in L} \eta_{i,j}(y_i, y_j) \phi_i(y_i) \prod_{Y_k \in N_i \cap Y \setminus Y_j} m_{k \rightarrow i}(y_i)$$
LBP example

\[ \Phi_1 = \Psi_1 \cdot \Psi_{13} \cdot \Psi_{16} \cdot \Psi_{18} = \begin{array}{cc}
\text{y}_1 & \Phi_1 \\
\text{SH} & 0.0096 \\
\text{CH} & 0.0216
\end{array} \]

\[ \Phi_2 = \Psi_2 \cdot \Psi_{25} \cdot \Psi_{26} = \begin{array}{cc}
\text{y}_2 & \Phi_2 \\
\text{SH} & 0.005 \\
\text{CH} & 0.405
\end{array} \]

Message \( i \rightarrow j \)

\[ m_{i \rightarrow j}(y_j) = \alpha \sum_{y_i \in L} \eta_{ij}(y_i, y_j) \phi_i(y_i) \prod_{Y_k \in N_i \cup Y \setminus y_j} m_{k \rightarrow i}(y_i) \]
LBP mechanics

1. Initially all messages are set to 1
2. Perform message passing until messages stabilize
3. Compute beliefs
   \[ b_i(y_i) = \alpha \phi_i(y_i) \prod_{Y_j \in N_i \cap Y} m_{j \rightarrow i}(y_i) \]
4. \( b_i(y_i) \) is the approximation of the marginal probability of \( y_i \) for node \( Y_i \)
Relaxation labeling via mean-field (MF)
1. MF is defined on MRF
2. MF can be described by the following fixed point equation:
   \[ b_i(y_i) = \alpha \phi_i(y_i) \prod_{Y_j \in N_j \cap Y} \prod_{y_j \in L} \eta_{ji}^{b_j(y_j)}(y_i, y_j) \]
3. Iterative method for computing the fixed point equation
Outline

1 Problem

2 Methods
   - Local methods
   - Global methods

3 Experiments
Experiments

1. Comparison of content-based (CO) and CC classification
2. Comparison of local classifiers for Local CC. Logistic regression (LR) versus Naive Bayes (NB)
3. Comparison of Global and Local CC
4. Eight different classifiers:
   1. CO + NB/LR
   2. ICA + NB/LR
   3. GS + NB/LR
   4. LBP
   5. MF
Experimental setup

1. **Real world data**
   1. CORA - $|V| = 2708, |E| = 5429, |L| = 7$
   2. Citeseer - $|V| = 3312, |E| = 4732, |L| = 6$

2. **Synthetic data** $|V| = 1000, |L| = 5$

3. Varying homophily and link density for synthetic data

4. 10-fold cross validation
Choice of features

1. Document terms for both CO and local CC methods
2. Count aggregation of terms
3. MRF with clique and node potentials for Global CC
Sampling for fold validation

1. Create folds for training and evaluation
2. "Snowball sampling" (SS) evaluation
   1. Select a random core node
   2. Expand, choosing a node based on the class distribution
   3. Expand $|X|/k$ times
   4. Create split.
   5. Use the $|X|/k$ sample for testing and the rest for training
3. Random sampling (RS) - Partition $|X|$ in $k$ folds randomly
Sampling challenges

1. SS may result in one and the same node appearing in multiple folds
2. Average the accuracy of each instance and than average over all training
3. Matched (M) average accuracy - only for instances that appear in at least one SS split
Learning the parameters

1. For CO and Local CC - local classifiers parameters
2. For MF and LBP - clique potentials
3. Gradient-based optimization approaches on the labeled nodes in the training splits
### Experimental results - real-world datasets

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cora</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SS</td>
<td>RS</td>
<td>M</td>
<td>SS</td>
<td>RS</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>CO-NB</td>
<td>0.7285</td>
<td>0.7776</td>
<td>0.7476</td>
<td>0.7427</td>
<td>0.7487</td>
<td>0.7646</td>
<td></td>
</tr>
<tr>
<td>ICA-NB</td>
<td>0.8054</td>
<td>0.8478</td>
<td>0.8271</td>
<td>0.7540</td>
<td>0.7683</td>
<td>0.7752</td>
<td></td>
</tr>
<tr>
<td>GS-NB</td>
<td>0.7613</td>
<td>0.8404</td>
<td>0.8154</td>
<td>0.7596</td>
<td>0.7680</td>
<td>0.7737</td>
<td></td>
</tr>
<tr>
<td>CO-LR</td>
<td>0.7356</td>
<td>0.7695</td>
<td>0.7393</td>
<td>0.7334</td>
<td>0.7321</td>
<td>0.7532</td>
<td></td>
</tr>
<tr>
<td>ICA-LR</td>
<td>0.8457</td>
<td>0.8796</td>
<td>0.8589</td>
<td>0.7629</td>
<td>0.7732</td>
<td>0.7812</td>
<td></td>
</tr>
<tr>
<td>GS-LR</td>
<td>0.8495</td>
<td>0.8810</td>
<td>0.8617</td>
<td>0.7574</td>
<td>0.7699</td>
<td>0.7843</td>
<td></td>
</tr>
<tr>
<td>LBP</td>
<td>0.8554</td>
<td>0.8766</td>
<td>0.8575</td>
<td>0.7663</td>
<td>0.7759</td>
<td>0.7843</td>
<td></td>
</tr>
<tr>
<td>MF</td>
<td>0.8555</td>
<td>0.8836</td>
<td>0.8631</td>
<td>0.7657</td>
<td>0.7732</td>
<td>0.7888</td>
<td></td>
</tr>
</tbody>
</table>

1. **CC dominates CO**
Experimental results - real-world datasets

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cora</th>
<th>Citeseer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SS</td>
<td>RS</td>
</tr>
<tr>
<td>CO-NB</td>
<td>0.7285</td>
<td>0.7776</td>
</tr>
<tr>
<td>ICA-NB</td>
<td>0.8054</td>
<td>0.8478</td>
</tr>
<tr>
<td>GS-NB</td>
<td>0.7613</td>
<td>0.8404</td>
</tr>
<tr>
<td>CO-LR</td>
<td>0.7356</td>
<td>0.7695</td>
</tr>
<tr>
<td>ICA-LR</td>
<td>0.8457</td>
<td>0.8796</td>
</tr>
<tr>
<td>GS-LR</td>
<td>0.8495</td>
<td>0.8810</td>
</tr>
<tr>
<td>LBP</td>
<td>0.8554</td>
<td>0.8766</td>
</tr>
<tr>
<td>MF</td>
<td>0.8555</td>
<td>0.8836</td>
</tr>
</tbody>
</table>

1. CC dominates CO
2. LR dominates NB
## Experimental results - real-world datasets

1. CC dominates CO
2. LR dominates NB
3. ICA and GS comparable by accuracy

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cora</th>
<th>Citeseer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SS</td>
<td>RS</td>
</tr>
<tr>
<td>CO-NB</td>
<td>0.7285</td>
<td>0.7776</td>
</tr>
<tr>
<td>ICA-NB</td>
<td>0.8054</td>
<td>0.8478</td>
</tr>
<tr>
<td>GS-NB</td>
<td>0.7613</td>
<td>0.8404</td>
</tr>
<tr>
<td>CO-LR</td>
<td>0.7356</td>
<td>0.7695</td>
</tr>
<tr>
<td>ICA-LR</td>
<td>0.8457</td>
<td>0.8796</td>
</tr>
<tr>
<td>GS-LR</td>
<td>0.8495</td>
<td>0.8810</td>
</tr>
<tr>
<td>LBP</td>
<td>0.8554</td>
<td>0.8766</td>
</tr>
<tr>
<td>MF</td>
<td>0.8555</td>
<td>0.8836</td>
</tr>
</tbody>
</table>
### Experimental results - real-world datasets

1. CC dominates CO
2. LR dominates NB
3. ICA and GS comparable by accuracy
4. Slight dominance of Global over Local

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cora</th>
<th>Citeseer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SS</td>
<td>RS</td>
</tr>
<tr>
<td>CO-NB</td>
<td>0.7285</td>
<td>0.7776</td>
</tr>
<tr>
<td>ICA-NB</td>
<td>0.8054</td>
<td>0.8478</td>
</tr>
<tr>
<td>GS-NB</td>
<td>0.7613</td>
<td>0.8404</td>
</tr>
<tr>
<td>CO-LR</td>
<td>0.7356</td>
<td>0.7695</td>
</tr>
<tr>
<td>ICA-LR</td>
<td>0.8457</td>
<td>0.8796</td>
</tr>
<tr>
<td>GS-LR</td>
<td>0.8495</td>
<td>0.8810</td>
</tr>
<tr>
<td>LBP</td>
<td>0.8554</td>
<td>0.8766</td>
</tr>
<tr>
<td>MF</td>
<td>0.8555</td>
<td>0.8836</td>
</tr>
</tbody>
</table>
Experimental results - synthetic datasets

![Graph showing accuracy vs. homophily for different methods and datasets.
Methods: CO-NB, ICA-NB, GS-NB, CO-LR, ICA-LR, GS-LR, LBP, MF.
X-axis: Homophily, Y-axis: Accuracy.]
Experimental results - synthetic datasets

The graph shows the accuracy of different models (CO-NB, ICA-NB, GS-NB, CO-LR, iCA-LR, GS-LR, LBP, MF) as a function of link density. The accuracy is measured on a scale from 0 to 1, with higher values indicating better performance. The link density varies from 0.1 to 0.9 on the x-axis.
Practical observations

1. MF and LBP are hard to work with. Initialization and convergence issues.
2. ICA is faster than GS (14m vs. 3h on Citeseer with NB)
3. ICA converges in <10 iterations, while GS requires 200 "burn-in" + 800 samples