Learning Sparse Markov Network Structure via Ensemble-of-Trees Models

Yuanqing Lin, Shenghuo Zhu
NEC Laboratories America

Daniel D. Lee\textsuperscript{a}, Ben Taskar\textsuperscript{b}
\textsuperscript{a} Department of Electrical and Systems Engineering
\textsuperscript{b} Department of Computer and Information Science
University of Pennsylvania
Introduction (1) -- why **sparse** Markov Network

A gene regulatory network  
(http://www.pnas.org/content/104/31/12890/F2.large.jpg)

- **Sparse structure:**
  1) real world are sparsely connected (e.g. genes, people, words)
  2) sparseness in Markov network encodes conditional independence
  3) discovering sparse patterns can be important for knowledge discovery
Introduction (2) -- existing work

- Low tree-width Markov network
  - Tree Markov network (Chow & Liu, 1968)
  - Thin tree width (Bach & Jordan, 2001; Srebro, 2001; Chechetka & Guestrin, 2007)

- Exact inference (but learning is approximate and very expensive)

- Gaussian Markov networks
  \[ X^* = \arg \min_{X>0} -\log \det(X) + \text{Tr}(S^T X) + \lambda \sum_{u \neq v} |X_{uv}| \]
  (Banerjee et al., 2006; Friedman et al., 2007)

- Convex, easy to optimize (1000x1000 problem solved in minutes)

- General Markov networks
  see next two slides …

- Limited in modeling practical data
Optimization is convex – general extension of Gaussian MRF

Partition function (and inference) is intractable in general

Pseudo-likelihood approach:

\[
\mathbf{w}_{/i}^* = \arg \min_{\mathbf{w}_{/i}} - \log P(x_i | x_{/i}, \mathbf{w}_{/i}) + \lambda \| \mathbf{w}_{/i} \|
\]

- Pseudo negative log-likelihood
- \(l_1\)-norm sparsity regularization


😊 Each individual optimization is convex (and simple)

😄 Pseudo-likelihood: requires a large amount of data for achieving good estimates
**Ensemble-of-trees (ET) model (1)**

--- probabilistic model

**Data likelihood:**

\[ P(x) = \sum_{T} P(x|T)P(T) \]

sum over *all possible* spanning trees

**Tree probability:**

\[ P(T) = \frac{1}{Z} \prod_{\{u,v\} \in T} \beta_{u,v} \]

where \( Z = \sum_{T} \prod_{\{u,v\} \in T} \beta_{u,v} \)

**ET model:**

- mixture of (all possible spanning) trees model
- partition function and data likelihood in closed form
ET model (2) -- partition function

Partition function: \[ Z = \sum_T \prod_{\{u,v\} \in T} \beta_{u,v} \]

Matrix tree theorem:

\[ \beta_{u,v} = \begin{cases} 1 & \text{if } \{u,v\} \text{ is an edge} \\ 0 & \text{otherwise} \end{cases} \]

\[ \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{bmatrix} \]

\[ B(\beta) = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \]

\[ Z = \det[Q(\beta)] \]

- Generalized matrix tree theorem: \( \beta_{u,v} \) can be any nonnegative number

- Partition function: has a closed form

- \( \beta \): encodes the structure of a graph
ET model (3) -- data likelihood

Data likelihood:

\[
P(x) = \sum_T \prod_{\{u,v\}\in T} \frac{1}{Z} \prod_u w_u,v \beta_{u,v} \prod_{\{u,v\}\in T} w_{u,v} \beta_{u,v} = \prod_u \frac{1}{Z} \sum_T \prod_{\{u,v\}\in T} w_{u,v} \beta_{u,v}
\]

\[
= \det[Q(\beta \otimes w)] \prod_u w_{0u} \frac{1}{\det[Q(\beta)]}
\]

Data likelihood on a tree:

\[
P(x|T) = \prod_{\{u,v\}\in T} P(X_u = x_u, X_v = x_v) \prod_u P(X_u = x_u)
\]

\[
= \prod_{\{u,v\}\in T} w_{u,v} \prod_u w_{0u}
\]

Data likelihood: has a closed form
ET model (4) -- ML estimation

Given data \( \{x_i^i\}_{i=1}^N \)

\[
\beta^* = \arg \min_{\beta} - \sum_{i=1}^N \log \det[Q(\beta \otimes w_i)] + N \log \det[Q(\beta)]
\]

Subject to: \( \beta_{u,v} \geq 0 \)

\[
\sum_{u,v:u \neq v} \beta_{u,v} = 1
\]

- **Marginals (and \( w_i \))**: estimated directly from data
- **Sparse solutions**: tree number regularization
ET model (5) -- nonconvex optimization

Initialization (by solving a convex upper bound):

\[
\beta^* = \arg \min_{\beta} - \sum_{i=1}^{N} \log \det [Q(\beta \otimes w_i)] + c
\]

Subject to: \( \beta_{u,v} \geq 0 \)

\[
\sum_{u,v:u\neq v} \beta_{u,v} = 1
\]

◆ Optimizations: solved by projected gradient descent (PGD)
Projected gradient descent:

\[
\tilde{\beta} \leftarrow \beta^k - \lambda \nabla \beta
\]

\[
\beta^{k+1} \leftarrow \mathcal{P}(\tilde{\beta})
\]

Step size \(\lambda\): determined by Armijo’s rule

Projection operation \(\mathcal{P}(\tilde{\beta})\): need to solve a quadratic programming

\[
\beta^{k+1} = \arg \min_{\beta} \| \beta - \tilde{\beta} \|_2^2
\]

Subject to:

\[
\beta_{u,v} \geq 0
\]

\[
\sum_{u,v: u \neq v} \beta_{u,v} = 1
\]

◆ PGD (with Armijo’s rule):
-- guaranteed to converge to a local optimizer

ET model (6) -- nonconvex optimization (cont’d)
Simulations (1) -- Gaussian Markov network

- **L1-norm regularized GMN vs. ET model:**

1) ET provides better performance when only a small number of data samples is available

2) ET does NOT need to assume Gaussian model

3) ET is free of $l_1$-norm regularization
Simulations (2) -- binary Markov network

ET model vs. pseudo-likelihood approach:

Again, ET provides better performance when only a small number of data samples is available.
Results of learning a word network

<table>
<thead>
<tr>
<th>apple</th>
<th>iphone</th>
<th>mac</th>
<th>ipod</th>
<th>itunes</th>
<th>store</th>
<th>nasdaq</th>
</tr>
</thead>
<tbody>
<tr>
<td>iphone</td>
<td>apple</td>
<td>apple</td>
<td>apple</td>
<td>apple</td>
<td>shop</td>
<td>newsletter</td>
</tr>
<tr>
<td>mac</td>
<td>ipod</td>
<td>os</td>
<td>apple</td>
<td>download</td>
<td>retail</td>
<td>e-mail</td>
</tr>
<tr>
<td>microsoft</td>
<td>mobile</td>
<td>pc</td>
<td>touch</td>
<td>music</td>
<td>apple</td>
<td>analyst</td>
</tr>
<tr>
<td>ipod</td>
<td>device</td>
<td>windows</td>
<td>itunes</td>
<td>ipod</td>
<td>purchase</td>
<td>blackberry</td>
</tr>
<tr>
<td>itunes</td>
<td>phone</td>
<td>macbook</td>
<td>music</td>
<td>amazon</td>
<td>music</td>
<td>expectation</td>
</tr>
<tr>
<td>store</td>
<td>rumor</td>
<td>user</td>
<td>device</td>
<td>podcast</td>
<td>price</td>
<td>trend</td>
</tr>
<tr>
<td>pc</td>
<td>blackberry</td>
<td>hardware</td>
<td>song</td>
<td>iphone</td>
<td>download</td>
<td></td>
</tr>
<tr>
<td>user</td>
<td>smartphone</td>
<td>computer</td>
<td>dvd</td>
<td>library</td>
<td>letter</td>
<td></td>
</tr>
<tr>
<td>device</td>
<td>pc</td>
<td>leopard</td>
<td>stream</td>
<td>item</td>
<td>print</td>
<td></td>
</tr>
<tr>
<td>hardware</td>
<td>gps</td>
<td>desktop</td>
<td>selling</td>
<td>network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>computer</td>
<td>user</td>
<td>safari</td>
<td>amazon</td>
<td>directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>macbook</td>
<td>nokia</td>
<td>pro</td>
<td>storage</td>
<td>responsibility</td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td>store</td>
<td>graphics</td>
<td>backup</td>
<td>hp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>upgrade</td>
<td>wireless</td>
<td>vista</td>
<td>package</td>
<td>portfolio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mobile</td>
<td>gadget</td>
<td></td>
<td></td>
<td>demonstration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>app</td>
<td></td>
<td></td>
<td>apple</td>
<td></td>
<td></td>
</tr>
<tr>
<td>leopard</td>
<td>australia</td>
<td></td>
<td></td>
<td>microsoft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nasdaq</td>
<td>carrier</td>
<td></td>
<td></td>
<td>exchange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>safari</td>
<td>samsung</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maker</td>
<td>touch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>apps</td>
<td>launch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>smartphone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>competitor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ET model:** is able to discover meaningful word network from documents
The ET model provides a novel paradigm for MRF structure learning

- both partition function and data likelihood are tractable
- ET model is versatile for different distributions: continuous or discrete, Gaussian or non-Gaussian, different tree-width …
- Empirical results show that ET model provides competitive performance compared to the state-of-art methods

We developed a projected gradient algorithm for efficiently solving the optimization arising in the ET model

Future work: theoretical analysis, more efficient algorithm (especially stochastic algorithm) for the ET model