Best Choice Edge Grafting For Efficient Learning of Markov Random Fields

Walid Chaabene, Bert Huang

Virginia Tech

walidch, bhuang @vt.edu

December 11, 2018
Overview

Introduction

Classical Structure Learning Methods

Best Choice Edge Grafting

Results

Conclusion
Outline

Introduction

Classical Structure Learning Methods

Best Choice Edge Grafting

Results

Conclusion
Introduction

Pairwise Markov Random Fields
A graphical model that represents joint probability distributions.

\[ G(V, E) : \begin{cases} 
V : \text{set of } n \text{ nodes (variables)}; \\
E : \text{set of edges (parametric interactions)}. 
\end{cases} \]

\[ p_w(X) = \frac{1}{Z(w)} \prod_{i \in V} \phi_i(x; w) \prod_{(i,j) \in E} \phi_{ij}(x; w), \tag{1} \]

where:

\[ \phi_c(x; w) = \exp \left( \sum_{k \in c} w_k f_k(x) \right) = \exp \left( w^\top f(x) \right). \tag{2} \]

\( f_k \) : state indicator functions (assigned one parameter each). For example:

\[ f_{k_{x_1=1}} = \begin{cases} 
1 & \text{if } x_1 = 1 \\
0 & \text{otherwise.} 
\end{cases} \]

\[ f_{k_{x_1=0, x_2=1}} = \begin{cases} 
1 & \text{if } x_1 = 0 \text{ and } x_2 = 1 \\
0 & \text{otherwise.} 
\end{cases} \]
Introduction

Structure learning problem:
Given $N$ observations of $n$ variables ($V$), find all relevant edges ($E$) and estimate their corresponding parameters.

Challenges
- $n$ variables $\Rightarrow O(n^2)$ possible edges.
- Learning requires large datasets.

This work
- Investigate major computational bottlenecks of $\ell_1$-based learning techniques of Markov Random Fields.
- Propose scalable structure learning approach with controllable trade-off between learning speed and quality.
Outline

Introduction

Classical Structure Learning Methods

$\ell_1$-Based Learning
Feature Grafting

Best Choice Edge Grafting

Results

Conclusion
**ℓ₁-Based Learning**

Minimizing ℓ₁-Regularized Negative Log-Likelihood

\[
L(\mathbf{w}) = -\frac{1}{N} \sum_{m=1}^{N} \log p_{\mathbf{w}}(x^{(m)}) = -\frac{1}{N} \sum_{m=1}^{N} (\mathbf{w}^T f(x^{(m)})) + \log Z(\mathbf{w})
\] (3)

\[
\mathbb{L}(\mathbf{w}) = L(\mathbf{w}) + \lambda ||\mathbf{w}||_1
\] (4)

\[
\min_{\mathbf{w}} \mathbb{L}(\mathbf{w})
\] (5)

\[
\delta_k L = -\frac{1}{N} \sum_{m=1}^{N} f_k(x^{(m)}) + E_{\mathbf{w}}[f_k(x)] = E_{\mathbf{w}}[f_k(x)] - E_D[f_k(x)]
\] (6)

**Limitation:**

- \(E_{\mathbf{w}}[f_k(x)]\) : performs inference at each gradient step (Message passing methods are expensive on fully graphs).
- \(E_D[f_k(x)]\) : requires pre-computing data expectations of each possible state (sufficient statistics).
Feature Grafting\(^1\)

**Idea**
Assume that all variables are independent and iteratively activate parameters (introduce dependency).

**Approach**
Active-set method: a working set \(S\) and a search set \(F\).

- \(S = \{\text{unary parameters}\}; F = \{\text{pairwise parameters}\}\).

- Alternate between two steps until convergence:
  - *Step 1*: Optimizing over the active set \(S\) using a sub-gradient method.
  - *Step 2*: Select top violating parameter from \(F\) and add to \(S\).

- Feature Activation Condition:

\[
\text{KKT optimality condition: } \begin{cases} 
\delta_k L = 0 \text{ if } w_k \neq 0 \\
|\delta_k L| \leq \lambda \text{ if } w_k = 0 
\end{cases} \quad (7)
\]

\[
\Rightarrow C_1 : j = \arg \max_k |\delta_k L| \quad \text{s.t. } |\delta_k L| > \lambda \quad (8)
\]

\(^1\)Lee et al, 2007
Feature Grafting

$t_0 : S = \emptyset$
$t_1 : S = \{ w_{x_1=0,x_4=1} \}$
$t_2 : S = \{ w_{x_1=0,x_4=1}, w_{x_1=1,x_4=1} \}$
$t_3 : S = \{ w_{x_1=0,x_4=1}, w_{x_1=1,x_4=1}, w_{x_1=1,x_4=0} \}$
$t_4 : S = \{ w_{x_1=0,x_4=1}, w_{x_1=1,x_4=1}, w_{x_1=1,x_4=0}, w_{x_1=0,x_4=0} \}$
$t_5 : S = \{ w_{x_1=0,x_4=1}, w_{x_1=1,x_4=1}, w_{x_1=1,x_4=0}, w_{x_1=0,x_4=0}, w_{x_2=1,x_5=0} \}$
$t_{40} : S = S^*$
Feature Grafting

**Algorithm 1 Grafting**

1: Initialize $\mathcal{F} = \{\text{set of all pairwise parameters}\}$
2: Compute sufficient statistics of $f \ \forall f \in \mathcal{F}$  \quad \# cost: $O(n^2 N s_{\text{max}}^2)$
3: **repeat**
4: Select the top violating feature $f^*$  \quad \# cost: $O(n^2 s_{\text{max}}^2)$
5: Activate $f^*$
6: Optimize the $\ell_1$-regularized $L$ over the active set
7: **until** convergence

**Limitations:**

- Parameters are treated as one homogeneous group. No structure information is used.
- Requires computing $O(n^2 N s_{\text{max}}^2)$ sufficient statistics and performing $O(n^2 s_{\text{max}}^2)$ parameter activation tests.
Outline

Introduction

Classical Structure Learning Methods

Best Choice Edge Grafting
   Edge Grafting
   Best Choice Edge Grafting
   Complexity analysis

Results

Conclusion
Edge Grafting

Problem reformulation: Grafting Edges

- Redefine the search space: \( F = \{\text{Edge-wise parameter groups}\} \)
- Introduce groups sparsity regularization in the loss function.

\[
\mathbb{L}(\mathbf{w}) = L(\mathbf{w}) + \sum_{g \in G} \lambda d_g \|\mathbf{w}_g\|_2 + \lambda_2 \|\mathbf{w}\|_2^2, \tag{9}
\]

where \( g \) refers to either a node or an edge and \( d_g \) compensates for different groups’ cardinalities.

\[
\min_{\mathbf{w}} \mathbb{L}(\mathbf{w}) \tag{10}
\]

KKT optimality condition:

\[
\begin{cases} 
\left\| \delta_g L \right\|_2 + \lambda_2 \|\mathbf{w}_g\|_2^2 = 0 & \text{if } \|\mathbf{w}_g\|_2 \neq 0 \\
\left\| \delta_g L \right\|_2 \leq \lambda & \text{if } \|\mathbf{w}_g\|_2 = 0
\end{cases} \tag{11}
\]
**Edge Grafting**

**Grafting Edges**

- Edge score:

\[
se = \frac{||\delta_e L||_2}{de}
\]  

(12)

- Group-wise gradient (pairwise probability error between model and data observations):

\[
\delta_e L = \hat{p}_w(e) - p_D(e)
\]  

(13)

- Necessary edge activation condition:

\[
C_2 : \arg \max_e |se| \quad s.t. \quad se > \lambda
\]  

(14)

**Limitations:** Requires computing \(O(n^2 N s_{max}^2)\) sufficient statistics and performing \(O(n^2)\) edge activation tests.
**Introduction**

**Classical Structure Learning Methods**

**Best Choice Edge Grafting**

**Results**

**Conclusion**

---

**Edge Grafting**

\[ t_0 : S = \emptyset \]

\[ t_1 : S = \{ w_{x_1 = 0,x_4 = 1}, w_{x_1 = 1,x_4 = 1}, w_{x_1 = 1,x_4 = 0}, w_{x_1 = 0,x_4 = 0} \} \]

\[ t_2 : S = \{ w_{x_1 = 0,x_4 = 1}, w_{x_1 = 1,x_4 = 1}, w_{x_1 = 1,x_4 = 0}, w_{x_1 = 0,x_4 = 0}, w_{x_2 = 0,x_5 = 1}, w_{x_2 = 1,x_5 = 1}, w_{x_2 = 1,x_5 = 0}, w_{x_2 = 0,x_5 = 0} \} \]

\[ t_{10} : S = S^* \]
Best Choice Edge Grafting

Best Choice Problem
Given a set of streaming candidates, make a decision without testing all possible ones. Similar to a hiring process.

Best Choice Edge Grafting Mechanism
- On-demand edge sufficient statistics computation.
- Reduced number of activation tests

Figure: High-level operational scheme of the edge activation mechanism.
Reservoir Sampling

**Benefits of reservoir sampling** We simulate the behavior in finite settings, sampling $|R|$ ranks from the list of all possible numbers from 1 to $\binom{n}{2}$ and taking the minimum.

**Figure:** Simulated edge ranks using the reservoir. (50 nodes).

![Graph showing simulated edge ranks using the reservoir](image)

**Two extremes**
- **First Hit** ($|R| = 1$) $\rightarrow$ Bad quality edges.
- **Edge Grafting** (using an unlimited reservoir) $\rightarrow$ Negligible gains over a small reservoir.
Reservoir Sampling

Reservoir management

- Before \( t_{\text{max}} \) is reached:
  - If reservoir full: replace minimum scoring edge \( R_{\text{min}} \) with incoming edge \( e \) if \( s_{R_{\text{min}}} < s_e \).
- When \( t_{\text{max}} \) is reached:
  - Compute mean reservoir scores:
    \[
    \mu = \frac{1}{|R|} \sum_{e \in R} s_e \tag{15}
    \]
  - Activation threshold as:
    \[
    \tau_\alpha = (1 - \alpha) \mu + \alpha \max_{e \in R} s_e, \tag{16}
    \]
    where \( \alpha \in [0, 1] \) controls a trade-off between quality of added edges and speed of edge activation.
Search Space Reorganization

Reorganizing search space

- Search History:
  - Edge violation offset $v_e$:
    \[ v_e = 1 - \frac{s_e}{\lambda} \]  \hspace{1cm} (17)
  - Store failing edges in $L$ and refill $pq$ when it is empty:
    \[ pq[e] = v_e \]  \hspace{1cm} (18)

- Partial structure information:
  - Idea: Promote a scale-free structure.
  - Detect hubs using degree centrality:
    \[ c_i = \frac{|N_i|}{|V| - 1} \]  \hspace{1cm} (19)
  - Construct Hub set:
    \[ H = \{ i \in V \text{ such that } c_i > \hat{c} \} \]  \hspace{1cm} (20)
  - Prioritizing edges incident to hubs such that $\forall h \in H$ and $\forall n \in V$:
    \[ pq[(h, n)] = pq[(h, n)] - 1 \]  \hspace{1cm} (21)
Summary of Complexities

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Suff. stats. at $j^{th}$ edge</th>
<th>Activation step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature grafting</td>
<td>$O(n^2 Ns_{\text{max}}^2)$</td>
<td>$O(n^2 s_{\text{max}}^2)$</td>
</tr>
<tr>
<td>Edge grafting</td>
<td>$O(n^2 Ns_{\text{max}}^2)$</td>
<td>$O(n^2 s_{\text{max}}^2)$</td>
</tr>
<tr>
<td>Best choice edge grafting</td>
<td>$O((n + j t_{\text{max}}) Ns_{\text{max}}^2)$</td>
<td>$O(t_{\text{max}} s_{\text{max}}^2)$</td>
</tr>
</tbody>
</table>
Outline

Introduction

Classical Structure Learning Methods

Best Choice Edge Grafting

Results
  Synthetic Experiments
  Real Data Experiments

Conclusion
Synthetic Experiments

Synthetic Data

<table>
<thead>
<tr>
<th></th>
<th>200</th>
<th>400</th>
<th>600</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>200</td>
<td>400</td>
<td>600</td>
</tr>
<tr>
<td>Number of states per variable</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Number of parameters</td>
<td>498,500</td>
<td>1,997,000</td>
<td>4,495,500</td>
</tr>
</tbody>
</table>

- Scale-free-structures: Few dominant hubs.
- Data generated using Gibbs sampler: 20,000 data points from each network, randomly split into train and held-out testing sets.
**Synthetic Experiments**

**Synthetic results**

**Figure**: Full convergence of different methods (200 nodes).

\[
\tau_\alpha = (1 - \alpha) \mu + \alpha \max_{e \in R} s_e
\]
Synthetic Experiments

Synthetic results

Figure: Learning objectives vs time for varying MRFs sizes.

(a) 200 nodes and 600 edges

(b) 400 nodes and 1,200 edges

(c) 600 nodes and 1,800 edges
Synthetic Experiments

Synthetic results

**Figure**: Negative Log Pseudo-Likelihood vs time for varying MRFs sizes.

![Graphs showing negative log pseudo-likelihood vs time for different MRF sizes.](attachment:image.png)

(a) 200 nodes and 600 edges  
(b) 400 nodes and 1,200 edges  
(c) 600 nodes and 1,800 edges
Synthetic Experiments

Synthetic results

Figure: Role of structure heuristics in improving the quality of the learned MRF (200 nodes)
Real Data Experiments

Real data

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Jester</th>
<th>Yummly recipes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of variables</td>
<td>100</td>
<td>153</td>
</tr>
<tr>
<td>Number of States per variable</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Number of parameters</td>
<td>124, 250</td>
<td>36, 450</td>
</tr>
<tr>
<td>Dataset size</td>
<td>73, 421</td>
<td>10, 000</td>
</tr>
</tbody>
</table>

- Jester\(^2\): user ratings of jokes.
- Yummly recipes\(^3\): recipes with different ingredients.

\(^2\)[http://goldberg.berkeley.edu/jester-data/]

\(^3\)[https://www.kaggle.com/c/whats-cooking]
Real Data Experiments

Real data results

**Figure:** Negative Log Pseudo-Likelihood vs time for varying MRFs sizes.

(a) Yummly Objective

(b) Yummly NLPL
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Classical Structure Learning Methods</th>
<th>Best Choice Edge Grafting</th>
<th>Results</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

Outline

Introduction

Classical Structure Learning Methods

Best Choice Edge Grafting

Results

Conclusion
Conclusion

Proposed work
- Reformulate learning problem by introducing structure information.
- Avoid costly batch $\ell_1$-learning on the entire problem space. Informed edge search through reservoir sampling and search space reorganization.

Result
- Faster edge activation and convergence.
- Controllable trade-off between learning speed and quality.
- Achieved better scalability.

Limitations and future work
- Assumption of scale free structure: Investigate better structure heuristics for a more efficient search space reorganization.
- Applied on pairwise MRFs: Generalize approach for higher order MRFs.

Contact us: walidch@vt.edu; bhuang@vt.edu