On Learning Discrete Graphical Models Using Greedy Methods

Ali Jalali, Christopher C. Johnson and Pradeep Ravikumar
University of Texas at Austin

• Two Contributions ::

• Statistical estimation with \textbf{sparse} parameters :: analysis of forward-backward 
\textbf{greedy algorithm}; better than ell_1!

• Application to Discrete Graphical Model Selection

\textbf{Statistical Model ::} \quad X^{(i)} \sim P(X; \theta^*)

\textbf{Neg. Log-Likelihood ::} \quad \mathcal{L}(\theta; D) = \frac{1}{n} \sum_{i=1}^{n} - \log P(X^{(i)}; \theta)

\textbf{Sparsity Constrained MLE ::} \quad \min_{\theta: \|\theta\|_0 \leq k} \mathcal{L}(\theta; D)

\textbf{ell_1?} \quad \text{Bias; Restrictive Model Conditions}

\textbf{Non-Convex?}
Learning Discrete Graphical Models

• Discrete Random Variables \( X = (X_1, X_2, \ldots, X_p) \)

• Discrete Graphical Model :: \( P(X; \theta, \mathcal{G}) \propto \exp \left\{ \sum_{(s,t) \in E(\mathcal{G})} \theta_{st} \phi_{st}(x_s, x_t) \right\} \)

• Given :: \( n \) samples \( D := (X^{(1)}, \ldots, X^{(n)}) \) where \( X^{(i)} \sim P(X; \theta^*, \mathcal{G}) \)

• Problem :: Estimate underlying graph \( \mathcal{G} \)
Forward-Backward Greedy Algorithm

• Generalization of [T. Zhang, 2008] greedy algorithm for linear regression to general sparse statistical estimation

• Algorithm (Stopping Threshold $\epsilon$) ::

  ‣ Forward: Find best co-ordinate to add; add if improvement greater than $\epsilon$; set $\delta =$ amount of improvement

  ‣ Backward: Prune co-ordinates with loss-increase smaller than $\delta$

Theorem [Sparsistency]: Recovers support of true parameter, given restricted strong convexity, sufficient stopping threshold
Comparison: Learning Discrete Graphical Models

<table>
<thead>
<tr>
<th></th>
<th>ell_1</th>
<th>greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model Assumptions</strong></td>
<td>Irrepresentable / Incoherence</td>
<td>Restricted Strong Convexity</td>
</tr>
<tr>
<td><strong>Sample Complexity</strong></td>
<td>$n = \Omega(d^3 \log p)$</td>
<td>$n = \Omega(d^2 \log p)$</td>
</tr>
<tr>
<td><strong>Comp. Complexity</strong></td>
<td>$O(p^4)$</td>
<td>$O(d^3 p^2)$</td>
</tr>
</tbody>
</table>

Better in all respects!
i.e. don’t use ell_1 regularization; use greedy!

Oh, and
Information-theoretically Optimal
(Santhanam, Wainwright 08)