DSCI 573: Model Selection and Feature Selection

Structure Learning
Winter 2018
Structure Learning: Unsupervised Feature Selection

• “News” data: presence of 100 words in 16k newsgroup posts:

```
<table>
<thead>
<tr>
<th></th>
<th>car</th>
<th>drive</th>
<th>files</th>
<th>hockey</th>
<th>mac</th>
<th>league</th>
<th>pc</th>
<th>win</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```

• Which words are related to each other?

• Problem of structure learning: unsupervised feature selection.
Structure Learning: Unsupervised Feature Selection

• Optimal tree structure: (ignore arrows)
Naïve Approach: Association Networks

• A naïve approach to structure learning ("association networks"):  
  – For each pair of variables, compute a measure of similarity or dependence.

• Using these $n^2$ similarity values either:  
  – Select all pairs whose similarity is above a threshold.  
  – Select the “top k” most similar features to each feature ‘j’.

• Main problems:  
  – Usually, most variables are dependent (too many edges).  
    • "Sick" is getting connected to "Tuesdays" even if "tacos" are a variable.  
  – “True” neighbours may not have the highest dependence.  
    • "Sick" might get connected to "Tuesdays" before it gets connected to "milk".

• (Variation: best tree can be found as minimum spanning tree problem.)
Example: Vancouver Rain Data

• Consider modeling the “Vancouver rain” dataset.

|       | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 |...
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Month 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Month 2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Month 3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Month 4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Month 5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Month 6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

• The strongest signal in the data is the simple relationship:
  – If it rained yesterday, it’s likely to rain today (> 50% chance that $x_{t-1} = x_t$).
  – But an “association network” might connect all days (all dependent).
Dependency Networks

• A better approach is dependency networks:
  – For each variable ‘j’, make it the target in a supervised learning problem.
  – Now we can use any feature selection method to choose j’s “neighbours”.
    • Forward selection, L1-regularization, ensemble methods, etc.

\[
\begin{bmatrix}
  x^1 \\
  x^2 \\
  x^3 \\
  x^4 \\
  x^5 \\
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
  \tilde{x}^1 \\
  \tilde{x}^2 \\
  \tilde{x}^3 \\
  \tilde{x}^4 \\
  \tilde{x}^5 \\
\end{bmatrix}
\]

• Can capture conditional independence:
  – Might connect “sick” to “tacos”, and “tacos” to “Tuesdays” (w/o sick-tacos).
Dependency Networks

• Dependency network fit to Vancouver rain data (different $\lambda$ values):
Dependency Networks

• Variation on dependency networks on digit image pixels:

Another popular structure learning method is the "PC" algorithm.
Summary

• Structure learning is “unsupervised” feature selection.
• Association networks make graph by finding similar features.
• Dependency networks use feature selection with feature ‘j’ as ‘y’.