Learning Structure in Directed Graphical Models

CS 290C Graphical Models

Jay Kreps

The Problem

Given a set of random variables, $Z = \{Z_1, ..., Z_M\}$, and training data $D = \{(\vec{x_1}, y_1), ..., (\vec{x_N}, y_N)\}$ find the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and parameters Θ that produce the best predictions.

Difficulties

- The usual difficulty: Complex models have lower training error
- Some added difficulties
  - Given a set of variables the number of possible graphs is exponential
  - The number of parameters per random variable is locally exponential
  - If some nodes are hidden, we must simultaneously maximize over Θ *and* $\mathcal{G}$

Scoring

We have to punish model complexity!

- BIC
- MDL
- BDe
- etc.

All are of the form log-liklihood + penalty, and all are asymptotically the same.

Search

- Search space is super-exponential ($2^{n \log n}$)
- Evaluation is expensive
- Some common simplifications
  - Bound the in-degree of nodes
  - Assume an ordering over nodes
- Even with the in-degree bounded at 2, finding the best graph is NP-Complete

Complete Data

For the completely observed graphical model things are simpler:

$$p(x|\theta, \mathcal{G}) = p(x_1|\theta_1, \mathcal{G}) \cdot \dots \cdot p(x_M|\theta_M, \mathcal{G})$$

Since we needn't estimate any parameters, changing the graph has only local effect.

In this case we can easily find a local minimum in our scoring function.

Structural E.M.

Let $t = 0$, and initialize $\mathcal{G}$ and θ at random

Repeat until convergence:

1. Find $\mathcal{G}^{t+1}$ that maximizes $Q(\mathcal{G}^t, \theta^t)$
2. Set $\theta^{t+1} = \arg\max Q(\mathcal{G}^{t+1}, \theta)$

Where Q is some scoring function.

Other Tricks: Context Specific Independence

Reduce the cost of dependence by storing only some values.

- Use a linked instead of tabular representation for the conditional probability table
- Allows fine-grained control
- Has specialized EM and scoring functions
- Empirical results are good, but there are no guarantees

Conclusion

- Computationally expensive
- Decent results possible for small graphs, with lots of data
- Very messy, and no more principled methods in sight