Probabilistic Graphical Models (Cmput 651): Learning With Partial Data

Matthew Brown
17/11/2008

Reading: Koller-Friedman Ch. 18, 19.3.3

Space of topics

Semantics
Inference
Learning

Continuous
Discrete
Directed
Undirected
Learning Markov nets

<table>
<thead>
<tr>
<th></th>
<th>complete data</th>
<th>partial data</th>
</tr>
</thead>
<tbody>
<tr>
<td>known structure</td>
<td>easy (sort of)</td>
<td>hard</td>
</tr>
<tr>
<td>unknown structure</td>
<td>hard</td>
<td>very hard</td>
</tr>
</tbody>
</table>

This lecture

Outline

**Short review**
- Markov net parameter learning
- Structure learning
Observation models (KF 18.1.1)

\[ X = \{X_1, \ldots, X_n\} \]
\[ O_X = \{O_{X_1}, \ldots, O_{X_n}\} \quad \text{observability variables} \]

\[ P_{\text{missing}}(X, O_X) = P(X) \cdot P_{\text{missing}}(O_X \mid X) \]

Can complicate likelihood function
no closed form for Bayes nets
(whereas exists closed form for complete data)

Observation models (KF 18.1.1)

\[ X = \{X_1, \ldots, X_n\} \]
\[ O_X = \{O_{X_1}, \ldots, O_{X_n}\} \quad \text{observability variables} \]

\[ P_{\text{missing}}(X, O_X) = P(X) \cdot P_{\text{missing}}(O_X \mid X) \]

Decoupling between \( X \) and \( O_X \) makes likelihood decomposable:

Missing completely at random (MCAR) \( P_{\text{missing}} \models (X \perp O_X) \)
- allowed to ignore missing data

Missing at random (MAR) \( P_{\text{missing}} \models (o_{X} \perp x^y_{\text{hidden}} \mid x^y_{\text{obs}}) \)
conditional decoupling (see next page)
eg: medical tests setting
Observation models (KF 18.1.1)

**Theorem:**
If $P_{\text{missing}}$ satisfies MAR, the likelihood $L(\theta, \psi : \mathcal{D})$ can be decomposed into (written as product of)

$L(\theta : \mathcal{D})$ and $L(\psi : \mathcal{D})$

$X$ parameters $O_X$ parameters

Identifiability (KF 18.1.4)

Can we uniquely “pin down” the model?

**Identifiability:** $P(X|\theta) = P(X|\theta') \iff \theta = \theta'$

**Local identifiability:** $P(X|\theta) = P(X|\theta') \iff \theta = \theta'$

as long as $||\theta - \theta'|| < \epsilon$
Likelihood function with partial data (KF 18.1.3)

partial data $\rightarrow$ likelihood becomes marginal probability over unobserved variables

$$P(x_{obs}|\theta) = \int_{x_{missing}} P(x_{obs}, x_{missing}|\theta) dx_{missing}$$

i.e. sum of complete data likelihood functions

$\rightarrow$ multimodal (not concave)

Learning with partial data in Bayes nets

Likelihood
no closed form
non-concave

Iterative approaches:
Gradient methods
EM
  alternately compute expected values of unobserved variables and then maximize parameters in “complete” data fashion

Gibbs sampling
Outline

Short review

Markov net parameter learning

Structure learning

Learning Markov nets w/ partial data (KF 19.3.3)

MAR, MCAR, ...

Issues with identifiability

Likelihood

no closed form to begin with (even w/ complete data)
no longer concave w/ partial data
MN gradient ascent w/ partial data (KF 19.3.3.1)

**Gradient ascent:**
Assume MAR
\( o[m] \) = observed entries in \( m \)th data point
\( H[m] \) = missing entries in \( m \)th data point
\((o[m], h[m])\) = complete assignment to \( X \)

Average log-likelihood
\[
\frac{1}{M} \ln P(D | \theta) = \frac{1}{M} \ln \left( \sum_{m=1}^{M} \sum_{h[m]} P(o[m], h[m] | \theta) \right) \\
= \frac{1}{M} \ln \left( \sum_{m=1}^{M} \sum_{h[m]} \tilde{P}(o[m], h[m] | \theta) \right) - \ln Z
\]

(\( \sim \) means unnormalized)

MN gradient ascent w/ partial data (KF 19.3.3.1)

Average log-likelihood
\[
\frac{1}{M} \ln P(D | \theta) = \frac{1}{M} \ln \left( \sum_{m=1}^{M} \sum_{h[m]} P(o[m], h[m] | \theta) \right) \\
= \frac{1}{M} \ln \left( \sum_{m=1}^{M} \sum_{h[m]} \tilde{P}(o[m], h[m] | \theta) \right) - \ln Z
\]

(\( \sim \) means unnormalized)

\[
\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta : D) = \frac{1}{M} \left[ \sum_{m=1}^{M} E_{h[m]} P(h[m]|o[m], \theta)[\phi_i] \right] - E_{\theta}[\phi_i]
\]

- Expectation conditional on observation \( m \)
- \( M \) inferences
- 1 inference
**MN gradient ascent w/ partial data (KF 19.3.3.1)**

\[
\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta : D) = \frac{1}{M} \left[ \sum_{m=1}^{M} E_{h[m] \sim P(h[m]|o[m], \theta)}[\phi_i] \right] - E_{\theta}[\phi_i]
\]

For each data point m, estimate \( P(h[m] | o[m], \theta) \) requires inference

A \[ \text{B} \] \[ \text{C} \]

Binary variables A, B, C

data point \((a^0, b^2, c^1)\)

\( \rightarrow \) need \( P(B | a^0, c^1, \theta) \)

---

**MN gradient: Complete vs. partial data (KF 19.3.3.1)**

Complete data

\[
\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta : D) = E_D[\phi_i[X]] - E_{\theta}[\phi_i]
\]

1 inference / gradient step

Partial data

\[
\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta : D) = \frac{1}{M} \left[ \sum_{m=1}^{M} E_{h[m] \sim P(h[m]|o[m], \theta)}[\phi_i] \right] - E_{\theta}[\phi_i]
\]

M+1 inferences / gradient step

\( (M = \# \text{ data points}) \)

\( \rightarrow \) much more expensive
Expectation Maximization (KF 19.3.3.2)

\[ E \text{-step:} \quad M_{\theta(t)}^{(c)}[\phi_t] = \frac{1}{M} \left[ \sum_{m=1}^{M} E_{H[m]} \sim P(H[m]|o[m], \theta) [\phi_t] \right] \]

- expected complete assignment
- expectation over missed variables in data point \( m \) -> requires inference for each \( m \)

E-step requires \( M \) (no. data points) inferences

M-step: treat expected variables from E-step as complete data, then max. likelihood estimation
- for Bayes nets, closed form
- for Markov nets, need inference inside gradient ascent
  gradient form \( \frac{\partial}{\partial \theta_t} \frac{1}{M} \ell(\theta : D) = \frac{1}{M} \sum_{m=1}^{M} E_{H[m]} \sim P(H[m]|o[m], \theta) [\phi_t] - E_{\theta} [\phi_t] \)

Comparison (KF 19.3.3.2)

Gradient ascent on log-likelihood

\( M+1 \) inferences / gradient step

\[ \frac{\partial}{\partial \theta_t} \frac{1}{M} \ell(\theta : D) = \frac{1}{M} \sum_{m=1}^{M} E_{H[m]} \sim P(H[m]|o[m], \theta) [\phi_t] - E_{\theta} [\phi_t] \]

Expectation Maximization

E step: computes 1st term (observation counts) and “caches” them

M step: gradient ascent in “complete” setting
  only 1 inference / gradient step
  “cached” counts become less relevant as ascent proceeds
  often better not to run ascent to convergence
Outline

Short review
Markov net parameter learning
**Structure learning**

Structure learning with partial data

Case 1:
- know all the nodes
- want to find the edges
- partially-observed data

Case 2:
- do not know all the nodes
  - i.e. hidden variables
- want to find edges and hidden nodes
- partially-observed data
Greedy MN structure learning (also see KF Fig. 19.3)

Total feature set \( \Omega \)
Initial feature set \( \Phi_0 \)
at all times: \( \theta_i = 0, \forall \phi_i \notin \Phi \)

Iterate {
    Optimize \( \theta_\Phi \) (parameter optimization)
    Iterate over modification operators \( \mathcal{O} \) to structure {
        \( \mathcal{O} \) creates \( \Phi_{\text{mod}} \)
        \( \hat{\Delta}_\mathcal{O} \) = improvement in score
    }
    choose set of modifications \( \mathcal{O} \) based on \( \hat{\Delta}_\mathcal{O} \)
    \(-\) new structure \( \Phi \)
}

Case 1 - no hidden variables

Somewhat like structure learning with complete data
(see previous slide)

But scores much more expensive with partial data
    marginal log likelihood expensive
    inside structure modification loop

Various methods to approximate / speed things up

Log-likelihood & overfitting
    Parameters & structure
    Regularization methods
Structure learning with hidden variables

Hidden variables are MCAR (trivially)
Not identifiable
Local maxima in likelihood
V. hard, generally
If wrong choice of hidden variables
  -> underconstrained (# parameters > # data)
Best when model overconstrained to start
  hidden variables then “useful” in modeling dependencies
  among observed variables
Proper initialization crucial