Graphical Models and Bayesian Networks

Bioinformatics Course Supplement

Outline

- Introduction
- Graphical Models
- Bayesian Networks
- EM Algorithms
- Summary
Learning and Probabilistic Inference

- Many probabilistic inference problems can be solved if we have a joint distribution of the variables involved.
- Joint distribution associated with a probabilistic inference problem can be decomposed into locally interacting factors.
- By taking advantage of probabilistic structure, inference can be performed more efficiently than the blind application of Bayes’ rule.
- Many learning problems can be formulated in the framework of probabilistic inference:
  - Supervised learning (classification)
  - Unsupervised learning
  - Data compression
  - Channel coding

Supervised Learning (Classification)

- A classifier
  - Estimates the probability $P_j | v$
  - Bayes optimal classifier
    - A minimum error rate is achieved by choosing the class $j$ that maximizes $P_j | v$
- Example: digit classification
  - $P(j)$: a priori distribution over the digit classes
  - $P(h | j)$: a distribution over a set of hidden attributes of a digit class
  - $P(v | h)$: a distribution over possible images given a set of features

$$ P(j, h, v) = P(j) P(h | j) P(v | h) $$

$$ P(j | v) = \frac{P(j, v)}{\sum_{j'} P(j', v)} = \frac{\sum_{h} P(j, h, v)}{\sum_{j'} \sum_{h} P(j', h, v)} $$
Unsupervised Learning

- **The Goal**
  - Process a set of training data and then extract underlying structure which is believed to be relevant to perception and conception.

- **A Possible Approach**
  - In addition to the input sensory variables $v$ there are hidden “concept” variables $h$.
  - $v, h$ are linked together by a parameterized probability model $P(v, h | \theta)$.
  - Given the hidden variables, the sensory inputs are independent.

$$P(v, h, \theta) = P(h | \theta) \prod_{i=1}^{N} P(v_i | h, \theta)$$

- Maximum likelihood parameter estimation

$$\theta_{ML} = \arg \max_t \prod_{i=1}^{T} P(v(t) | \theta)$$

Probabilistic Sturcture and Graphical Models

- **Probabilistic structure**
  - Can be characterized by a set of conditional independence.
  - Express the joint distribution as a product of factors.
  - Each factor depends on a subset of the random variables.
  - Takes advantage of a graphical description of the dependencies between random variables.

- **Graphical models**
  - Graph theory provides a succinct way to represent probabilistic structure.
  - A graphical representation for probabilistic structure + functions that can be used to derive the joint distribution.
  - Concisely capture probabilistic structure, and forms a framework for computing useful probabilities.
Graphical models are a marriage between graph theory and probability theory. They clarify the relationship between neural networks and related network-based models such as HMMs, MRFs, and Kalman filters. Indeed, they can be used to give a fully probabilistic interpretation to many neural network architectures. Some advantages of the graphical model point of view include:

- Inference and learning are treated together
- Supervised and unsupervised learning are merged seamlessly
- Missing data handled nicely
- A focus on conditional independence and computational issues
- Interpretability (if desired)
Learning and Inference in Graphical Models

- A key insight from the graphical model point of view: 
  
  *It is not necessary to learn that which can be inferred*

- The weights in a network make local assertions about the relationships between neighboring nodes

- Inference algorithms turn these local assertions into global assertions about the relationships between nodes
  - e.g., correlations between hidden units conditional on an input-output pair
  - e.g., the probability of an input vector given an output vector

- This is achieved by associating a joint probability distribution with the network

Graphical Models and Independence

- The most common simplifying trick
  - Some independence between the variables
  - Some conditional independence of subsets of variables, conditioned on other subsets of variables

- Graph
  - Node: variable
  - Missing edge: independence relationship
  - Independent relationship: the global high-dimensional probability distribution $P$ over all variables can be factored into a product of *simpler local probability distributions* over lower-dimensional spaces associated with smaller clusters of variables
Graphical Model Structure: An Example

\[
P(A, B, C, D, E) = P(A)P(B \mid A)P(C \mid A, B)P(D \mid A, B, C)P(E \mid A, B, C, D)
\]

\[
P(A, B, C, D, E) = P(A)P(B)P(C \mid A)P(D \mid A, B)P(E \mid D)
\]

\[
P(X) = \prod_{i=1}^{n} P(X_i \mid Pa_i)
\]

Classes of Graphical Models

- Boltzmann Machines
- Markov Random Fields
- Bayesian Networks
- Latent Variable Models
- Hidden Markov Models
- Generative Topographic Mapping
- Non-negative Matrix Factorization
Graphical Models: Undirected and Directed

- Undirected models
  - Edge: symmetric interactions
  - Statistical mechanics, image processing
  - Markov random fields, undirected probabilistic independence networks, Boltzmann machines, Markov networks, log-linear models

- Directed models
  - Edge: not symmetric interactions, causal relationship, time irreversibility
  - Expert systems, problems based on temporal data
  - Bayesian networks, belief networks, directed probabilistic independence networks, causal networks, influence diagrams

Applications of Graphical Models

<table>
<thead>
<tr>
<th>Method</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden Markov Models</td>
<td>Speech recognition, bioinformatics</td>
</tr>
<tr>
<td>Mixture Models</td>
<td>Star catalog, market analysis, digit recognition</td>
</tr>
<tr>
<td>Bayesian Networks</td>
<td>Causal models, sensor fusion, expert systems, bioinformatics</td>
</tr>
<tr>
<td>Markov Random Fields</td>
<td>Vision, image processing</td>
</tr>
<tr>
<td>Linear Structural Eq.</td>
<td>Econometric models, social sciences</td>
</tr>
<tr>
<td>Phylogenetic Trees</td>
<td>Evolution, bioinformatics</td>
</tr>
</tbody>
</table>
**Graphical Models for Gene Expression Analysis**

- **Learning**
  - Processed data
  - Learning algorithm
  - Gene C, Gene B, Gene D, Gene A
  - Target
  - The values of Gene C and Gene B are given.
  - Belief propagation
  - Probability for the target is computed.

- **Inference**
  - Gene C
  - Gene B
  - Gene D
  - Gene A
  - Target
  - Mix: mixture
  - Red-dim: reduced dimension
  - Dyn: dynamics
  - Distrib: distributed representation
  - Hier: hierarchical
  - Nonlin: nonlinear
  - Switch: switching

**Variations of (Generative) Graphical Models**
Bayesian Networks

- The method of choice for representing uncertainty in AI
- Utilize explicit representation of structure to:
  - Provide a natural and compact representation of large probability distributions.
  - Allow for efficient method for answering a wide range of queries.
- Have been used in numerous applications:
  - Medical diagnosis (PathFinder, QMR)
  - Hardware diagnosis (Microsoft troubleshooter, NASA/Rockwell Vista project)
  - Information retrieval (Ricoh helpdesk)
Bayesian Networks: Example

- Network structure encodes conditional independencies:
  - X-ray is conditionally independent of Smoker given Lung cancer

| S | C | P(E|S,C) |
|---|---|---------|
| s | c | 0.9     |
| s | c | 0.1     |
| s | c | 0.7     |
| s | c | 0.3     |
| s | c | 0.2     |
| s | c | 0.8     |
| s | c | 0.01    |
| s | c | 0.99    |

node = random variables
edges = direct causal influence

Bayesian Networks: Semantics

- Compact & natural representation:
  - Nodes have \( \leq k \) parents \( \Rightarrow O(2^k n) \) vs. \( O(2^n) \) parents
  - Parameters natural and easy to elicit
Bayesian Networks: 3 Main Tasks for any Bayes Net Toolbox

- **Model specification**
  - Qualitative: graph structure (DAG)
  - Quantitative: parameters of the conditional probability distributions (CPDs), i.e., \( \text{Pr}(\text{node}|\text{parents}) \)

- **Inference**
  - Goal: compute \( P(Q|V) = \sum_h P(h, Q|V) \)
  - Different algorithms make different tradeoffs between simplicity, generality, accuracy, speed, etc.

- **Learning**
  - Qualitative: graph structure (hard)
  - Quantitative: parameters of the CPDs (easy)
  - Learning with partial observability uses inference as a subroutine

Bayesian Networks: Model Specification I
- **Structure**

- The structure is a directed acyclic graph (DAG), represented as a (sparse) adjacency matrix
- The nodes must always be numbered in topological order, i.e., ancestors before descendants
  - Future work:
    - Add a GUI
    - Add a file parser for some of the standard formats
Bayesian Networks: Model Specification I
- Structure

- Tabular (multinomial)
- Gaussian
- Softmax (logistic/sigmoid)
- Noisy-or
- Deterministic
- Fully object-oriented, so easy to new CPDs

Bayesian Networks: Inference I

- Exact inference for static BNs
  - Junction tree
  - Variable elimination
  - Brute force enumeration (for discrete nets)
  - Linear algebra (for Gaussian nets)
  - Pear’s algorithm (for polytrees)
  - Quickscore (for QMR)

- Approximate inference for static BNs
  - Likelihood weighting
  - Loopy belief propagation

- Each inference engine is an object, so easy to add more.
Bayesian Networks: Inference II

- Each inference algorithm is implemented in BNT as an “inference engine” object in the following respects [Gatsby Neuroscience Unit, UCL].
- Designed for static or dynamic models?
- Exact or approximate inference?
- Works for all topologies or makes restrictions?
- Works for all node types or makes restrictions?
- Handles any pattern of evidence, or must be fixed?
- What computation is done when the following are specified
  - Structure
  - Parameters
  - Evidence
  - Query

Bayesian Networks: Inference III

<table>
<thead>
<tr>
<th>Name</th>
<th>Exact?</th>
<th>Node type?</th>
<th>Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>cond_gauss</td>
<td>exact</td>
<td>CG</td>
<td>all</td>
</tr>
<tr>
<td>enumerative</td>
<td>exact</td>
<td>allD</td>
<td>all</td>
</tr>
<tr>
<td>gaussian</td>
<td>exact</td>
<td>allG</td>
<td>all</td>
</tr>
<tr>
<td>jtree</td>
<td>exact</td>
<td>D,G,CG</td>
<td>all</td>
</tr>
<tr>
<td>var_elim</td>
<td>exact</td>
<td>D,G,CG</td>
<td>all</td>
</tr>
<tr>
<td>pearl</td>
<td>exact</td>
<td>D,G</td>
<td>polytree</td>
</tr>
<tr>
<td>quickscore</td>
<td>exact</td>
<td>D,G</td>
<td>QMR</td>
</tr>
<tr>
<td>lik_weight</td>
<td>approx</td>
<td>any</td>
<td>all</td>
</tr>
<tr>
<td>loopy_pearl</td>
<td>approx</td>
<td>D,G</td>
<td>all</td>
</tr>
</tbody>
</table>
### Bayesian Networks: Learning

- **Algorithms**
  - Batch EM parameter learning
  - Sequential Bayesian parameter learning
  - Structure learning (for observed tabular nodes only)
- **Regularization**
  - Any node can have its parameters clamped (made non-adjustable)
  - Any set of compatible nodes can have their parameters tied (cf. weight sharing in a neural net)
  - Some node types (e.g., tabular) support priors for MAP estimation
  - Gaussian covariance matrices can be declared full or diagonal, and can be tied across states of their discrete parents (if any)
- **Modularity**
  - Each node type has its own M method, e.g. softmax nodes use IRLS
  - Each inference engine implements its own E method

### Expectation Maximization (EM)

- A general method for optimizing parameters
- Use an initial guess for parameters to find better ones
- **Rough idea**: use current parameters to “complete” counts

\[
D = \begin{array}{ccc}
X & Y & Z \\
H & ? & T \\
T & ? & ? \\
H & H & ? \\
T & T & T \\
T & H & \\
\end{array}
\]

\[
P(Y=H|H,?,T)=0.3 \\
P(Y=H|T,?,?)=0.4 \\
\]

\[
\begin{array}{ccc}
N(X,Y) \\
X & Y & # \\
H & H & 1.3 \\
T & H & 0.4 \\
H & T & 1.7 \\
T & T & 1.6 \\
\end{array}
\]

\[
\begin{array}{ccc}
N(X,Y) \\
X & Y & # \\
H & H & 1.3 \\
T & H & 0.4 \\
H & T & 1.7 \\
T & T & 1.6 \\
\end{array}
\]
The EM Algorithm for Learning with Hidden Variables:

- Assume a model parameterised by $\theta$ with observable variables $Y$ and hidden variables $X$
- Goal: maximize log likelihood of observables

\[
L(\theta) = \ln P(Y \mid \theta) = \ln \sum_X P(Y, X \mid \theta)
\]

- E-step: first infer $P(X \mid Y, \theta_{old})$, then
- M-step: find $\theta_{new}$ using complete data learning

- The E-step requires solving the inference problem: finding explanations, $X$, for the data, $Y$ given the current model $\theta$

(C) 2001 SNU CSE Artificial Intelligence Lab (SCAI)
Model Search: Structural Learning

- Finding the BN structure with the highest score among those structures with at most $k$ parents is NP hard for $k > 1$ (Chickering, 1995)
- Heuristic methods
  - Greedy
  - Greedy with restarts
  - MCMC methods

Bayesian Networks: Current Issues

- Fancy distributions (decision trees, logit, probit, noisy or continuous variables)
- Model selection
- Hidden variables and missing data
- Time-varying domains (cf. dynamic models)
- Online learning
- Causality and causal interpretations for Bayes nets
Summary

- Graphical models provide a principled and rigorous approach to inference and learning.
- They offer a general framework for treating supervised and unsupervised learning together.
- Graphical models are interpretable (cf. multilayer perceptrons).
- Directed graphical models allow for causal relationships to be represented and discovered.
- Graphical models can be used to clarify the relationship between different machine learning methods.
- Drawbacks: computationally intensive
- But, work is in progress to develop efficient learning and inference algorithms for graphical models.