PGMs for Deep Learning: Inference

Sargur N. Srihari
srihari@cedar.buffalo.edu

Sampling from Graphical Models

- Graphical models facilitate drawing samples from a model
- One advantage of using a directed graphical model is that a procedure called *ancestral sampling* can produce samples from the joint distribution represented by the model

Ancestral Sampling

- Start with lowest numbered node
- Draw a sample from the distribution $p(x_1)$ which we call $\hat{x}_1$
- Work through each of the nodes in order
  - For node n we draw a sample from conditional distribution $p(x_n|pa_n)$
  - Where parent variables are set to their sampled values
- Once final variable x_K is sampled
  - Achieved objective of obtaining a single sample from joint distribution
- To sample from marginal distribution
  - Sample from full distribution and discard unnecessary values
  - E.g., to draw from distribution $\mathrm{p}(\mathrm{x}_2,\mathrm{x}_4)$ simply sample from full distribution, retain values $\mathrm{x}_2,\mathrm{x}_4$^ and discard remaining values $\{\hat{x}_{j\neq 2,4}\}$

Sampling from Undirected graphs

- Ancestral sampling is applicable only to directed models
- We can sample from undirected models by converting them to directed models
  - But involves solving intractable inference problems
    - To determine marginal for root nodes of directed graph
  - Or introducing so many edges that the resulting directed model becomes intractable
- So drawing samples from an undirected graphical model is an expensive multi-pass process

Gibbs Sampling

- The conceptually simplest approach for drawing samples from an undirected graph
- Suppose we have a graphical model over an n-dimensional vector of random variables $\mathbf{x}$
- We iteratively visit each variable x_i and draw a sample conditioned on all the other variables, i.e., from $p(\mathrm{x}_i|\mathrm{x}_{-i})$
- Due to the separation properties of the graphical model, we can equivalently condition on only the neighbors of x_i

Gibbs Sampling with M variables

- Initialize first sample: $\{z_i,\ i=1,..,M\}$
- For $t=1,..,T$, T = no of samples
  - Sample $z_1^{(t+1)} \sim p(z_1|z_2^{(t)},z_3^{(t)},\ldots,z_M^{(t)})$
  - Sample $z_2^{(t+1)} \sim p(z_2|z_1^{(t+1)},z_3^{(t)},\ldots,z_M^{(t)})$
  - .....
  - Sample $z_j^{(t+1)} \sim p(z_j|z_1^{(t+1)},..z_{j-1}^{(t+1)},\ z_{j+1}^{(t)}\ \ldots,z_M^{(t)})$
  - .....
  - Sample $z_M^{(t+1)} \sim p(z_M|z_1^{(t+1)},z_2^{(t+1)},\ldots,z_{M-1}^{(t+1)})$
- $p(z_j|z_{-j})$ is called a *full conditional* for variable j

Gibbs Sampling Termination

- Unfortunately, after one pass through the graphical model and sampled all n variables, we still do not have a fair sample from $p(\mathbf{x})$
- Instead we must repeat the process and resample all n variables using the updated values of the neighbors
- Asymptotically after many repetitions, process converges to sampling from correct distribution
- Difficult to determine when the samples have reached a sufficiently accurate approximation

Advantages of Structured Modeling

- Primary advantage of using PGMs:
  - Allow us to dramatically reduce cost of representing probability distributions as well as learning and inference
- Sampling is accelerated for directed models
  - Situation is more complicated for undirected models
- Allow us to explicitly separate:
  - representation of knowledge
  - learning of knowledge or
  - inference given existing knowledge

Learning about Dependencies

- A generative model has to capture distribution over observed or "visible" variables $\mathbf{v}$
- Often elements of $\mathbf{v}$ are depend on each other
  - In deep learning, approach used to capture these dependencies is to introduce several latent or "hidden" variables $\mathbf{h}$
  - Model can then capture dependencies between any pair of variables v_i and v_j indirectly
    - Via direct dependencies between v_i and h and direct dependencies between h and v_j

Computational savings by using h

- A good model of v which did not contain any latent variables h will need to have
  - A very large number of parents per node in a Bayesian network or a
  - A very large no. of cliques in a Markov network
- Just representing these interactions is costly
  - Exponential no of parameters
  - Wealth of data needed to estimate the parameters

PGM structure learning improvement

- When searching for PGM structure, it is infeasible to connect all visible variables
  - Structure learning algorithms perform greedy search
    - Structure is proposed, model is trained, then scored
    - Score rewards training accuracy & penalizes complexity
    - Candidate structures with a small no of edges added/ removed are proposed at next step
    - Search proceeds to new structure expected to increase score
  - Using latent variables, instead of adaptive structure:
    - Avoids need to perform discrete searches and multiple rounds of training

Advantage of PGM with fixed structure

- A fixed structure with both visible and hidden variables can use
- *Direct* interactions between visible-hidden units to impose *indirect* interactions between visible units
- Simple parameter learning techniques can be used to learn a model with a fixed structure that imputes the right structure on the marginal $p(\mathbf{v})$

Variables $\mathbf{h}$ provide alternative to $\mathbf{v}$

- New variables $\mathbf{v}$ provide an alternative representation for $\mathbf{v}$
- Mixture of Gaussians model learns a latent variable that corresponds to which category of examples the input is drawn from
  - This means that the latent variable can be used to perform classification

Inference and Approximate Inference

- Ask questions about how variables relate
  - Given medical tests, what disease a patient has
  - In a latent variable model extract features $\mathbb{E}[\mathbf{h}|\mathbf{v}]$ describing observed variables $\mathbf{v}$
  - Solve such problems in order to perform other tasks
    - We want to compute $p(\mathbf{h}|\mathbf{v})$ to determine $p(\mathbf{v})$
- These are inference problems
  - Predict variables given other variables
  - Predict distributions of some variables given values of other variables

Intractability of Inference

- Even when we use PGMs inference problems are intractable
- Graph structures allow complicated high-dimensional distributions with reasonable no of parameters
- But resulting graphs are not restrictive enough to allow efficient inference

Complexity Class of PGM Inference

- Computing marginal probability is $\#\mathrm{P}$ hard
- The complexity class $\#\mathrm{P}$ is a generalization of class NP
- Problems in NP requires only whether a problem has a solution, and if so find it
- Whereas problems in $\#\mathrm{P}$ requires counting all possible solutions
- This motivates the use of approximate inference

Approximate Inference

- In the context of deep learning approximate inference refers to variational inference
- We approximate the distribution $p(\mathbf{h}|\mathbf{v})$ by another distribution $q(\mathbf{h}|\mathbf{v})$ that is as close to the true one as possible

Deep Learning approach to PGMs

- Deep learning does not involve deep graphical models
  - For PGMs in deep learning, depth of a model is in terms of PGM graph rather than computational graph
    - Latent variable h_i is at depth j if the shortest path from h_i to an observed variable is j steps
    - Depth of a model is the greatest depth of any h_i

Use of Latent Variables in PGMs

- Traditional graphical models
  1. Few latent variables
     - Most variables are observed
  2. Designed for semantics
     - e.g., intelligence, topic of documents
  3. Structure learning used to get complicated models
- Deep learning models
  1. More latent variables than observed variables
  2. Latent variables have no pre-specified semantics
  3. Use single large layer of latent variables
     - Nonlinear interactions between variables accomplished via indirect interactions through latent variables

Connectivity in Traditional PGMs

- Very few connections
- Choice of connections for each variable may be individually designed
- Design of model structure may be tightly linked to inference algorithm
  – Aim to keep exact inference tractable
  – If this constraint is too limiting, approximate inference called loopy belief propagation is used

Connectivity in DGMs

- Deep Graphical Models typically have a large no of units connected to other groups of units
- So that interactions between the two groups may be described by a single matrix
- Graphs are not sparse enough for traditional exact inference and loopy belief propagation

Inference in DGMs

- Striking difference between PGM and DGM communities is that loopy belief propagation is never used in DGMs
- Most DGMs are designed to make Gibbs sampling or variational inference more exact
- Due to very large no of latent variables, efficient numerical code is essential
  - Matrix operations like block-diagonal matrix products or convolutions