Advanced Deep Learning

Structured Probabilistic Models for Deep Learning-2

U Kang
Seoul National University

Outline

- ☑ Challenge of Unstructured Modeling
- ☑ Using Graphs to Describe Model Structure
- ➡ □ **Sampling from Graphical Models**
- □ Advantages of Structured Modeling
- □ Learning about Dependencies
- □ Inference and Approximate Inference
- □ Deep Learning Approach to Structured PM

A definition of sampling

- Graphical models facilitate the task of drawing samples from a model.

- A sample over a set of variables = an instantiation of all variables
  - Example: given a set of variables X={X_1, X_2, … , X_n}
    - → S = {x_1, x_2, … , x_n} is a sample
    - → Where for all i in [1:n], x_i is a possible value of X_i

Ancestral sampling (1)

- Ancestral sampling - Basic idea:
  - Sort all the X_i in the graph into a topological ordering, so that for all j > i, X_i is a parent of X_j
  - Sample the variables in this order. That is to say:
    - first sample $P(X_1)$,
    - then $P(X_2|parents(X_2))$,
    - until the end where we sample $P(X_n|parents(X_n))$

- We then obtain the joint distribution: $p(X) = \Pi_i p(X_i|parents(X_i))$ by multiplying all of our precendent results

Ancestral sampling (2)

- Ancestral sampling - Advantages:
  - Simple and efficient procedure over Bayesian network to produce a sample from the joint distribution represented by the model.
  - The ancestral sampling works for all existing topological orderings.
  - As long as we sorted the variables right, we can calculate $p(x_i|parents(x_i))$ for all i.
  - As long as each conditional probability is easy to calculate, the joint probability will be as easy to calculate.

Ancestral sampling (3)

- Ancestral sampling - Drawbacks:
  - Doesn’t work for every conditional sampling operation
    - To sample a variable given some others, we need to have already sampled the conditioning ones.
  - It only applies to Bayesian network

Gibbs sampling

- To draw sample from an undirected graphical model, the Gibbs sampling method is the conceptually most simplest one:
  - Let X = {X_1, … , X_n} be a vector of n random variables.
  - For all i, sample p(X_i|X\{X_i})
  - Repeat the process with the updated values until the process converges to sampling from p(X).

Outline

- ☑ Challenge of Unstructured Modeling
- ☑ Using Graphs to Describe Model Structure
- ☑ Sampling from Graphical Models
- ➡ □ **Advantages of Structured Modeling**
- □ Learning about Dependencies
- □ Inference and Approximate Inference
- □ Deep Learning Approach to Structured PM

Primary Advantages of Structured Modeling

- Allow us to considerably reduce the cost of:
  - Representing probability distributions
  - Learning
  - Inference
- Accelerate the process of sampling in the case of Bayesian network
- This is because with graphical models, we do not model some interactions but still convey the information → use less runtime and memory

Outline

- ☑ Challenge of Unstructured Modeling
- ☑ Using Graphs to Describe Model Structure
- ☑ Sampling from Graphical Models
- ☑ Advantages of Structured Modeling
- ➡ □ **Learning about Dependencies**
- □ Inference and Approximate Inference
- □ Deep Learning Approach to Structured PM

Learning About Dependencies (1)

- A good generative model of X = $\{X_1, X_2, \ldots , X_n\}$ needs to accurately capture the distribution over the observed variables of X.
- But the elements of X are often highly dependent of each other
  - In the context of deep learning, we introduce hidden (=latent) variables H = $\{H_1, H_2, \ldots , H_m\}$
  - This way, dependencies between any X_i and X_j are indirectly captured via direct dependencies between X_i and H, and H and X_j.

Learning About Dependencies (2)

- A good model of X that doesn’t have hidden variables might have:
  - Either a very large numbers of parents per node (in the case of Bayesian networks)
  - Or very large cliques (in the case of Markov networks)
- An entire field of machine learning called “structure learning” is dedicated to the problem of designing graphs which
  - Connect the tightly coupled visible variables
  - Omit edges between other variables

Structure Learning

- Most of the times, structure learning techniques are greedy searches consisting of
  - Proposing a structure
  - Training a model with this structure
  - Giving a score to the model
    - →Rewarding accuracy
    - →Penalizing model complexity
  - Proposing a small structure with small number of edges added or removed, which is supposed to increase the precedent score
  - ...

Outline

- ☑ Challenge of Unstructured Modeling
- ☑ Using Graphs to Describe Model Structure
- ☑ Sampling from Graphical Models
- ☑ Advantages of Structured Modeling
- ☑ Learning about Dependencies
- ➡ □ **Inference and Approximate Inference**
- □ Deep Learning Approach to Structured PM

Inference and Approximate Inference (1)

- Main utility of probabilistic models: figure out and ask how variables are related to each other.
  - E.g., given a set of medical tests, ask what disease a patient might have
- In a latent variable model, we might want to extract features E[h|v] describing the observed variables v
- Inference problem: predict the value (or distribution) of some variables given other variables

Inference and Approximate Inference (2)

- But computing the marginal probability of a general graphical model is #P hard (harder than NP).
- In practical real world scenarios, NP hard graphs commonly arise.

→This motivates the use of approximate inference (approximating the results with finite samples).

→In deep learning, variational inference is preferably used: we seek an approximate distribution q(H|X) as close as possible to the true one p(H|X)

Outline

- ☑ Challenge of Unstructured Modeling
- ☑ Using Graphs to Describe Model Structure
- ☑ Sampling from Graphical Models
- ☑ Advantages of Structured Modeling
- ☑ Learning about Dependencies
- ☑ Inference and Approximate Inference
- ➡ □ **Deep Learning Approach to Structured PM**

The Deep Learning Approach to Structured Probabilistic Models

- We define the depth of a graphical model as follows:
  - A latent variable h_i is at depth n_i if the shortest path from h_i to an observed variable is n_i steps.
  - The depth of the graphical model is $n = \max(n_i)$

- Many generative models used for deep learning have 0 or 1 layer of hidden variables

Deep Learning Models vs. Traditional Graphical Models (1)

- Difference in the number of latent variables:
  - Deep learning
    - Typically have more latent variables than observed variables
    - Complicated nonlinear interactions between variables accomplished via indirect connections through multiple latent variables
  - Traditional graphical models
    - Contain mostly variables that are at least occasionally observed
    - Mostly use higher order terms and structure learning to capture the complicated nonlinear interactions between variables
    - If there are latent variables, they are usually few in number

Deep Learning Models vs. Traditional Graphical Models (2)

- The way latent variables are designed also differs in deep learning:
  - Deep learning: practitioners do not intend for the latent variables to take on any semantics ahead of time: any concept can be invented by the training algorithm if needed

    →not easy for the human to interpret (even if visualization techniques might give a rough idea)

    →but reusable in many different contexts
  - Traditional graphical models: latent variables are often designed with a specific semantics in mind

    →more interpretable by human

    →but less able to scale to complex problems

Deep Learning Models vs. Traditional Graphical Models (3)

- The kind of connectivity:
  - Deep learning: there are large groups of units all connected to other units → the interactions between two groups can be described by a single matrix
  - Traditional graphical model: very few connections → for each variable, the choice of connection can be individually designed

Deep Learning Models vs. Traditional Graphical Models (4)

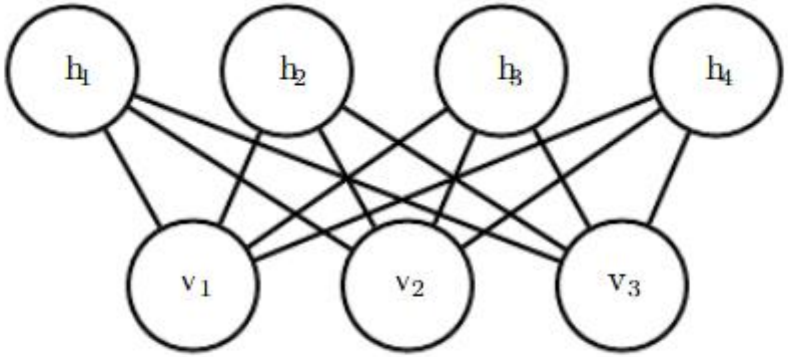
- The design of the model structure is tightly linked with the choice of inference algorithm.
  - Deep learning: we tend to connect each visible unit X_i to as many hidden unit H_j as possible → this way, H can provide a distributed representation of X_i (and probably others X_k)
  - Traditional graphical model: we try to maintain the tractability of exact inference

    →when this constraint is too limiting, a popular approximate inference algorithm is the loopy belief propagation

EXAMPLE: THE RESTRICTED BOLTZMANN MACHINE

Restricted Boltzmann Machine

- RBM = restricted Boltzmann machine = harmonium
  - A single layer of latent variables: not a deep model itself
  - This layer can be used to learn a representation for the input



An RBM drawn as a Markov network.

Restricted Boltzmann Machine

- The RBM is the best example of how graphical models are used for deep learning:
  - Its units are organized into large groups
  - The connectivity between layers is described by a matrix
  - The connectivity is relatively dense
  - The model is designed to allow efficient Gibbs sampling

Restricted Boltzmann Machine

- Important aspects of the RBM model:
  - It is energy based, with the following energy function:

$$E(\boldsymbol{v},\boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W}\boldsymbol{h}$$

Where **b**, **c**, and **W** are unconstrained, real valued, and learnable parameters.

v and **h** are the previously called X and H.

  - There are no direct interaction between any two visible units or any two hidden units → independencies giving us the following properties:

$$p(\mathbf{h} \mid \mathbf{v}) = \Pi_i p(\mathrm{h}_i \mid \mathbf{v})$$

$$p(\mathbf{v} \mid \mathbf{h}) = \Pi_i p(\mathrm{v}_i \mid \mathbf{h})$$

Restricted Boltzmann Machine

- We can also compute the individual conditionals (as follows in the case of a binary RBM):

$$P(h_i = 1|v) = \sigma\left(v^T W_{:,i} + c_i\right)$$
$$P(h_i = 0|v) = 1 - \sigma(v^T W_{:,i} + c_i)$$

where σ is the sigmoid function: $\sigma(x) = \frac{1}{1+\exp(-x)}$,

- We can then use Gibbs sampling efficiently, as it alternates between sampling all of H, then all of V.
- We can also easily take gradient of the energy function:

$$\frac{\partial}{\partial W_{i,j}} E(\mathbf{v}, \mathbf{h}) = -\mathrm{v}_i \mathrm{h}_j$$

Restricted Boltzmann Machine

- So we have two properties making the training convenient:
  - Efficient Gibbs sampling
  - Efficient gradient

- Overall, we have seen that the RBM demonstrates the typical deep learning approach to graphical models:
  - Representation learning via layers of latent variables
  - Efficient interactions between layers parametrized by matrices.

What you need to know

- Challenge of Unstructured Modeling
- Using Graphs to Describe Model Structure
- Sampling from Graphical Models
- Advantages of Structured Modeling
- Learning about Dependencies
- Inference and Approximate Inference
- Deep Learning Approach to Structured PM

Questions?