CS281: Advanced ML **October 2, 2017**

Lecture 9: Undirected Graphical Models

Lecturer: Sasha Rush *Scribes: Chris Hase, Denis Ellenrieder, Shuran Zheng, Rafi Small, Tim Menke*

9.1 Left from last lecture: Conditional independence properties of DGMs

When we have a directed graphical model (DGM), how can we "read" the graph and learn about the independence properties of the variables? To begin we note that conditional independence follows if the marginal probability factors in the following way:

$$p(A, B|C) = p(A|C)\, p(B|C) \Rightarrow A \perp B|C \tag{9.1}$$

Let (B) denote observing B, which informs our understanding on how A and C are conditional independent or not. Our cases are then:

$$(A) \rightarrow (B) \rightarrow (C),\ A \perp C \times \tag{9.2}$$
$$(A) \rightarrow \textcircled{B} \rightarrow (C),\ A \perp C|B \checkmark \tag{9.3}$$
$$(A) \leftarrow (B) \rightarrow (C),\ A \perp C \times \tag{9.4}$$
$$(A) \leftarrow \textcircled{B} \rightarrow (C),\ A \perp C|B \checkmark \tag{9.5}$$
$$(A) \rightarrow (B) \leftarrow (C),\ A \perp C \checkmark \tag{9.6}$$
$$(A) \rightarrow \textcircled{B} \leftarrow (C),\ A \perp C|B \times \text{ explaining away} \tag{9.7}$$

Information is being blocked in cases 2, 4, and 5 but flowing freely in all other cases. It's useful to think about the concept of "explaining away" to understand what is going on in the last case. '"Explaining away" is a common pattern of reasoning in which the confirmation of one cause of an observed or believed event reduces the need to invoke alternative causes.'
(http://strategicreasoning.org/wp-content/uploads/2010/03/pami93.pdf)

How would a directed graphical model be interesting in practice? One example is probabilistic programming: demonstrated in the IPython notebook "DGM.ipynb". The demonstration shows how we can convert programs from a directed to an undirected form. We can also specify priors on our features and visualize the flow of the model.

The purpose of creating a DGM is to specify the relationship between variables of interest, in order to to facilitate understanding of the independence properties.

9.2 Undirected Graphical Models (UGM)

Differences of UGMs as opposed to DGMs:

1. There are no arrows on lines
2. No longer model *local* probabilistic decisions (the term "local" is important and defined later)
3. UGMs are Markov random fields (similar to exponential families) or conditional random fields (similar to generalized linear models)
4. Nice part: The rules are much simpler, especially for conditional independence

5. Downside: The math is much more complicated
6. Sasha's personal bias: UGMs are much more useful

9.2.1 Independence properties

As stated above, we have conditional independence if the marginal probability factors in the following way:

$$p(A, B|C) = p(A|C)\, p(B|C) \Rightarrow A \perp B|C \tag{9.8}$$

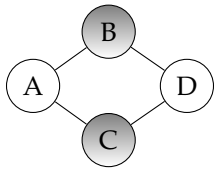
$$\text{A} - \text{B} - \text{C},\ A \perp C \times \tag{9.9}$$

$$\text{A} - \text{B} - \text{C},\ A \perp C|B \checkmark \tag{9.10}$$

$$\tag{9.11}$$

We can say that we have "conditional independence" between two nodes given some third node if all paths between the two nodes are blocked. For our simple example with three nodes, this is when the third node is in the evidence, that is, when the third node is "seen".

Here is an example of how independence works in the UGM.

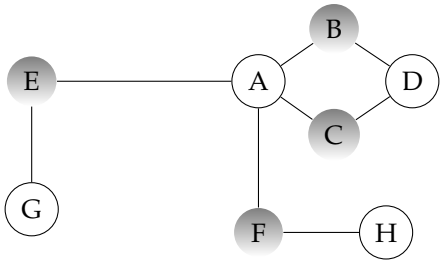


$$A \perp D|S \text{ if S blocks all the paths} \tag{9.12}$$

$$\text{Here: } A \perp D|B, C \checkmark \tag{9.13}$$

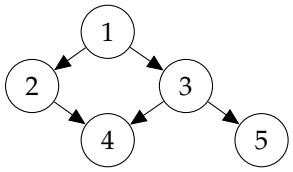
Fundamental consequence: Imagine we have a graph with a node A and some complicated connection of nodes around A. We can make A conditionally independent from all other nodes by conditioning on the "Markov blanket" of A. The Markov blanket is defined to be the *neighbors of A*. We will see later in this class that is is very nice if we can establish these independence properties. We will be able to look at a point in a graphical model and, if we can condition on the Markov blanket of the node of interest, we can ignore the rest of the graph.

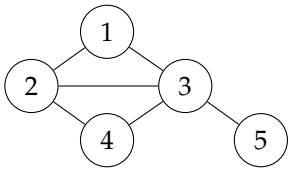
In the example below, conditioning on the Markov blanket of A means conditioning on B, C, E, and F.



9.2.2 Converting directed to undirected graphs

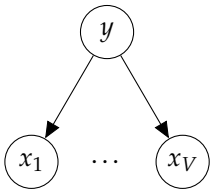
In order to convert directed to undirected graphs, we will use the (socially improperly termed) technique of *moralization*, i.e. "marry the parents".

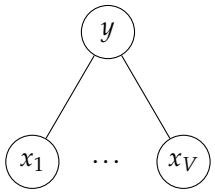




To carry out this process, we take all the directed edges and make them undirected. We then create additional edges by "marrying" the parents of a node. In this case we gain an extra edge between 2 and 3, which comes from marrying the parents of 4.

Let's write Naive Bayes out in a directed graphical model:



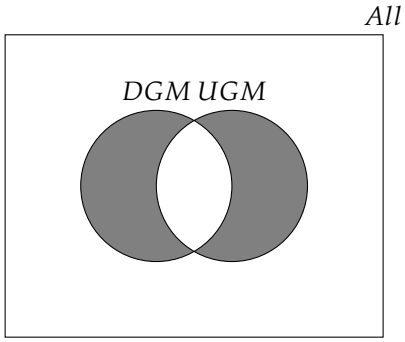


For this case, the UGM and DGM are the same. However, if we condition the other way, i.e. the features $x_1, ..., x_D$ are on top of the DGM with directed arrows towards y at the bottom, we would need to add connections in the UGM between all of the features. In the illustrated example, the joint probability having seen y is

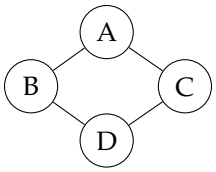
$$\prod_d p(x_d)p(y|x_1, ..., x_v). \tag{9.14}$$

9.3 Corner cases

Unfortunately, there are lots of corner cases in UGMs. DGMs and UGMs represent only a subset of all graphical models. There is some overlap between the UGM and DGM classes within the set of all independence structures. In the corner cases, we cannot convert between DGMs and UGMs and retain all of the independence information encoded within.



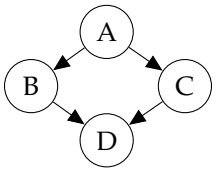
We will now consider an example of a UGM and attempt to convert it to a DGM.



Here we have

$$A \perp D|B,C \checkmark \tag{9.15}$$
$$B \perp C|A,D \checkmark \tag{9.16}$$
$$\tag{9.17}$$

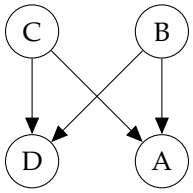
One potential DGM we can create is



In contrast to the UGM, the DGM has the following independence properties.

$$A \perp D|B,C \checkmark \tag{9.18}$$
$$B \perp C|A,D \times \tag{9.19}$$

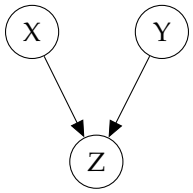
Another DGM may be



And now we see that

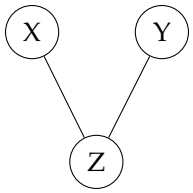
$$A \perp D|B,C \checkmark \tag{9.20}$$
$$B \perp C|A,D \times \tag{9.21}$$

So there is no directed graphical model structure that gives us same properties that the original UGM had. How about another example of a DGM

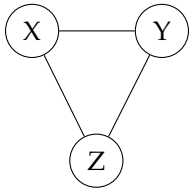


This graph has two properties:

$$(1)\ X \not\perp Y|Z \tag{9.22}$$
$$(2)\ X \perp Y \tag{9.23}$$



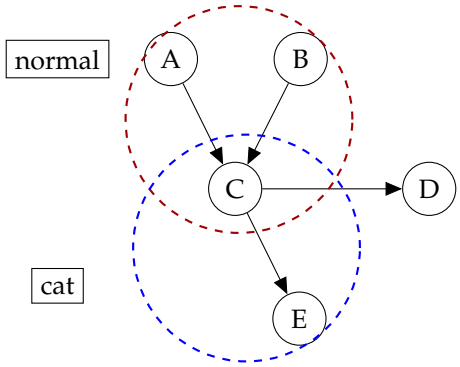
This graph has neither property 1 nor property 2.



Here we do have property 1 but not property 2.

9.4 Parametrization of UGMs

The following section is some of the harder material that we will cover. However, this is the last time we are introducing a new model. Thereafter, we will do inference on the models we have discussed.



Suppose that in this example the nodes within the red circle follow a local normal distribution, but that the nodes within the blue circle follow a local categorical distribution. The notation relating a node X to its parents is $p(x|\text{pa}(x))$, where $\text{pa}(c)$ refers to the parents, and the conditional probability table is "locally normalized", "sums to one", and is non-negative.

For UGMs we use "global normalization". All is fine locally as long as the whole global probability sums up to make whole joint distribution normalized. For this, we treat everything as an *exponential family*.

$$p(x_1, ..., x_d) = \text{multivariate exp. fam.} = \exp\left\{\theta^T \phi(x_1, ..., x_D) - A(\theta)\right\} \tag{9.24}$$

Here, θ are the parameters and $\phi(x_1, ..., x_D)$ the sufficient statistics. For every "clique" in the graph, we associate a set of sufficient statistics. A "clique" is defined as a set of nodes that are all connected to each other. Note: a set of one or two nodes forms a trivial clique.

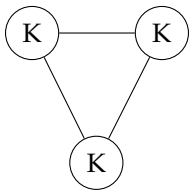
In the discrete case:

$$\phi(x_1, ..., x_D) = [\phi_c(x_c)...]^T \tag{9.25}$$

Each entry is an indicator of the value set for a clique.

$$\phi_c^v(x) = \begin{cases} 1 & \text{if clique } c \text{ has } x_c = v \\ 0 & \text{otherwise} \end{cases} \tag{9.26}$$

Suppose we have the following UGM in which each node can take on one of k possible discrete outcomes:



This means $v \in K^3$ in this example. We also have a Θ associated with each of these values. Since $\phi(x)$ is big (and awkward), we use the following notation:

$$\theta^T \phi(x) = \sum_c \theta^T [0, ..., 0, \phi_c(x), 0, ...,]^T = \underbrace{\sum_c \theta_c(x_c)}_{\text{convenient notation}} \tag{9.27}$$

Now we write out the whole thing:

$$p(x_1, ..., x_D) = \exp \left\{ \underbrace{\sum_c \theta_c(x_c)}_{\text{(neg) energy}} - A(\theta) \right\} \tag{9.28}$$

Finally, we can compute the value $A(\theta)$:

$$A(\theta) = \log \underbrace{\sum_{x'}}_{\text{“everything”}} \exp\{\sum_c \theta_c(x'_c)\} \tag{9.29}$$

The sum over “everything” is our nemesis because it is over something big. And in general, this is NP-Hard or even #P-Hard. But in practice, the structure of the graph determines the difficulty. Examples of the “everything” include: all possible images, social network graphs... (How we can exchange the sums over x' and c depends on the structure of the graph.)

9.5 Examples

9.5.1 Example 1: Naive Bayes model

$$\text{(A)} - \text{(B)} - \text{(C)}, \text{ all binary} \tag{9.30}$$

We have two cliques: (A) − (B) and (B) − (C)
Define features and parameters:

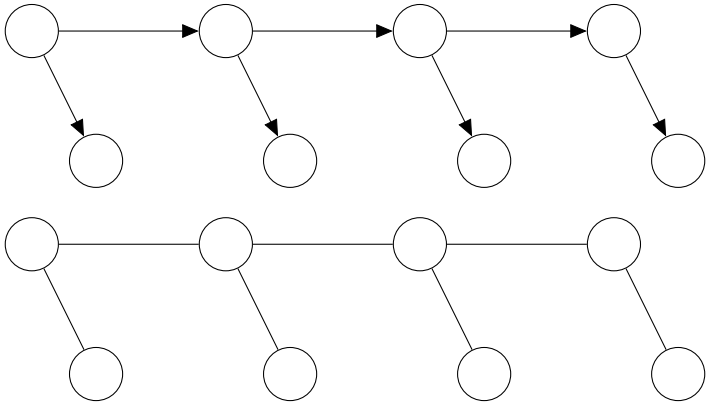
$$\phi(x) = \begin{bmatrix} \mathbb{1}(A=0, B=0) \\ \mathbb{1}(A=0, B=1) \\ \mathbb{1}(A=1, B=0) \\ \mathbb{1}(A=1, B=1) \\ \mathbb{1}(B=0, C=0) \\ \mathbb{1}(B=1, C=0) \\ \mathbb{1}(B=0, C=1) \\ \mathbb{1}(B=1, C=1) \end{bmatrix}, \ \theta = \begin{bmatrix} \theta_{AB}(0,0) \\ \theta_{AB}(1,0) \\ \theta_{AB}(0,1) \\ \theta_{AB}(1,1) \\ \theta_{BC}(0,0) \\ \theta_{BC}(1,0) \\ \theta_{BC}(0,1) \\ \theta_{BC}(1,1) \end{bmatrix} \tag{9.31}$$

Then we have

$$p(A = a, B = b, C = c) = \exp\{\theta_{AB}(a, b) + \theta_{BC}(b, c) - A(\theta)\}, \tag{9.32}$$

where $A(\theta) = \log\left(\sum_{a',b',c'} \exp\{\theta_{AB}(a', b') + \theta_{BC}(b', c')\}\right)$.

9.5.2 Example 2: Hidden Markov model



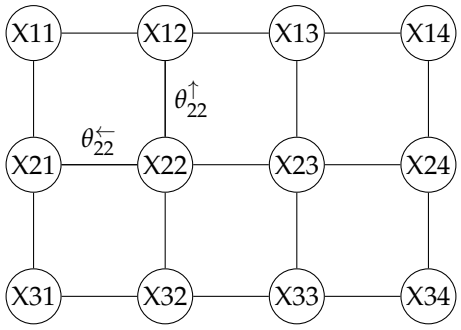
Any distribution obeying this structure has an exponential family parametrization. We convert the local distribution

$$p(y_1) \prod_i p(y_i|y_{i-1}) p(x_i|y_i) \tag{9.33}$$

to the globally normalized distribution

$$p(x, y) = \exp\{\sum_i [\theta_{i,i-1}(y_i, y_{i-1}) + \theta_i(y_i, x_i)] - A(\theta)\}. \tag{9.34}$$

9.5.3 Example 3: Ising model



For this example, the shown grid could connect to other such grids. As an example, suppose we want to detect the foreground and background of an image. For this, we perform a binary classification of each pixel (0=pixel in background; 1=pixel in foreground). We want to obtain a probability that a pixel is in either class. The class should depend on the neighboring pixel so that there is consistency among neighboring pixels - it would be weird if every other pixel is in a different class.

This results in a binary model with neighbor scores. In order to force this to be a probability distribution, we treat the whole thing as an exponential family:

$$p(x) = \exp\left\{\sum_{ij}\left(\theta_{ij}^{\uparrow}(x_{ij}, x_{i-1,j}) + \theta_{ij}^{\leftarrow}(x_{ij}, x_{i,j-1}) + \theta_{ij}(x_{ij})\right) - A(\theta)\right\}, \tag{9.35}$$

where $A(\theta) = \log\left(\sum_{x'} \exp \sum_c \theta_c(x'_c)\right)$. Note that A is again very hard to calculate. Also note that the "missing" $\theta_{22}^{\downarrow}$ and $\theta_{22}^{\rightarrow}$ in the diagram are given by other $\theta^{\uparrow}$ and $\theta^{\leftarrow}$ so that when we sum over all i and j we are not double counting any of the connections.