208891 Probabilistic Graphical Models **Spring 2019**

Lecture 1: December 19

Lecturer: Donlapark Pornnopparath

The notes for this course will closely follow Koller and Friedman (2009) and Kuleshov and Ermon (2019).

1.1 Motivation

In this course, we try to solve real-world problems using *probabilistic modeling.* As a motivating example, suppose we want to predict the price of a house from various factors such as the location, the number of bedrooms, the total floor area, etc. Probably the most basic model for this type of tasks is the linear regression,

$$y = \sum_{i=0}^{m} \beta_i x_i,$$

where y is the house price and x_i's are the other factors. However, real life decisions often involves *uncertainty* we have to take into account. For example, if there is a new mall opening nearby, then the house prices around the area is going to rise. Hence, it might be appropriate to deal with the uncertainties using the *probabilistic model*,

$$\mathbb{P}(Y = y | X_1 = x_1, X_2 = x_2, \ldots, X_m = x_m).$$

Here, Y and X_i's are *random variables* that can have multiple values. Assume, for simplicity, that all variables are discrete, then the most straightforward way to build the model is to compute the joint probability

$$\mathbb{P}(Y = y, X_1 = x_1, X_2 = x_2, \ldots X_m = x_m),$$

for all possible y and x_i's. However, the computation becomes intractable when the number of variables is very high as in the case of DNA sequences or images. We therefore introduce the *probabilistic graphical model* framework represents the relations between variables in a compact way and in turn reduces the amount of computation.

Example 1.1. To illustrate this point, we consider the problem of medical diagnosis between cold and allergy, where we also have the following related factors: season and symptoms of having a fever and congestion. Then the number of joint probabilities we need to compute is $2 \times 2 \times 4 \times 2 \times 2 = 64$ which is already relatively high compared to the size of the problem. Alternatively, we could use our prior knowledge about the relations among these variables and create the graphical model as in Figure 1.1. This model indicates

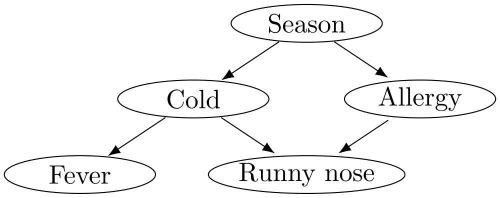

Figure 1.1: Graphical representation of the medical diagnosis between cold and allergy.

the follows independencies:

$$\begin{aligned}(C \perp\!\!\!\perp A|S)\\(A \perp\!\!\!\perp F, C|S)\\(F \perp\!\!\!\perp R, A, S|C)\\(R \perp\!\!\!\perp F, S|C, A)\end{aligned}$$

We will expand on how to find these later in the course. Together with $\Pr(S)$, the joint distribution of five variables can be computed using

$$\mathbb{P}(S, C, A, F, R) = \mathbb{P}(S)\mathbb{P}(C|S)\mathbb{P}(A|S)\mathbb{P}(F|C)\mathbb{P}(R|C, A).$$

In other words, the full probabilistic model can be *parametrized* by the following five quantities: $\mathbb{P}(S)$, $\mathbb{P}(C|S)$, $\mathbb{P}(A|S)$, $\mathbb{P}(F|C)$ and $\mathbb{P}(R|C, A)$. To completely characterize the model, we are now required to compute only $3 + 4 + 4 + 2 + 4 = 17$ parameters instead of 63. Therefore, by injecting prior knowledge into the probabilistic model, we can reduce it into a representation whose distribution can be tractably estimated from data. ◇

Example 1.2. Another example comes from the most basic spam filtering method – the Naïve Bayes classifier. Suppose that we want to classify each email into either spam ($y = 1$) or not spam ($y = 0$) from the words in the email. Then the probabilistic model we want to look at is

$$\mathbb{P}(y = 1|x_1, x_2, \ldots, x_m),$$

where x_i indicates whether the i-th English word in the dictionary appears in the email. To characterize the full model, we would need to compute $2^{m+1} - 1$ parameters which is intractable even with modern computing power. The idea behind the Naïve Bayes classifier is to simplify the model by imposing an assumption that all words are *conditionally independent* i.e.

$$\mathbb{P}(x_1, x_2, \ldots, x_m|y) = \mathbb{P}(x_1|y)\mathbb{P}(x_2|y)\ldots\mathbb{P}(x_m|y),$$

which has a graphical representation as in Figure 1.2. Then we have the following factorization of the joint probability

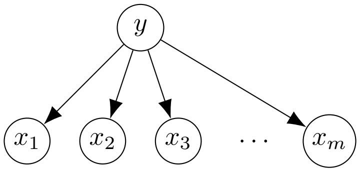

Figure 1.2: Graphical representation of the Naïve Bayes classifier.

$$\mathbb{P}(y, x_1, x_2, \ldots, x_m) = \mathbb{P}(y)\mathbb{P}(x_1|y)\mathbb{P}(x_2|y)\ldots\mathbb{P}(x_m|y),$$

which requires only $2m + 1$ nonredundant parameters. ◇

1.2 Applications

There are many more tasks that benefit from graphical models. Here are some examples.

- **Images**: Each node in the graph represents each pixel and every two nodes are connected if they are adjacent. If we consider a *segmentation problem* that separate the foreground object from the background, then we might want to impose some probabilistic conditions that encourage the adjacent pixels with similar color to have the same label.

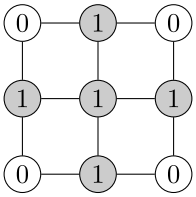

Figure 1.3: A graphical representation of an image.

- **DNA sequencing**: We can model a long sequence of DNA genomes by a Markov chain, where each node can take values in $\{A, C, G, T\}$.

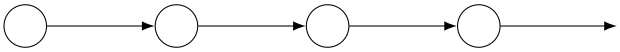

Figure 1.4: A Markov chain.

- **Part-of-speech tagging**: We can also fit the task of tagging each word in a sentence with its part of speech in a probabilistic framework. One way to do this is to treat each tag as a *hidden states* that is connected to the corresponding word and has a probability of transitioning into each of all possible part of speeches. Then, an appropriate sequence of tags is the one that maximize the probability of the observed sentence. This is an example of a *hidden Markov model*.

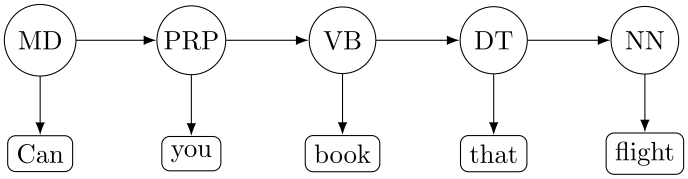

Figure 1.5: A hidden Markov model for part-of-speech tagging

- **Click modeling** (Chapelle and Zhang, 2009): We can also build a personalized model that can predict whether a user will click on a particular link in a search list. Let i be the position of the link and C_i indicates whether the user click on it. Then we can model C_i using several hidden binary variables:

  - $E_i = 1$ if the user examined the url, 0 otherwise.
  - $A_i = 1$ if the user was attracted to the url, 0 otherwise.
  - $S_i = 1$ if the user was satisfied by the page, 0 otherwise,

  together with some probability conditions that are conform with these definitions. The resulting model (Figure 1.6) is a *dynamic Bayesian network* which relates variables to each others across the sequence. After the inference, we can use the model to compute the probability that the user will click the i-th url in a search query.

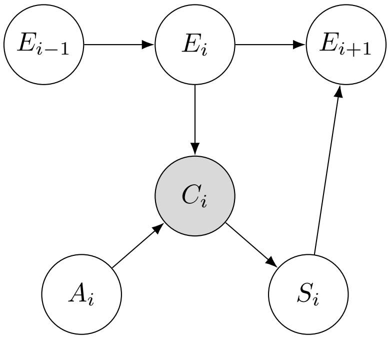

Figure 1.6: Dynamic Bayesian network of the click model

1.3 Basic definitions

1.3.1 Graphs

As we have seen, there are two types of graphs depending on whether the nodes are connected by simple edges or directed by arrows.

Definition 1.3. A (undirected) graph is a pair of set $G = (V, E)$ where $V = \{X_1, X_2, \ldots, X_n\}$ is the set of nodes and $E \subseteq \{\{X_i, X_j\}, 1 \leq i \neq j \leq n\}$ is the set of edges. We say that G is a directed graph if the edges are the ordered pairs of nodes i.e. $E \subseteq \{(X_i, X_j), 1 \leq i \neq j \leq n\}$

Definition 1.4. We write $X_i - X_j$ for $\{X_i, X_j\} \in E$. In this case, we say that X_i is a *neighbor* of X_j. The set of all neighbors of X_i is denoted by $\mathcal{N}(X_i)$.

Definition 1.5. We write $X_i \to X_j$ for $(X_i, X_j) \in E$. In this case, we say that X_i is a *parent* of X_j and X_j is a *child* of X_i. The set of all parents of X_j is denoted by $\mathrm{Pa}(X_j)$ and the set of all children of X_i is denoted by $\mathrm{Ch}(X_i)$.

Definition 1.6. We say that X_i is *adjacent* to X_j if $X_i \to X_j$ or $X_j \to X_i$.

We sometimes consider a type of graphs that contain all possible edges.

Definition 1.7. A graph G is *complete* if every pair of node is connected by an edge.

1.4 Subgraphs

Definition 1.8. $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$, $E' \subseteq E$ and for any $\{X_i, X_j\} \in E'$ (or $(X_i, X_j) \in E'$) we have $X_i, X_j \in V'$.

Definition 1.9. A *clique* C in a graph G is a complete subgraph of G. We say that a clique is *maximal* if there is no other clique in G that strictly contains C.

1.5 Paths, cycles and connected components

Definition 1.10. A path in a graph $G(V, E)$ is a sequence of distinct nodes $(X_1, X_2, \ldots, X_k)$ such that $(X_i, X_{i+1}) \in E$ (or $\{X_i, X_j\} \in E$) for all $i = 1, 2, \ldots, k - 1$.

Therefore, in a directed graph G, we can traverse along any path in the direction of the edges. On the other hand, if a sequence in G contains a subgraph of the form $X_{i-1} \leftarrow X_i \rightarrow X_{i+1}$ or $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then it is not a path. The following names will be used to relate any two distinct nodes in a path.

Definition 1.11. In a directed graph. A node X is an *ancestor* of a node Y if there is a path from X to Y.

Definition 1.12. In a directed graph. A node X is a *decendant* of a node Y if there is a path from Y to X.

Sometimes we want to travel along edges ignoring their directions. In this case, we might want to consider *trails*.

Definition 1.13. A sequence of distinct nodes $(X_1, X_2, \ldots, X_k)$ in a directed graph G is a trail if it is a path in the undirected version of G.

Definition 1.14. An undirected graph is *connected* if for every distinct X_i, X_j there is a path from X_i to X_j. A directed graph is connected if its undirected version is connected.

Definition 1.15. A *connected component* H is a maximal connected subgraph, that is, there is no larger connected subgraph that strictly contains H.

Definition 1.16. A cycle in a graph $G = (V, E)$ is a sequence of nodes $(X_1, X_2, \ldots, X_k)$ such that $(X_i, X_{i+1}) \in E$ for all $i = 1, 2, \ldots, k - 1$, $X_1 = X_k$ and $X_i \neq X_j$ for $(i, j) \neq (1, k)$.

Definition 1.17. A directed acyclic graph (DAG) is a directed graph that contains no cycles.

Example 1.18. The graph in Figure 1.7 has two connected components, a path (b, a, e, f, g), a cycle (a, b, c, d, e, a) and a maximal clique (a, b, c, d). $\mathcal{N}(a) = \{b, c, d, e\}$ and $\mathcal{N}(f) = \{e, g\}$. ⋄

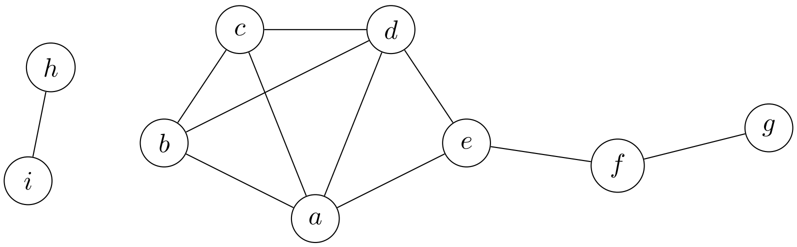

Figure 1.7: A graph with two connected components.

Example 1.19. In the directed graph in Figure 1.8, a, b are ancestors of c and d, e are descendant of c. Here, (b, c, e, b) is a cycle while (a, c, d, a) is not. ⋄

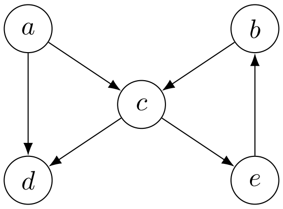

Figure 1.8: An example of a directed graph

References

Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10. ACM, 2009.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

Volodymyr Kuleshov and Stefano Ermon. Stanford CS228 lecture notes. https://ermongroup.github.io/cs228-notes, 2019. Accessed: 2019-12-20.