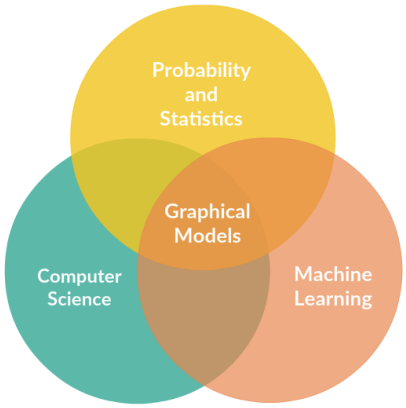
IFT 6269: Probabilistic Graphical Models **Fall 2017**

Lecture 1 — September 5

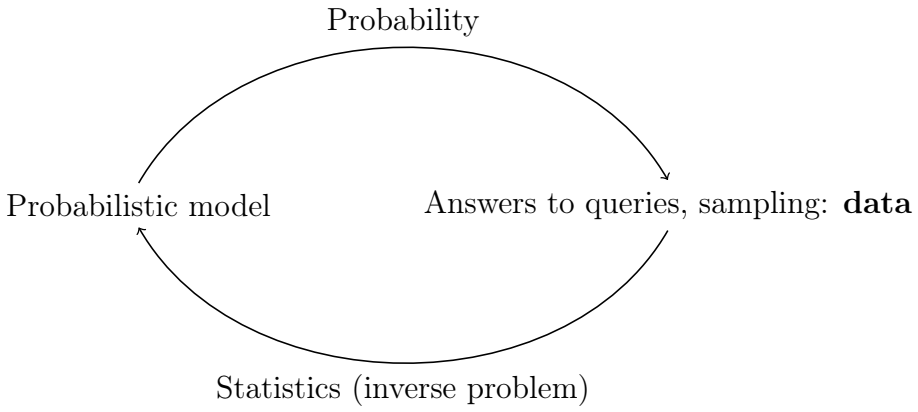
Lecturer: Simon Lacoste-Julien *Scribe: Isabela Albuquerque*

1.1 Probabilistic Graphical Models

- Goal: Model multivariate data
- Mix of graph and probability theory. Or, more illustratively:



- Probability vs. Statistics:



1.2 Applications

Some illustrative examples of Hidden Markov Models (HMM) applications.

Notation:

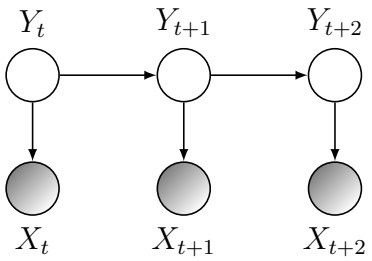
- X_t: **Observed** random variable. Represented in the graphical model as a **shaded** node.
- Y_t: **Not observed** random variable. Represented in the graphical model as an **empty** node.
- Graph edges ($-$): Represents possible correlations between random variables in the graphical models. Lack of edges in the graph will represent **conditional independence** assumptions, as we will see later.

Important!

- When modeling a problem using graphical models, random variables represent the quantities of interest.
- In the context of PGM, a random vector is often just called a *random variable* – thus a random variable might be scalar or vector valued.

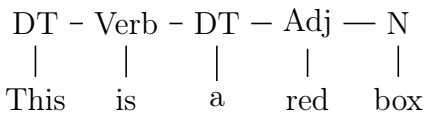
1.2.1 Example 1: Speech Recognition

X_t: Sound wave encoding for a small time window (e.g. as a spectral decomposition)
Y_t: Phoneme



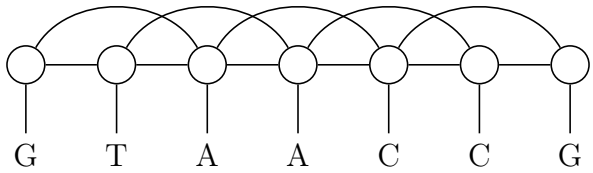
1.2.2 Example 2: Part-of-speech tagging

X_t: Word
Y_t: Part-of-speech (word grammatical classification)



1.2.3 Example 3: Gene finding

X_t: Sequence of nitrogenous base
Y_t: Coding or non-coding (i.e. $Y_t \in \{0; 1\}$)



1.2.4 Example 4: Control system

$$y_{t+1} = Ay_t + Bv_t + \epsilon_t, \qquad (y_t \text{ is the latent state})$$
$$x_t = Cy_t + \epsilon'_t, \qquad (\text{Observation})$$

where y_t and x_t are continuous vectors and v_t is a given control term. The terms ϵ and ϵ' represent the noise in the system. If they are modeled as Gaussian noise, this HMM is a Kalman Filter.

1.3 Why Graphical Models?

Back to the part-of-speech tagging example:

Notation:

- An observation of T words is represented as $(x_1, x_2, \ldots, x_T) \triangleq x_{1:T}$
- For a vocabulary of size k, $x_t \in \{1, \ldots, k\}$

Problem: We want to model $p(x_{1:T})$, which corresponds to an exponential size state space. Thus, $\approx K^T$ parameters have to be estimated to define a probability distribution on $x_{1:T}$

Trick: make a factorization assumption about the distribution $p(x_{1:T})$.

$$p(x_1, \ldots, x_T) = f_1(x_1) f_2(x_2|x_1) f_3(x_3|x_2) \ldots f_T(x_T|x_{T-1}).$$

Each factor f can be seen as a clique in the graphical model and needs $\approx K^2$ parameters to be specified. As we have T factors in this factorization, we reduce the total number of parameters from K^T (exponentially grows with T) to TK^2 (linearly grows with T).

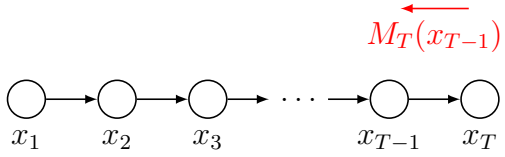
Now, back to our problem, say we want to compute the marginal probability of x_1, $p(x_1) = \sum_{x_2,\dots,x_T} p(x_{1:T})$. Using the factorization assumption, we can write $p(x_1)$ as:

$$p(x_1) = \sum_{x_2,\dots,x_T} f_1(x_1) f_2(x_2|x_1) f_3(x_3|x_2) \dots f_T(x_T|x_{T-1}). \tag{1.1}$$

Applying the distributive property of the product over a sum ($a(b+c) = ab + ac$), we can rewrite equation 1.1 as

$$p(x_1) = f_1(x_1) \left(\sum_{x_2} f_2(x_2|x_1) \left(\sum_{x_3} f_3(x_3|x_2) \dots \left(\sum_{x_T} f_T(x_T|x_{T-1}) \right) \dots \right) \right). \tag{1.2}$$

This organized and efficient way to compute the marginal $p(x_1)$ is known as the **Message passing algorithm**. The term $\sum_{x_T} f_T(x_T|x_{T-1})$ is named message and denoted as $M_T(x_{T-1})$. The following figure illustrates $M_T(x_{T-1})$ (represented by the red arrow) passing through a graph.



1.4 Key Themes

1. Representation: how to represent structured probability distributions.
   - Related to parameterization (*e.g.* full table, exponential family)
2. Estimation: given data samples, how do we learn the parameters of the distribution underlying the observations?
   - Related to learning (*e.g.* Maximum Likelihood Estimation)
3. Inference: answer questions about the data, as computing conditional distributions $p(y|x)$ or marginals $p(x_1)$.
   - Efficient computation (*e.g.* Message passing algorithm)