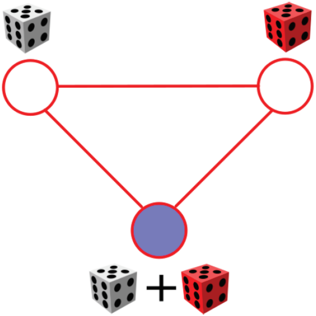
Deep Learning

Probabilistic graphical models



Learning goals

- probabilistic graphical models
- latent variables
- directed graphical models

Probabilistic graphical models

GRAPHICAL MODELS

Probabilistic graphical models describe probability distributions by mapping conditional dependence and independence properties between random variables on a graph structure.

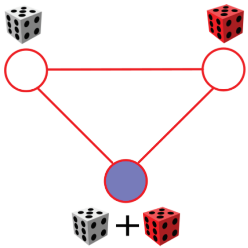
WHY AGAIN GRAPHICAL MODELS?

1. Graphical models visualize the structure of a probabilistic model; they help to develop, understand and motivate probabilistic models.

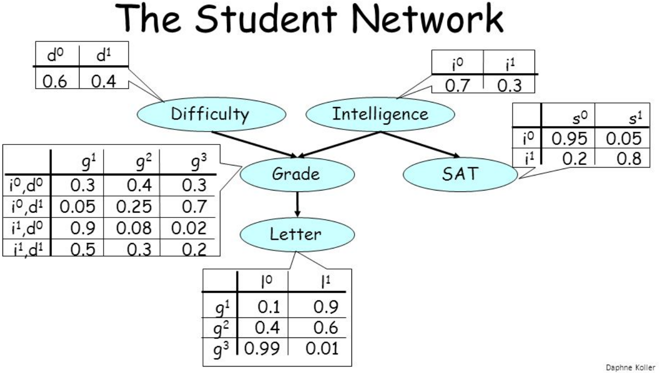
WHY AGAIN GRAPHICAL MODELS?

1. Graphical models visualize the structure of a probabilistic model; they help to develop, understand and motivate probabilistic models.

2. Complex computations (e.g., marginalization) can derived efficiently using algorithms exploiting the graph structure.



GRAPHICAL MODELS: EXAMPLE



Credit: Daphne Koller

Figure: A graphical model representing five variables and their (in-)dependencies along with the corresponding marginal and conditional distributions. The variable 'Grade', for example, is affected by 'Difficulty' (of the exam) and 'Intelligence' (of the student). This is captured in the corresponding conditional distribution. 'Letter' refers to a letter of recommendation. In this model, 'Letter' is conditionally independent of 'Difficulty' and 'Intelligence', given 'Grade'.

Latent Variables

LATENT VARIABLES: MOTIVATION

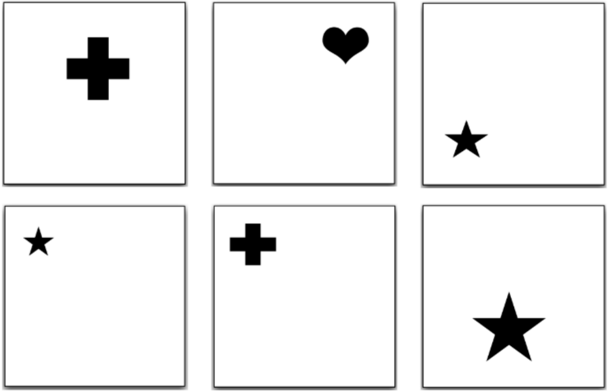


Figure: A simple illustration of the relevance of latent variables. Here, six 200 x 200 pixel images are shown where each pixel is either black or white. Naively, the probability distribution over the space of all such images would need $2^{40000} - 1$ parameters to fully specify. However, we see that the images have three main factors of variation : object type (shape), position and size. This suggests that the actual number of parameters required might be significantly fewer.

LATENT VARIABLES

- Additional nodes, which do not directly correspond to observations, allow to describe complex distributions over the visible variables by means of simple conditional distributions.
- The corresponding random variables are called *hidden* or *latent* variables.

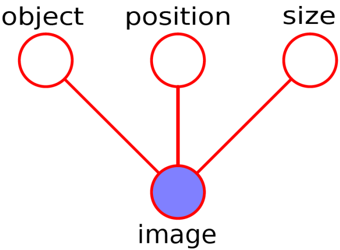


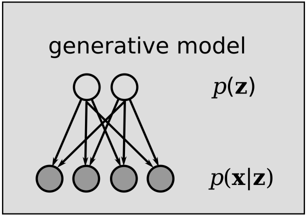
Figure: 'Object', 'position' and 'size' are the latent variables behind an image.

Directed generative models

DIRECTED GENERATIVE MODELS

Goal: Learn to generate **x** from some latent variables **z**

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z}) dz = \int p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}) dz$$



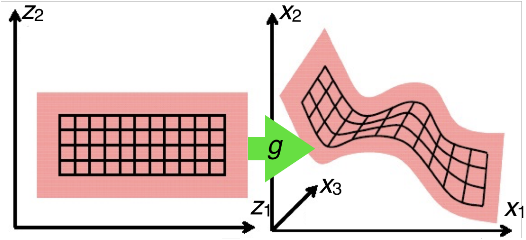


Image from: Ward, A. D., Hamarneh, G.: **3D Surface Parameterization Using Manifold Learning for Medial Shape Representation**, Conference on Image Processing, Proc. of SPIE Medical Imaging, 2007

Figure: *Left*: An illustration of a directed generative model. *Right*: A mapping (represented by g) from the 2D latent space to the 3D space of observed variables.

DIRECTED GENERATIVE MODELS

The latent variables $\mathbf{z}$ must be learned from the data (which only contains the observed variables $\mathbf{x}$).

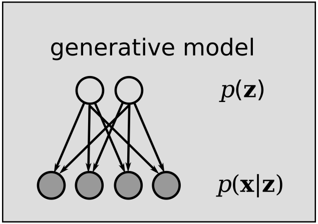
- The posterior is given by $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$.
- But $p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})d\mathbf{z}$ is intractable and common algorithms (such as Expectation Maximization) cannot be used.

DIRECTED GENERATIVE MODELS

The latent variables **z** must be learned from the data (which only contains the observed variables **x**).

- The posterior is given by $p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})}$.
- But $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$ is intractable and common algorithms (such as Expectation Maximization) cannot be used.

The classic DAG problem: How do we efficiently learn $p_\theta(\mathbf{z}|\mathbf{x})$?



Popular approaches to this problem:

- **Variational Autoencoders (VAEs)**
- **Generative Adversarial Networks (GANs)**