Deep Generative Models (Background)

Dr. Zulqarnain Khan

Slides from : Stefano Erman

Overview

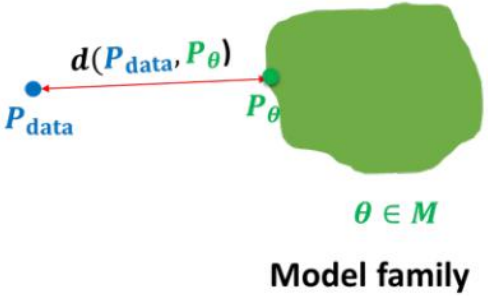
- What is a generative model
- Representing probability distributions
  - Curse of dimensionality (what makes generative modeling hard?)
  - Crash course on graphical models (Bayesian networks) – to get familiar with how to represent joint in the form conditionals
  - Generative vs discriminative models – why do generative
  - Neural models – from generative to *deep* generative

Learning a Generative Model

Learning a Generative Model

- We are given a training set of examples, e.g., images of dogs
- We want to learn a probability distribution $p(x)$ over images 'x' such that we can do:
  - **Generation:** If we sample $x_{new} \sim p(x)$, x_{new} should look like a dog (sampling)
  - **Density Estimation:** $p(x)$ should be high if 'x' looks like a dog, and low otherwise (anomaly detection)
  - **Unsupervised representation learning:** We should be able to learn what these images have in common, e.g., ears, tail, etc. (features) – possible with at least some of the generative models.
- First question: how to represent $p(x)$

$\mathbf{x}_i \sim P_{\text{data}}$
$i = 1, 2, \ldots, n$



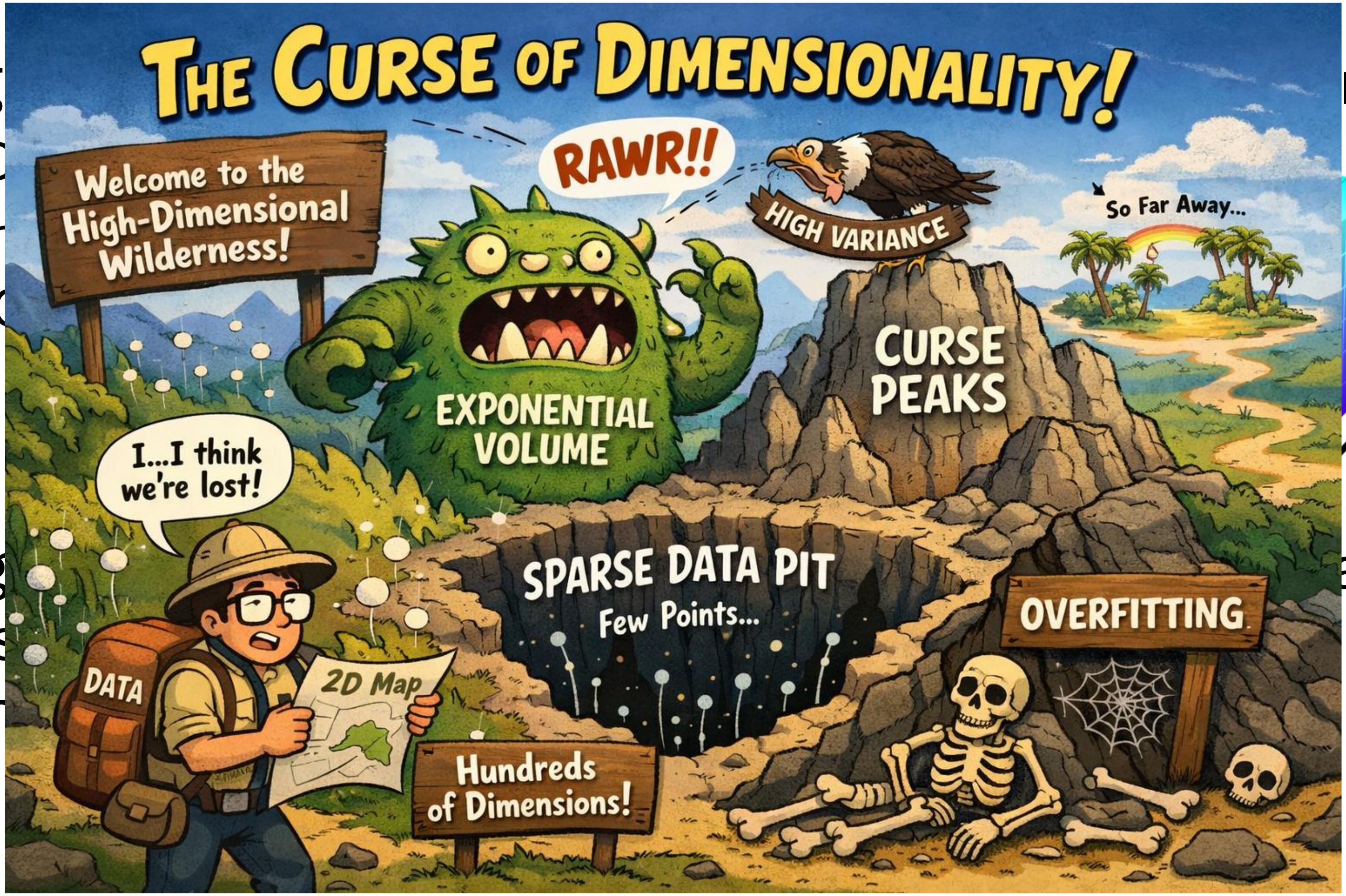
Basic Discrete Distributions

- Bernoulli distribution: (biased) coin flip
  - D = {Heads,Tails}
  - Specify P(X = Heads) = p. Then P(X = Tails) = 1 − p.
  - Write: X ∼ Ber(p)
  - Sampling: flip a (biased) coin
- Categorical distribution: (biased) m-sided dice
  - D = {1, · · · , m}
  - Specify $P(Y = i) = p_i$, such that $\sum p_i = 1$
  - Write: $Y \sim Cat(p_1, \cdots, p_m)$
  - Sampling: roll a (biased) die

Example of joint distribution

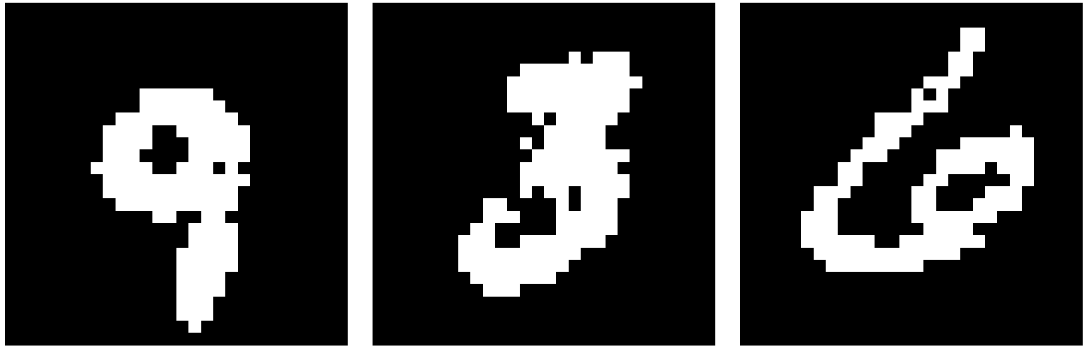
Modeling riables:

- Red C
- Greer
- Blue (



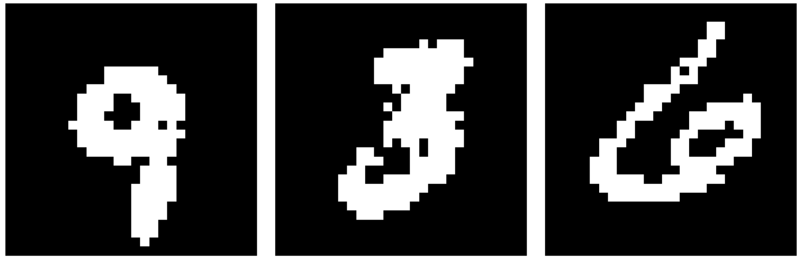
Sampling andomly
generates we need to
specify th

Example of joint distribution

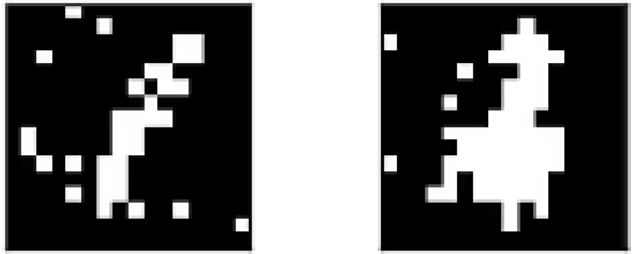


- Suppose $X_1, \ldots, X_n$(n pixels) are binary (Bernoulli) random variables i.e.
$$Val(X_i) = \{0, 1\} = \{Black, White\}$$
- How many possible images?
- Sampling from $p(x_1, \ldots, x_n)$ generates an image
- How many parameters to specify the joint distribution$p(x_1, \ldots, x_n)$ over n binary pixels?
- These are really big numbers even with black and white pixels!
- We clearly can't do this in full generality. **Solution: Make assumptions!**

Structure through independence



- If $X_1, \ldots, X_n$ are independent, then
$$p(x_1, x_2, \ldots, x_n) = ?$$
- How many possible images?
- How many parameters now?
- However, independence assumption is too strong. Model not likely to be useful. We are choosing each pixel independently when we sample from this model, and will get:



Two Important Rules

Chain rule Let $S_1, \ldots S_n$ be events, $p(S_i) > 0$.

$$p(S_1 \cap S_2 \cap \cdots \cap S_n) = p(S_1)p(S_2 \mid S_1) \cdots p(S_n \mid S_1 \cap \ldots \cap S_{n-1})$$

Bayes' rule Let S_1, S_2 be events, $p(S_1) > 0$ and $p(S_2) > 0$.

$$p(S_1 \mid S_2) = \frac{p(S_1 \cap S_2)}{p(S_2)} = \frac{p(S_2 \mid S_1)p(S_1)}{p(S_2)}$$

Structure through conditional independence

- Using Chain Rule

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_n \mid x_1, \cdots, x_{n-1})$$

- How many parameters? Still $2^n - 1$ (why?)
  - $p(x_1)$ requires 1 parameter
  - But, $p(x_2|x_1)$ requires 2 (one each for $x_1 = 0$, 1), and so on...
- Still exponential, chain rule alone doesn't help
- Now suppose $\mathrm{X_{i+1}} \perp X_1, \ldots, X_{i-1} \mid X_i$ (aka Markov), then

$$\begin{aligned} p(x_1, \ldots, x_n) &= p(x_1)p(x_2 \mid x_1)p(x_3 \mid \cancel{x_1}, x_2) \cdots p(x_n \mid \cancel{x_1, \cdots,} x_{n-1}) \\ &= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \cdots p(x_n \mid x_{n-1}) \end{aligned}$$

- This gets us to 2n – 1 parameters, exponential reduction! You are essentially saying that you only care about the previous variable (e.g. previous word).

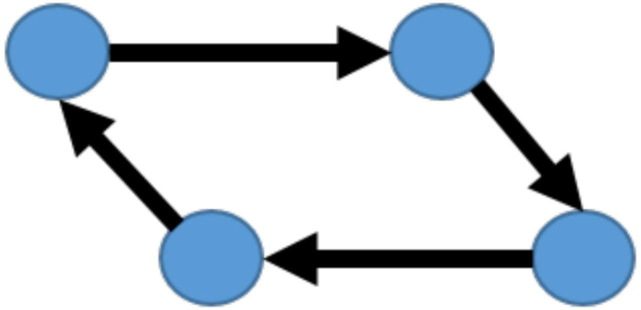
Bayes Network: General Idea

- Use conditional parameterization (instead of joint parameterization)
- For each random variable X_i , specify $p(x_i|\boldsymbol{x}_{\boldsymbol{A_i}})$ for set $\boldsymbol{x}_{\boldsymbol{A_i}}$ of random variables
- Then get joint parametrization as

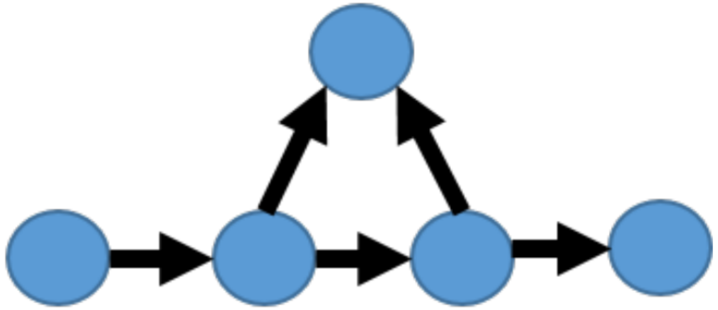
$$p(x_1, \ldots, x_n) = \prod_i p(x_i|\mathbf{x}_{\mathbf{A_i}})$$

- Has to be a valid probability (correspond graph needs to be a DAG)

Bayes Network: General Idea



Directed cycle



DAG

DAG stands for Directed Acyclic Graph

Bayesian networks

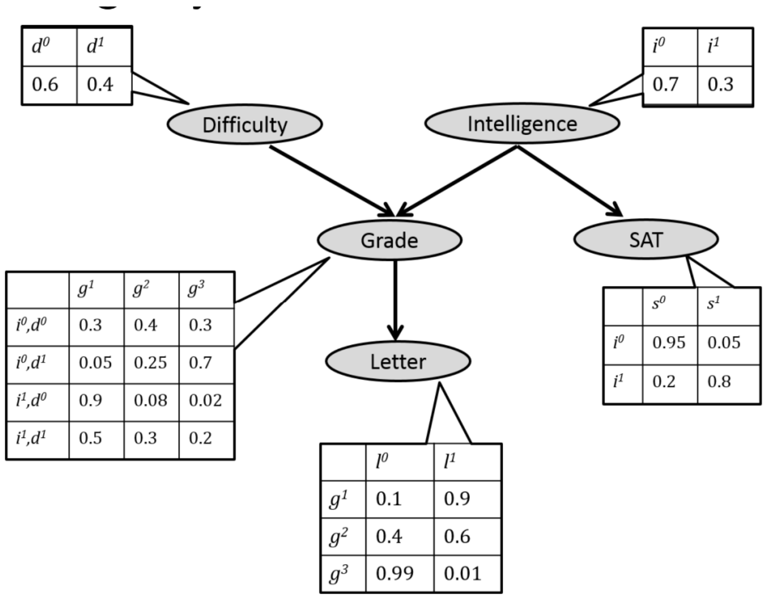
- A Bayesian network is specified by a directed acyclic graph (DAG) G = (V, E) with:
  - One node i ∈ V for each random variable X_i
  - One conditional probability distribution (CPD) per node, $p(x_i|\boldsymbol{x}_{\boldsymbol{Pa(i)}})$specifying the variable's probability conditioned on its parents' values
- Graph G = (V, E) is called the structure of the Bayesian Network
- Defines a joint distribution:

$$p(x_1, \ldots x_n) = \prod_{i \in V} p(x_i \mid \mathbf{x}_{\mathrm{Pa}(i)})$$

- Economical representation: exponential in |Pa(i)|, not |V|

Example

- Consider the following Bayesian network:



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

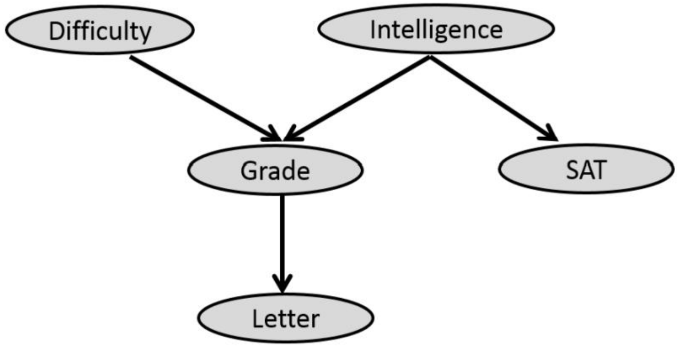
	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

- What is its joint distribution? How many parameters?

Bayesian network structure implies conditional independencies!



- The joint distribution corresponding to the above BN factors as

$$p(d, i, g, s, l) = p(d)p(i)p(g \mid i, d)p(s \mid i)p(l \mid g)$$

- Generally, by the chain rule, any distribution can be written as

$$p(d, i, g, s, l) = p(d)p(i \mid d)p(g \mid i, d)p(s \mid i, d, g)p(l \mid g, d, i, s)$$

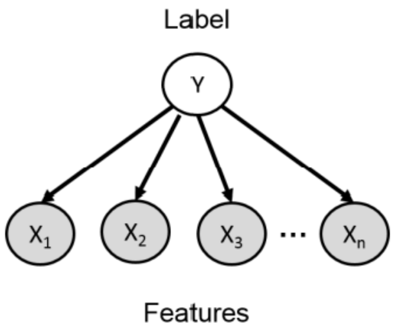
- Here, we are assuming the following additional independencies:

$$D \perp I, \qquad S \perp \{D, G\} \mid I, \qquad L \perp \{I, D, S\} \mid G$$

Summary

- Bayesian networks given by (G, P) where P is specified as a set of local conditional probability distributions associated with graph G's nodes
- Efficient representation using a graph-based data structure
- Computing the probability of any assignment is obtained by multiplying CPDs
- Can sample from the joint by sampling from the CPDs according to the DAG ordering
- Can identify some conditional independence properties by looking at graph properties
- In this class, graphical models will be simple (e.g., only 2 or 3 random vectors)
- **Next: generative vs. discriminative**

Naive Bayes for single label prediction



- Classify e-mails as spam (Y = 1) or not spam (Y = 0)
  - Let 1 : n index the words in our vocabulary (e.g., English)
  - $X_i = 1$ if word i appears in an e-mail, and 0 otherwise
  - E-mails are drawn according to some distribution $p(Y, X_1, \ldots, X_n)$
- Suppose that the words are conditionally independent given Y . Then,

$$p(y, x_1, \ldots x_n) = p(y) \prod_{i=1}^{n} p(x_i \mid y)$$

Estimate parameters from training data. Predict with Bayes rule:

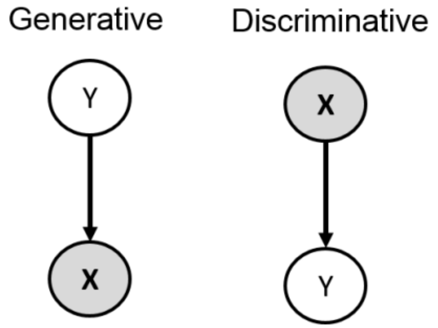
$$p(Y = 1 \mid x_1, \ldots x_n) = \frac{p(Y = 1) \prod_{i=1}^{n} p(x_i \mid Y = 1)}{\sum_{y=\{0,1\}} p(Y = y) \prod_{i=1}^{n} p(x_i \mid Y = y)}$$

Naive Bayes for single label prediction

- Did we make reasonable assumptions in the previous slide?
- Answer: depends. All models are “wrong”, but many are nonetheless useful

Discriminative versus generative models

- Using chain rule $p(Y, X) = p(X \mid Y)p(Y) = p(Y \mid X)p(X)$. Corresponding Bayesian networks:



- However, **suppose if all we need for prediction is $p(Y \mid X)$**
- In the left model, we need to specify/learn both $p(Y)$ and $p(X \mid Y)$, then compute $p(Y \mid X)$ via Bayes rule
- In the right model, it suffices to estimate just the conditional distribution $p(Y \mid X)$
  - We never need to model/learn/use $p(X)$!
  - Called a discriminative model because it is only useful for discriminating Y's label when given X

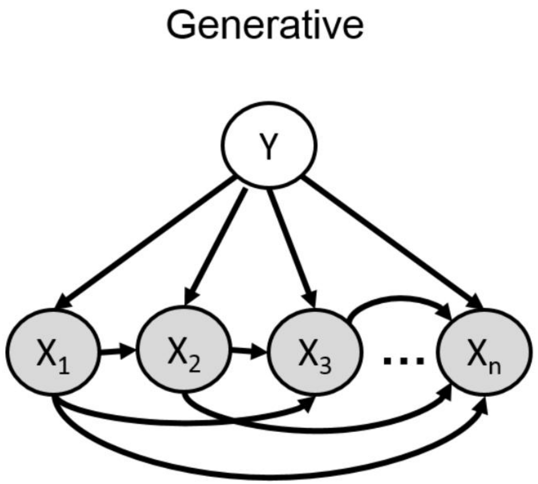
Discriminative versus generative models

- If all we care about is being able to predict a label -> discriminative
- If we also care about the relationships of all the variables x and y in the data -> generative

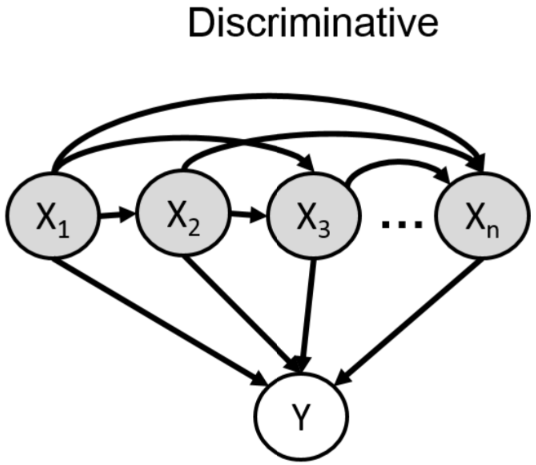
Discriminative versus generative models

Since X is a random vector, chain rule will give:

$$p(Y, \mathbf{X}) = p(Y)p(X_1 \mid Y)p(X_2 \mid Y, X_1) \cdots p(X_n \mid Y, X_1, \cdots, X_{n-1})$$



$$p(Y, \mathbf{X}) = p(X_1)p(X_2 \mid X_1)p(X_3 \mid X_1, X_2) \cdots p(Y \mid X_1, \cdots, X_{n-1}, X_n)$$



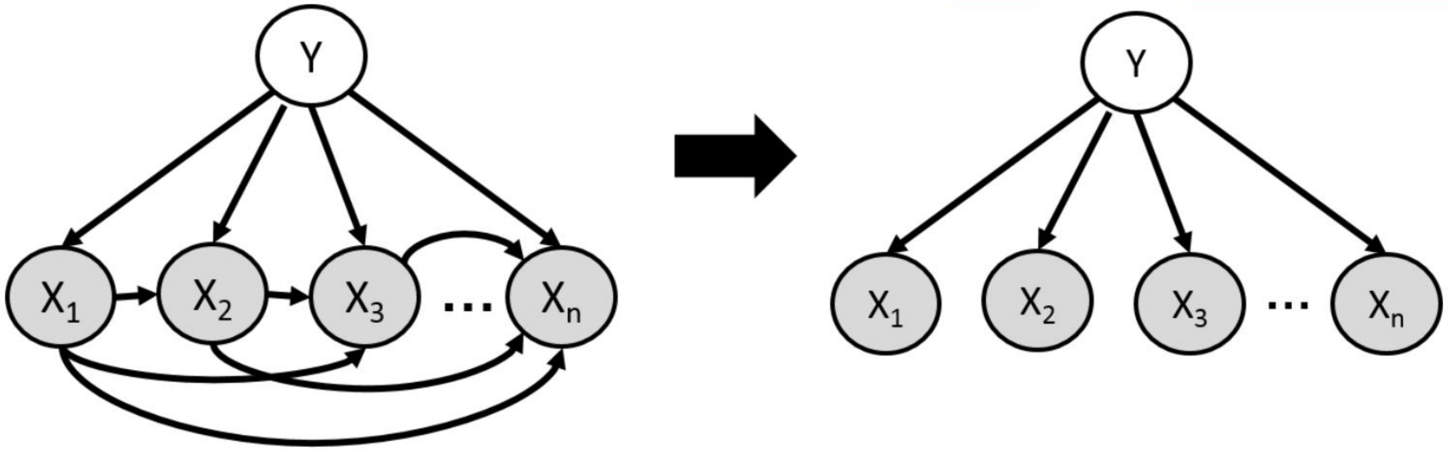
Discriminative versus generative models

We must make the following choices:

- In the generative model, p(Y) is simple, but how do we parameterize $p(X_i|\boldsymbol{X}_{\boldsymbol{pa(i)}}, Y)$?
- In the discriminative model, how do we parameterize p(Y | X)? Here we assume we don't care about modeling p(X) because X is always given to us in a classification problem

Naive Bayes

- For the generative model, assume that $X_i \perp \boldsymbol{X_{-i}} | Y$ (naive Bayes)



Logistic regression

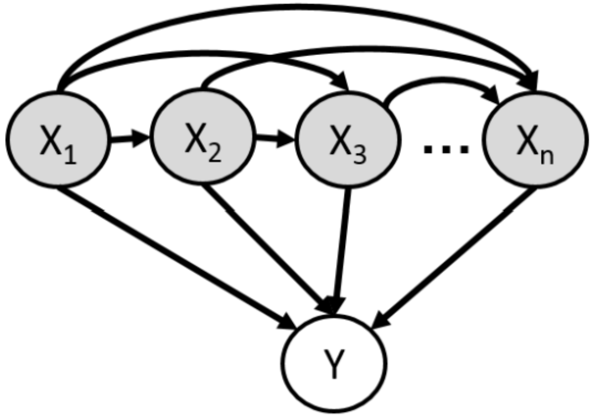
- For the discriminative model, assume that

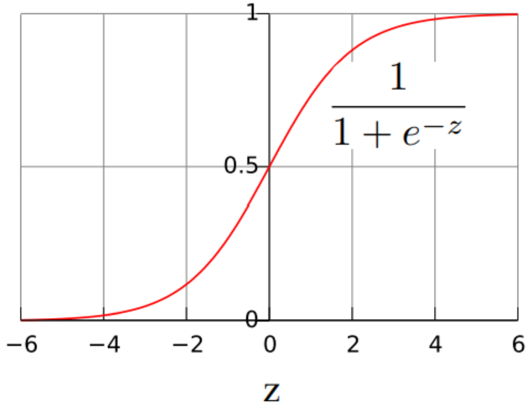
$$p(Y = 1 \mid \mathbf{x}; \alpha) = f(\mathbf{x}, \alpha)$$

- Not represented as a table anymore. It is a parameterized function of x (regression)
    - Has to be between 0 and 1
    - Depend in some simple but reasonable way on $x_1, \ldots, x_n$
    - Completely specified by a vector α of n + 1 parameters (compact representation)

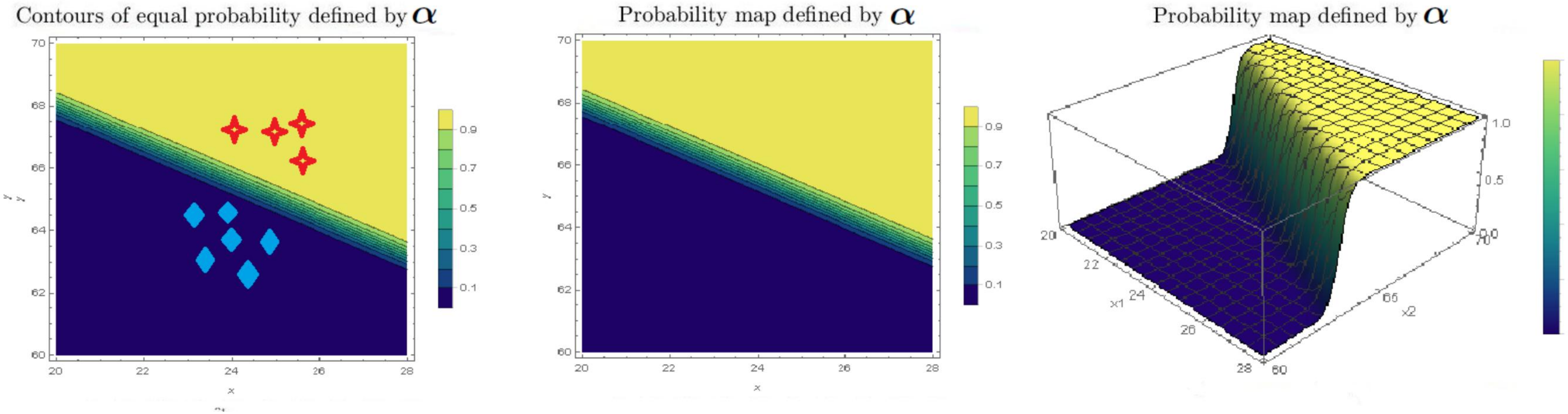
Logistic Regression

- **Linear dependence**: Let $z(\alpha, x) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i$. Then, $p(Y = 1\,|\boldsymbol{x}; \boldsymbol{\alpha}) = \sigma\big(z(\alpha, x)\big), where\ \sigma(z) = 1\backslash 1 + e^{-z}\ is\ the\ logistic\ function$.





Logistic Regression



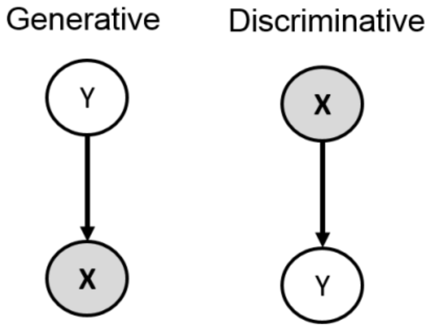
- Decision boundary p(Y = 1 | x; α) > 0.5 is linear in x
- Equal probability contours are straight lines
- Probability rate of change has very specific form (third plot)

Discriminative models can be powerful

- Make fewer assumptions. e.g. Logistic model does not assume $X_i \perp X_{-i} | Y$, unlike naive Bayes.
- This is helpful when there is plenty of data available and you are only interested in prediction.
- This can make a big difference in many applications.

Generative models are still very useful

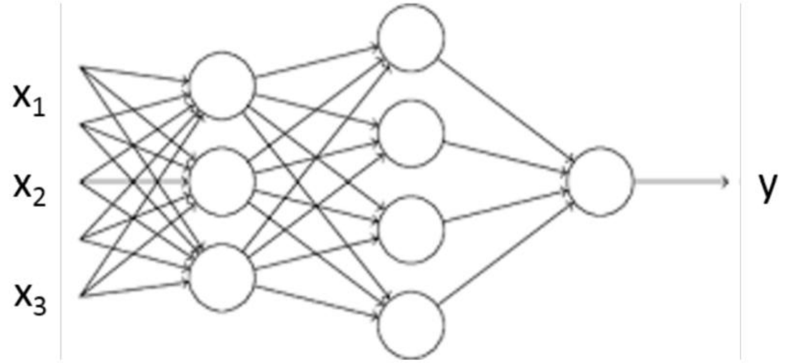
- Using chain rule p(Y , X) = p(X | Y)p(Y) = p(Y | X)p(X). Corresponding Bayesian networks:



- Using a discriminative conditional model is only possible when X is always observed.
  - When some X_i variables are unobserved, the generative model allows us to compute $p(Y|X_{evidence})$ by marginalizing over the unseen variables
  - Priors can help when there isn't enough data.

Neural Models

- In discriminative models, we assume that
$$p(Y = 1 \mid \mathbf{x}; \boldsymbol{\alpha}) = f(\mathbf{x}, \boldsymbol{\alpha})$$
- Linear dependence: Let $z(\alpha, x) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i$. Then, $p(Y = 1 \,|\boldsymbol{x}; \boldsymbol{\alpha}) = \sigma\big(z(\alpha, x)\big), where\ \sigma(z) = 1\backslash 1 + e^{-z}\ is\ the\ logistic\ function$. This dependence might be too simple.
- Non-Linear dependence: Let $h(A, b, x) = f(Ax + b)$ be a non-linear transformation of the inputs. Which makes $p(Y = 1 \,|\boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{A}, \boldsymbol{b}) = f(\alpha_0 + \sum_{i=1}^{h} \alpha_i h_i)$
    - More flexible
    - More parameters: A, b, α
    - Can repeat multiple times to get a neural network



Bayesian networks vs neural models

- Using Chain Rule (fully general)

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2)p(x_4 \mid x_1, x_2, x_3)$$

- Bayes Net (assumes conditional independencies)

$$p(x_1, x_2, x_3, x_4) \approx p(x_1)p(x_2 \mid x_1)p(x_3 \mid \cancel{x_1}, x_2)p(x_4 \mid x_1, \cancel{x_2, x_3})$$

- Neural Models (assume specific functional form for the conditionals. A sufficiently deep neural net can approximate any function.)

$$p(x_1, x_2, x_3, x_4) \approx p(x_1)p(x_2 \mid x_1)p_{\text{Neural}}(x_3 \mid x_1, x_2)p_{\text{Neural}}(x_4 \mid x_1, x_2, x_3)$$

Continuous variables

- If X is a continuous random variable (scalar), we can usually represent it using its probability density function. Typically consider parameterized densities, like Gaussian, or Uniform.
- Correspondingly for a continuous random vector X, we can usually represent it using its *joint* probability density function. E.g.

$$\text{Gaussian: if } p_X(x) = \frac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})\right)$$

- **Chain rule, Bayes rule, etc all still apply**. For example,

$$p_{X,Y,Z}(x,y,z) = p_X(x)p_{Y|X}(y \mid x)p_{Z|\{X,Y\}}(z \mid x,y)$$

Continuous variables

- This means we can still use Bayesian networks with continuous (and discrete) variables. Examples:
- Mixture of 2 Gaussians: Bayes net $Z \rightarrow X$ with factorization $p_{Z,X}(z,x) = p_Z(z)p_{X|Z}(x|z)$ and,
  - Z ~ Bernoulli(p), i.e. z is binary
  - $X|Z = 0 \sim N(\mu_0, \sigma_0), X|Z = 1 \sim N(\mu_1, \sigma_1)$
  - The parameters are $p, \mu_0, \sigma_0, \mu_1, \sigma_1$
- Bayes net $Z \rightarrow X$ with factorization $p(z,x) = p(z)p(x|z)$
  - $Z \sim U(a,b)$
  - X | (Z = z) ~ N (z, σ)
  - The parameters are a, b, σ