

# 1 Probabilistic Graphical Models (Bayesian and Markov Networks)

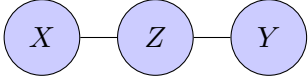
## 1.1 Data Dependency

Given a dataset whose random variables obviously break independence (IID) assumptions, how can we construct a likelihood function  $p(D | \theta)$ ? In the context of graphical models, we make use of *conditional independence*. First, if we have two independent random variables  $X$  and  $Y$  we can denote this as  $X \perp Y$  and we immediately know that we can factorize the joint likelihood as the product of the marginals, i.e.  $p(X, Y) = p(X)p(Y)$  and  $p(X | Y) = p(X)$ ,  $p(Y | X) = p(Y)$ . If, on the other hand,  $X$  and  $Y$  are conditionally independent given some other variable  $Z$  ( $X \perp Y | Z$ ), then we have instead that  $p(X, Y | Z) = p(X | Z)p(Y | Z)$ . We can also write the full joint distribution in terms of some other functions  $p(X, Y, Z) \propto g(X, Z) \cdot g(Y, Z)$ . In essence, conditional independence asserts that information can only be exchanged between  $X$  and  $Y$  *indirectly*, via the variable  $Z$ . In comparison, full independence asserts that no information can be exchanged between  $X$  and  $Y$  at all.

## 1.2 Graphs

### 1.2.1 Graphical Representation of Conditional Independence

A graph is a pair  $G = (V, E)$  of vertices (or nodes)  $V$  and edges  $E$ . For our purposes, random variables are represented by nodes in the graph and the presence of an edge between two nodes indicates direct information exchange. For example,

for  $X \perp Y | Z$  we have .

### 1.2.2 Graphical Probability Distribution Factorization

Given a graph  $G = (V, E)$  a probability distribution  $p(X)$  factorizes over  $G$  if  $p(X) \propto \prod_{c \in C} g_c(X^{(c)})$  where  $C$  is the set of all *cliques*<sup>1</sup> in  $G$  and  $g_c$  is the restriction of the function  $g$  to the subset  $X^{(c)}$ .

Actually, the graphical representations of conditional independence and factorization are equivalent [A.1](#).

## 1.3 Markov Network

A probability distribution  $p(X)$  whose conditional independence is represented as an undirected graph is called a *Markov Network*.

---

<sup>1</sup>Fully connected subgraphs.

### 1.3.1 Gaussian Markov Network

Consider  $\mathbf{x} \in \mathbb{R}^d$ ,  $\mathbf{x} \sim N(\mathbf{0}, \Theta^{-1})$  and

$$\begin{aligned} p(\mathbf{x}) &\propto \exp\left(-\sum_{u,v} \Theta^{(u,v)} x^{(u)} x^{(v)}\right) \\ &\propto \prod_{u,v; \Theta^{(u,v)} \neq 0} \exp(-\Theta^{(u,v)} x^{(u)} x^{(v)}) \\ &\propto \prod_{u,v; \Theta^{(u,v)} \neq 0} g_{u,v}(x^{(u)}, x^{(v)}) \end{aligned}$$

i.e. we see we can factorize  $p$  over  $G$ .  $G$  is defined by the adjacency matrix

$$A^{(u,v)} = \begin{cases} 0 & \Theta^{(u,v)} = 0 \\ 1 & \Theta^{(u,v)} \neq 0 \end{cases}$$

$G$  here is an undirected graph since we are not making any claims about the directionality of the relationship between the variables, i.e. whether one specifically influences another, rather than the other way around. From the adjacency representation we can see that we can infer the sparsity of the precision matrix  $\Theta$  from  $G$  and vice versa. For example, given a graph  $G = (5, 3)$  we immediately know that we have 11 non-zero elements in  $\Theta$ . Similarly, given a  $\Theta$  we can construct a graph of the conditional independence, which means if we do not know the dependence structure a priori, we can estimate it by  $\hat{\Theta} = \operatorname{argmax}_{\Theta} \log p(D | \Theta)$ .

The Gaussian Markov Network is a member of a more general class of distributions, the *Exponential Family* of distributions defined by

$$p(\mathbf{x}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(\langle \boldsymbol{\theta}, \mathbf{f}(\mathbf{x}) \rangle)$$

where  $Z$  is the normalization factor and  $\mathbf{f}(\mathbf{x})$  is a feature transform of  $\mathbf{x}$ .

**Graphical Lasso** We can also regularize a Gaussian Markov Network by instead solving the problem

$$\begin{aligned} \hat{\Theta} &= \operatorname{argmax}_{\Theta} \log p(D | \Theta) - \lambda \|\Theta\|_1 \\ &= \operatorname{argmax}_{\Theta} -\operatorname{Tr}(\mathbf{S}\Theta) + \log \det \Theta - \lambda \|\Theta\|_1 \end{aligned}$$

where  $\mathbf{S}$  is the sample covariance and  $\|\Theta\| = \sum_{i,j} |\Theta^{(i,j)}|$  (A.2.3).

### 1.3.2 Conditional Markov Network

A conditional probability distribution  $p(Y | X)$  factorizes over a graph  $G(X \cup Y, E)$  if

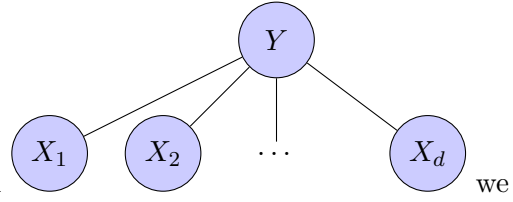
$$p(Y | X) = \frac{1}{N(X)} \prod_{c \in C} g_c(V_c)$$

where  $C = \{c \in C \mid V_c \not\subseteq X\}$  and  $N(X) = \int \prod_{c \in C} g_c(V_c) dY$  is the normalization constant.  $p(Y | X)$  does not include factors defined only on subsets of the

conditioning variable  $X$ .

$$\begin{aligned} p(Y | X) &= \frac{g_1(Y, X)g_2(X)}{\int g_1(Y, X)g_2(X)dY} \\ &= \frac{g_1(Y, X)g_2(X)}{g_2(X) \int g_1(Y, X)dY} \\ &= \frac{g_1(Y, X)}{\int g_1(Y, X)dY} \end{aligned}$$

### 1.3.3 Logistic Model



If we now take  $Y = \{1, -1\}$  and the graph see that we have the following conditional probability

$$\begin{aligned} p(Y | X) &= \frac{1}{N(X)} \prod_i g_i(Y, X_i) \\ N(X) &= \sum_{Y \in \{-1, 1\}} \prod_i g_i(Y, X_i) \end{aligned}$$

since the only edges are between  $Y$  and different dimensions of  $X$ . If we set  $g_i(Y = y, X_i = x^{(i)}; \beta^{(i)}, \beta_0) = \exp(y(\beta^{(i)}x^{(i)} + \beta_0))$  then we get

$$\begin{aligned} p(y | \mathbf{x}; \boldsymbol{\beta}, \beta_0) &= \frac{1}{N(\mathbf{x})} \prod_i \exp(y(\beta^{(i)}x^{(i)} + \beta_0)) \\ &= \frac{1}{N(\mathbf{x})} \exp(y(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0)) \\ N(\mathbf{x}) &= \sum_{y \in \{1, -1\}} \exp(y(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0)) \\ &= \exp(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0) + \exp(-\langle \boldsymbol{\beta}, \mathbf{x} \rangle - d\beta_0) \end{aligned}$$

which is equivalent to the logistic regression model we saw previously (A.3.1). We use this model by estimating the parameters  $\boldsymbol{\beta}, \beta_0$  by maximum likelihood, i.e.  $\hat{\boldsymbol{\beta}}, \hat{\beta}_0 = \operatorname{argmax}_{\boldsymbol{\beta}, \beta_0} \sum_{i=1}^n \log p(y_i | \mathbf{x}_i; \boldsymbol{\beta}, \beta_0)$ .

## 1.4 Bayesian Network

A Bayesian Network is a DAG that represents the factorization of some probability distribution  $p(X)$ .

### 1.4.1 Directed Acyclic Graph (DAG)

A DAG is a directed graphical model where the edges now have directions (unlike the Markov network). This encodes additional information regarding the nature of the relationship between nodes. Often, but not necessarily, the direction implies causality. Importantly, DAGs are acyclic, i.e. cannot contain any directed cycles or loops. In DAGs we have terms to describe the relationship between nodes:

$A \rightarrow B \implies A$  is the *parent* of  $B \Leftrightarrow B$  is the *child* of  $A$ .

$A \rightarrow \cdot \rightarrow B \implies B$  is the *descendant* of  $A$ .

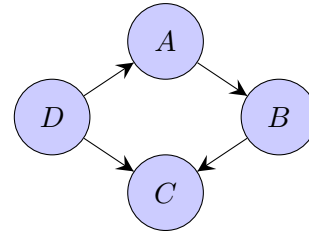
where we denote a *directed edge* as  $\rightarrow$  and a *directed path* as  $\rightarrow \cdot \rightarrow$ .

For example, we can describe the relationships of  $A$  in the following graph

**Parent**( $A$ ) :  $D$

**Children**( $A$ ) :  $B$

**Descendants**( $A$ ) :  $B, C$

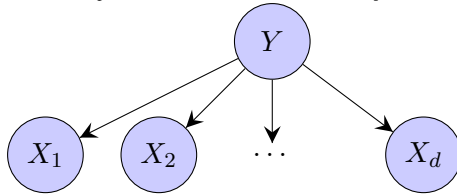


DAGs can represent the factorization of a probability distribution if

$$p(X) = \prod_{v \in V} p(X_v \mid X_{\text{parent}(X_v)})$$

DAGs can also represent conditional independence, since conditional independence over  $G$  is equivalent to factorization over  $G$ .

**Naïve Bayes** If we draw a Bayesian Network for classification, i.e.



and hence write down the conditional probability

$$p(Y \mid X) = \frac{\prod_i p(X^{(i)} \mid Y)p(Y)}{p(X)}$$

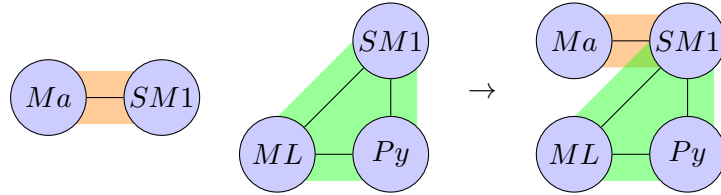
which describes a Naïve Bayes model.

# Appendices

## A Appendix

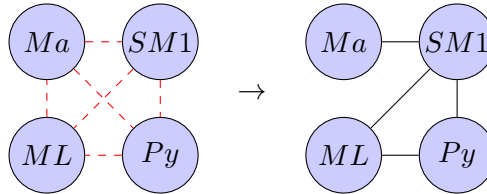
### A.1 Graphical Representations

Construct a graph  $G$  representing the factorization of the joint distribution  $p(Ma, SM1, Py, ML) \propto g_1(Ma, SM1) \cdot g_2(Py, ML, SM1)$ . Since we know that the functions  $g$  represent the fully connected subgraphs, we can start by constructing them and then combine:



Now for comparison we can derive the *same* graph from the known conditional relationships by constructing a fully connected graph and deleting edges that do not satisfy them:

$$\begin{aligned} Ma &\perp ML \mid SM1 \\ Ma &\perp Py \mid SM1 \\ Ma &\perp ML \mid SM1, Py \\ Ma &\perp Py, ML \mid SM1 \\ Ma &\perp Py \mid SM1, ML \end{aligned}$$



The graph on the left with dashed red lines indicates the “test” graph, whilst the right graph with solid black lines represents the actual graph with incorrect edges deleted.

### A.2 Gaussian Markov Network

#### A.2.1 Exponential Family

Given the definition of an exponential family distribution as  $p(\mathbf{x}; \boldsymbol{\theta}) = \frac{e^{\langle \boldsymbol{\theta}, \mathbf{f}(\mathbf{x}) \rangle}}{Z(\boldsymbol{\theta})}$  and a  $2^{nd}$  order polynomial transform  $\mathbf{f}(\mathbf{x})$ :

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= [1, \sqrt{2}\mathbf{x}^T, (\mathbf{x} \otimes \mathbf{x})^T]^T \\ p(\mathbf{x}; \boldsymbol{\theta}) &= \frac{1}{Z} \exp[\theta_1 + \sqrt{2}\boldsymbol{\theta}_2^T \mathbf{x} + \boldsymbol{\theta}_3^T \text{vec}(\mathbf{x}\mathbf{x}^T)] \end{aligned}$$

Where we have decomposed the long  $\boldsymbol{\theta}$  vector into 3 parts,  $\theta_1$  a scalar,  $\boldsymbol{\theta}_2 \in R^d$  and  $\boldsymbol{\theta}_3 \in R^{d^2}$ . If we now define an inverse to the  $\text{vec}(\cdot)$  operation,  $\text{vec}^{-1}(\cdot)$  that

reconstructs a matrix that would, when vectorised, return the original vector, and use that  $\text{Tr}(A^T B) = \text{vec}(A)^T \text{vec}(B)$ , then

$$\begin{aligned}\Theta &= \text{vec}^{-1}(\theta_3) \\ \text{vec}(\Theta)^T \text{vec}(\mathbf{x}\mathbf{x}^T) &= \text{Tr}(\Theta^T \mathbf{x}\mathbf{x}^T) = \mathbf{x}^T \Theta \mathbf{x}\end{aligned}$$

where we have set the matrix  $\Theta$  to be symmetric without loss of generality (since  $\mathbf{x}\mathbf{x}^T$  is symmetric), and so we get

$$p(\mathbf{x}; \theta) = \frac{1}{Z} \exp[\theta_1 + \sqrt{2}\theta_2^T \mathbf{x} + \mathbf{x}^T \Theta \mathbf{x}]$$

If we now set the entries of  $\Theta$  to be  $\leq 0$  to get a negative semi-definite matrix, then we see we have a Gaussian with covariance  $-\Theta^{-1}$  and mean  $\frac{1}{\sqrt{2}}\Theta^{-1}\theta_2$ .

### A.2.2 Likelihood

If  $(x_0, \mathbf{x})$  are drawn from a joint Gaussian  $p(x_0, \mathbf{x})$  then for the likelihood, we have

$$p(x_0 | \mathbf{x}) = N_{x_0}(\mu_0 - \Theta_{00}\Theta_{0x}^T(\mathbf{x} - \boldsymbol{\mu}_x), \Theta_{00}^{-1})$$

which we get from 6.66 and 6.67 in [1]. Now if we set  $\mu_0 = 0$ ,  $\boldsymbol{\mu}_x = \mathbf{0}$  and  $\Theta_{00} = b$  then we get for the log-likelihood

$$\begin{aligned}\log p(x_0 | \mathbf{x}) &= -\frac{1}{b}(x_0 + \frac{1}{b}\Theta_{0x}^T \mathbf{x})^2 + \text{const.} \\ &= -\frac{1}{b}(x_0 - \boldsymbol{\beta}^T \mathbf{x})^2 + \text{const.}\end{aligned}$$

This is useful for feature selection, since we know that the sparseness of  $\Theta_{0x}$  tells us which variables are conditionally independent, i.e. any zeros indicate features we can ignore.

### A.2.3 Graphical Lasso

Given a Gaussian Markov Network with

$$\begin{aligned}p(D | \Theta) &= \prod_{i=1}^n N_{\mathbf{x}_i}(\mathbf{0}, \Theta^{-1}) = \prod_{i=1}^n \frac{1}{Z} e^{-\mathbf{x}_i^T \Theta \mathbf{x}_i} \\ \log p(D | \Theta) &= -\sum_{i=1}^n (\log Z + \mathbf{x}_i^T \Theta \mathbf{x}_i) \\ &= -\sum_{i=1}^n \log Z - \text{Tr} \left( \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T \Theta \right) \\ &= -n \log \det \Theta - \text{Tr}(\mathbf{S}\Theta) + \text{const.}\end{aligned}$$

## A.3 Conditional Markov Network

### A.3.1 Logistic Model

From the model definition

$$\begin{aligned}p(y | \mathbf{x}; \boldsymbol{\beta}, \beta_0) &= \frac{1}{N(\mathbf{x})} e^{y\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0} \\ N(\mathbf{x}) &= e^{\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0} + e^{-\langle \boldsymbol{\beta}, \mathbf{x} \rangle - d\beta_0}\end{aligned}$$

we can show that this is equivalent to the logistic regression formulation we saw before by multiplying throughout by  $e^{y\langle\beta, \mathbf{x}\rangle + d\beta_0}$  for a particular value of  $y = \{1, -1\}$ . Defining the exponent as  $t$  to simplify the expressions, we get:

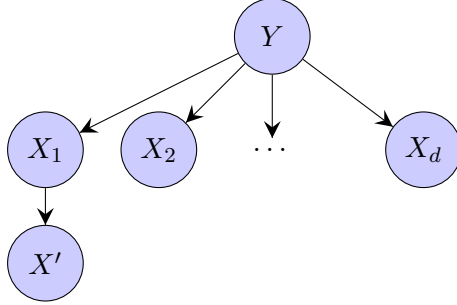
$$\begin{aligned} f(t) &= \frac{e^{\pm t}}{e^t + e^{-t}} \\ &= \frac{1}{1 + e^{\mp 2t}} \\ \implies p(\pm 1 \mid \mathbf{x}; \beta, \beta_0) &= \frac{1}{1 + e^{\mp 2\langle\beta, \mathbf{x}\rangle \mp d\beta_0}} \end{aligned}$$

which we can see has the same form as the logistic regression model, with a scaling factor of 2 which we can simply absorb into the coefficients.

## A.4 Bayesian Network

### A.4.1 Naïve Bayes

Given a Bayesian Network



for a classification task, we would not include the feature  $X'$  since it's contribution would cancel in the the numerator and denominator of the conditional distribution, since it is not a function of  $Y$ :

$$\begin{aligned} p(Y \mid X) &= \frac{p(X' \mid X^{(1)}) \prod_{i=1} p(X^{(i)} \mid Y)p(Y)}{p(X)} \\ &= \frac{p(X' \mid X^{(1)}) \prod_{i=1} p(X^{(i)} \mid Y)p(Y)}{\int p(X, Y) dY} \\ &= \frac{p(X' \mid X^{(1)}) \prod_{i=1} p(X^{(i)} \mid Y)p(Y)}{p(X' \mid X^{(1)}) \int \prod_{i=1} p(X^{(i)} \mid Y)p(Y) dY} \\ &= \frac{\prod_{i=1} p(X^{(i)} \mid Y)p(Y)}{\int \prod_{i=1} p(X^{(i)} \mid Y)p(Y) dY} \end{aligned}$$

which is the same result we get if we had ignored  $X'$  from the start.

## References

- [1] Christopher M Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, 2006. Softcover published in 2016.