1 Conditional Independence

We often assume independence of observations for ease of calculations. For independent variables X and Y, we denote $X \perp Y$ and we have the properties

$$p(X,Y) = p(X)p(Y) \iff p(X|Y) = p(X) \iff p(Y|X) = p(Y) \tag{1}$$

meaning we have no information exchange between X and Y.

Similarly, we can factorise probabilities for conditionally independent variables. For variables X and Y independent given Z, we write $X \perp Y|Z$ and

$$p(X,Y|Z) = p(X|Z)p(Y|Z) \iff p(X|Y,Z) = p(X|Z) \iff p(Y|X,Z) = p(Y|Z) \tag{2}$$

meaning any information exchanged between X and Y happens through Z. We can also factorise as follows,

$$p(X,Y,Z) \propto g_1(X,Z) \cdot g_2(Y,Z) \tag{3}$$

where g_1 and g_2 are functions that we will define later.

2 Undirected Graphical Models

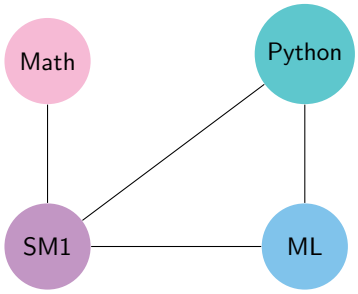
Often, with larger, more complex models, there will be many of these types of relationships, which is tedious to write out. Further it's often more difficult to visualise the relations between variables, hence we turn to graphical representation, where nodes represent random variables and edges represent dependence.

2.1 Encoding Dependencies

Given a list of conditional indepences, we can construct a graph.

Example Suppose we consider students taking four modules: Math, Machine Learning (ML), Statistical Methods 1 (SM1) and Python. We know the following conditional independences hold:

- Math $\perp$ ML | SM1
- Math $\perp$ Python | SM1
- Math $\perp$ ML | SM1, Python
- Math $\perp$ Python, ML | SM1
- Math $\perp$ Python | SM1, ML



and we want to construct a graph from these. We know Math is independent from ML and Python conditional on SM1 and so there is no edge between Math and these subjects. There is no conditional independence between Math and SM1 or any combination of subjects not including Math, so we know the graph which encodes these dependencies is the one on the left of this page.

2.2 Factorisation Over Graphs

We might also consider how a probability distribution relates to a graph. We can see from equations (2) and (3) that the joint probability of large models will factorise to contain many factors, which can be hard to interpret.

Given a graph $G = (V, E)$, we say a distribution $p(X)$ factorises over G if

$$p(X) \propto \prod_{c \in C} g_c(X^{(c)}) \tag{4}$$

where C is the set of all the largest possible connected subgraphs (called cliques) of G. We can illustrate this by returning to the example presented above.

Example Suppose we have the same modules as above, but instead of conditional independences we are given the following factorisation:

$$p(\text{Math, SM1, Python, ML}) \propto g_1(\text{Math, SM1}) \cdot g_2(\text{Python, ML, SM1}) \tag{5}$$

Then we would expect the corresponding graph to have two cliques: one containing Math and SM1, the other containing Python, ML and SM1. However, we can see this is the exact graph produced by the list of conditional independencies. Hence we can see p as defined in (5) satisfies all the conditional independences listed in the previous example. Further the p which satisfies these independencies (and thus produce the graph shown) must also create the factorisation given in (5)

Returning to the general case, this equivalency holds for all graphs G

$$\begin{gathered} p \text{ factorises over } G \\ \iff \\ p \text{ satisfies all conditional independence represented by G} \end{gathered} \tag{6}$$

So far, the graphs we have been considering are undirected graphical models, also called Markov networks. We now look at the special case, where p is Gaussian.

2.3 Gaussian Markov Networks

Suppose $\mathbf{x} \in \mathbb{R}^d$ follows a multivariate Gaussian distribution, $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) = \mathcal{N}(\mathbf{0}, \mathbf{\Theta})$. Then

$$\begin{aligned} p(\mathbf{x}) &\propto \exp\left[-\frac{\mathbf{x}\mathbf{\Sigma}^{-1}\mathbf{x}^T}{2}\right] \propto \exp\left[-\frac{\sum_{u,v}\mathbf{\Theta}^{(u,v)}x^{(u)}x^{(v)}}{2}\right] \\ &\propto \prod_{u,v;\mathbf{\Theta}^{(u,v)}\neq 0} \exp\left(-\mathbf{\Theta}^{(u,v)}x^{(u)}x^{(u)}\right) \end{aligned} \tag{7}$$

and so if we define $g_{u,v}\left(x^{(u)}, x^{(v)}\right) = \exp\left(-\mathbf{\Theta}^{(u,v)}x^{(u)}x^{(u)}\right)$, we can see $p(\mathbf{x})$ factorises over G. Further the matrix indicating the non-zero entries of $\mathbf{\Theta}$ defines the adjacency matrix of G. We can see this would only be valid in undirected graphs due to the symmetry of $\mathbf{\Theta}$. Hence we can infer the sparsity of $\mathbf{\Theta}$ given a graph G.

Example Suppose a graph G encodes all dependencies for a Gaussian distribution and contains 5 nodes and 3 edges. Then the non-zero elements would be the 5 on the diagonal and the six entries corresponding to the edges (since it is undirected, edges on the graph correspond to 2 edges. Hence there are 11 non-zero elements in our matrix $\mathbf{\Theta}$.

2.3.1 Estimating Sparsity

We can use this property to infer dependencies from data using maximum likelihood methods, using a graphical lasso as this forces small entries to 0:

$$\begin{aligned} \hat{\mathbf{\Theta}} &= \arg\max_{\mathbf{\Theta}} \log p(D; \mathbf{\Theta}) - \lambda||\mathbf{\Theta}||_1 \\ &= \arg\max_{\mathbf{\Theta}} -\text{t}r(\mathbf{S}\mathbf{\Theta}) + \log\det\mathbf{\Theta} - \lambda||\mathbf{\Theta}||_1 \end{aligned} \tag{8}$$

where $\mathbf{S}$ is the sample covariance.

2.4 Conditional Markov Network

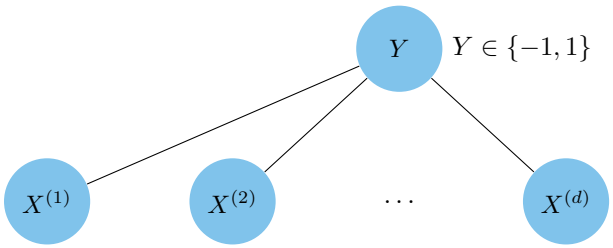
In many ML scenarios, we want to find a conditional distribution. We saw $p(X|Y)$ factorises over a G such that $V = X \cup Y$ if

$$p(Y|X) = \frac{1}{N(X)} \prod_{c \in C} g_c(V_c) \tag{9}$$

where $C \coloneqq \{c \text{ is a clique in } G \mid V_c \not\subseteq X\}$ and $N(X) \coloneqq \int \prod_{c \in C} g_c(V_c) dY$ is the normalising constant.

2.5 Logistic Regression

We now consider a standard binary classification task under this framework. The Markov network for this would look like:



We can use the factorisation rule (9) defined in Section 2.4 to obtain

$$p(Y|X) = \frac{1}{N(X)} \prod_i g_i(Y, X^{(i)}); \qquad N(X) = \sum_{Y \in \{-1,1\}} \prod_i g_i(Y, X^{(i)}) \tag{10}$$

If we set $g_i(Y = y, X^{(i)} = x^{(i)}; \beta_i \beta_0 \coloneqq \exp\left(y \cdot \left(\beta_i \cdot x^{(i)} + \beta_0\right)\right)$ we get the likelihood of the logistic model as

$$\begin{aligned} p(y|\mathbf{x}; \boldsymbol{\beta}, \beta_0) &= \frac{1}{N(X)} \prod_i \exp\left(y \cdot \left(\beta_i \cdot x^{(i)} + \beta_0\right)\right) \\ &= \frac{1}{N(X)} \exp\left(y \cdot \left(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0\right)\right) \\ N(X; \boldsymbol{\beta}, \beta_0) &= \sum_{y \in \{-1,1\}} \exp\left(y \cdot \left(\langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0\right)\right) \end{aligned} \tag{11}$$

and so if we define $f'(\mathbf{x}; \boldsymbol{\beta}, \beta_0) = \langle \boldsymbol{\beta}, \mathbf{x} \rangle + d\beta_0$ the likelihood becomes:

$$\begin{aligned} p(y|\mathbf{x}; \mathbf{w}) &= \frac{\exp(y \cdot f'(\mathbf{x}; \boldsymbol{\beta}, \beta_0))}{\exp(f'(\mathbf{x}; \boldsymbol{\beta}, \beta_0)) + \exp(-f'(\mathbf{x}; \boldsymbol{\beta}, \beta_0))} \\ &= \frac{1}{1 + \frac{\exp(-y \cdot f'(\mathbf{x}; \boldsymbol{\beta}, \beta_0))}{\exp(y \cdot f'(\mathbf{x}; \boldsymbol{\beta}, \beta_0))}} = \frac{1}{1 + \exp(-2y \cdot f'(\mathbf{x}; \boldsymbol{\beta}, \beta_0))} \\ &= \frac{1}{1 + \exp(-y \cdot f(\mathbf{x}; \mathbf{w}))} \end{aligned}$$

where $\mathbf{w} = [\mathbf{w}', w_0]^T = [2\boldsymbol{\beta}, 2\beta_0]^T$ and $f(\mathbf{x}; \mathbf{w}) = \langle \mathbf{x}, \mathbf{w} \rangle + w_0$ which is exactly the same likelihood expression for logistic regression.

3 Directed Graphical Models

Some dependencies are better encoded by directed edges (e.g. causal relations), and so we use directed acyclic graphs (DAGs) to illustrate these. If there is an edge $A \to B$ we say A is a parent of B and B is a child of A. Further if there is a directed path from A to B, we say B is a decendent of A.

3.1 Factorisation and Dependencies

We say a probability distribution $p(X)$ factorises over a DAG G if

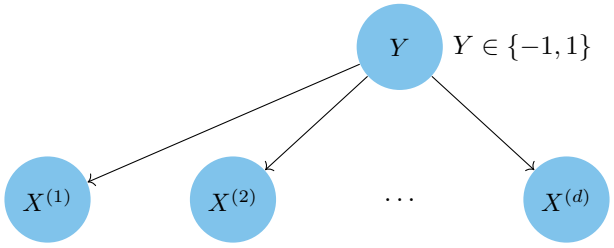
$$p(X) = \prod_{v \in V} p\left(X_v | X_{parent(X_v)}\right) \tag{12}$$

We further know X_v is independent of $X_{non-descendent(X_v)}$ given $X_{parent(X_v)}$. We can relate this to Markov networks, as here we acheive conditional independence given a set of nodes (namely the parent nodes).

The equivalence of factorisation and dependencies in distributions stated in (6) also holds in the case of directed graphs. We call a distribution $p(X)$ that factorises over a DAG a Bayesian network.

3.2 Naive Bayes

We again consider a binary classification task. The directed graph for this would be:



Using the factorisation rule (12) and Bayes rule, we get $p(X|Y)$ as

$$p(X|Y) = \frac{\prod_i p(X^{(i)}|Y)p(Y)}{p(X)} \propto \prod_i p(X^{(i)}|Y)p(Y) \tag{13}$$

and we can then use the naive Bayes prediction

$$\hat{y} \coloneqq \arg\max_y \prod_i p(X^{(i)}|y)p(y) \tag{14}$$

4 Two Approaches to Classification

We have now seen two graphs to represent classification, with the same structure. We might consider how they lead to different approaches to clasification.

The undirected graph relied on factorisation over cliques whereas the DAG used a factorisation of conditionally independent probabilities.

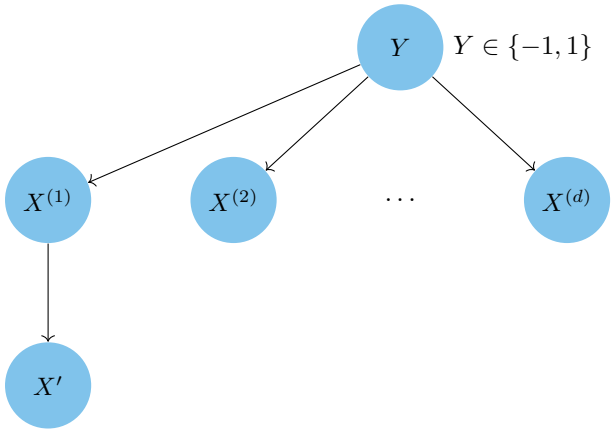
Both graphs use $p(Y|X)$ to make predictions, however the naive Bayes framework does not fully specify $p(Y|X)$.

When training a classifier, the logistic regression estimates $p(Y|X)$ directly, whereas the naive Bayes estimates $p(X|Y)$ to make a decision.

Despite the classifier being different, they both use the prediction rule $\hat{y} \coloneqq \arg\max_y p(Y|X)$

A Appendix

Suppose we have the graph



We might want to know whether to include X' in our decision. We can see that Y is not a descendent of X' and thus $X' \perp Y | X'_{parent}$. Hence, as long as the parent of X' ($X^{(1)}$) is included in the model, X' will not contribute any additional knowledge to the distribution of Y.