C19 Unsupervised Machine Learning

Lecture notes written by Tuan Anh Le* for a course by

Frank Wood†
Engineering Department
University of Oxford, UK

February 21, 2017

Contents

*tuananh@robots.ox.ac.uk
†fwood@robots.ox.ac.uk

1 Introduction

These notes are heavily based on a chapter on unsupervised learning by Ghahramani (2004) and the "Pattern recognition and machine learning" book by Bishop (2006). The parts on sampling and Markov Chain Monte Carlo are based on a review by Neal (1993).

Machine learning is usually divided into supervised, unsupervised and reinforcement learning. In supervised learning, our data consists of (x, y) pairs, where x is some input and y is the corresponding label. For example, x can be an image of a cat and y can be the label "cat." These are used to train our model (e.g. regression) so that during test time, when we can accurately predict a label for a new input. This approach works very well but requires a lot of labelled inputs. Unsupervised learning is all about getting insights from data when we don't have labels.

In order to do this, we have to either explicitly or implicitly place a model on the structure of inputs and labels. Bayesian modelling is a way to coherently do this. It is coherent in the sense that everything we do with our data follows the rules of probability theory which in turn corresponds well with how we update beliefs about the world (see Cox Axioms or the Dutch Book Theorem).

1.1 Basic Probability

Random variables and probability distributions are the basic building blocks of probabilistic models. In this course, we will restrict ourselves to *continuous* random variables whose support is a subset of $\mathbb{R}^d$ and *discrete* random variables whose support is a subset of countable set (of symbols or numbers or whatever).

Notation. We will use P and p to denote probability mass functions of discrete random variables and probability densities of continuous random variables respectively.[1] We will denote random variables with a capital letter (e.g. X) and their values with the corresponding small letter (e.g. x). We will write $P(x|y)$ instead of $P(x|Y = y)$ to denote the conditional density/probability of the variable X given $\{Y = y\}$. Vectors in $\mathbb{R}^d, d > 1$ will be bolded, e.g. $\mathbf{x}$. A set of variables $x_i, x_{i+1}, \ldots, x_j$ (with $i \leq j$) is sometimes denoted as $x_{i:j}$. We use $\sim$ to mean "is sampled from" or "follows the distribution"; e.g. $x \sim \mathrm{Normal}(\mu, \sigma^2)$ means x is sampled from a normal distribution with mean μ and variance σ^2 and $X \sim \mathrm{Normal}(\mu, \sigma^2)$ means that X is a random variable distributed according to the normal distribution with mean μ and variance σ^2. Densities or probability mass functions of known probability distributions are functions of the same name, with the distribution parameters on the right of the conditioning bar, e.g. $\mathrm{Normal}(x|\mu, \sigma^2)$ or $\mathrm{Categorical}(x|\boldsymbol{\pi})$. For a summary of commonly used probability distributions, see Appendix 1.

Discrete random variables must follow

$$P(x) \geq 0, \quad \forall x \qquad \text{and} \qquad \sum_x P(x) = 1. \tag{1}$$

Continuous random variables must follow

$$p(x) \geq 0, \quad \forall x \qquad \text{and} \qquad \int p(x)\,\mathrm{d}x = 1. \tag{2}$$

All manipulations of probabilistic models in this course boil down to using the following identities (and their corresponding versions for continuous random variables):

$$P(y) = \sum_{x'} P(x', y) \qquad \text{(sum rule)} \tag{3}$$

$$P(x, y) = P(y|x)P(x). \qquad \text{(product rule)} \tag{4}$$

Bayes rule arises from the combination of the sum and product rules:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} = \frac{P(y|x)P(x)}{\sum_{x'} P(y|x')P(x')}. \tag{5}$$

2 Graphical Models

Graphical models refer to a family of frameworks for representing probabilistic models using *graphs*. Doing this is useful for two reasons:

- We can visualise statistical dependencies between random variables in the model.
- Inference and learning in the probabilistic model can be formulated in terms of operations on a graph.

[1] We will use p for mixtures of continuous and discrete random variables. This should hopefully be clear from the context.

2.1 Directed Graphical Models

Consider a probabilistic model of random variables $x_1, x_2, \ldots, x_N$. Applying the product rule repeatedly, we can write down the *joint distribution* (or just "joint") of these random variables as follows:

$$P(x_1, \ldots, x_N) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \cdots P(x_N|x_1, \ldots, x_{N-1}). \tag{6}$$

This is a full factorisation of the joint. A full factorisation is usually an over-representation of the actual joint: what if $P(x_3|x_1, x_2) = P(x_3|x_2)$, i.e. X_1 and X_3 are conditionally independent given X_2 (write $X_1 \perp\!\!\!\perp X_3|X_2$)? A factorisation which incorporates this information is more desirable.

Consider the directed graphical model in Figure 1a which represents the following factorisation of the joint distribution of $P(x_{1:5})$:

$$P(x_{1:5}) = P(x_1)P(x_2)P(x_3|x_1, x_2)P(x_4|x_2, x_3)P(x_5|x_3, x_4). \tag{7}$$

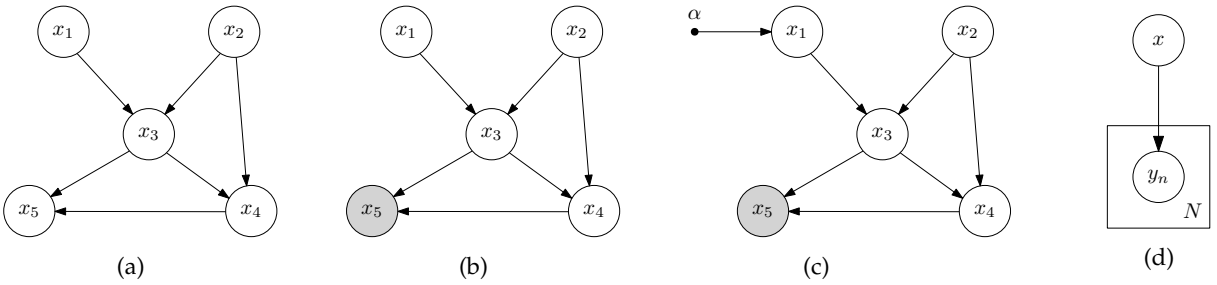

Figure 1: Examples of directed graphical models (DGM). (a) Unconditioned DGM, (b) DGM with observed random variable x_5, (c) DGM with an observed random variable x_5 and a fixed, deterministic hyperparameter α, and (d) Example of the plate notation: y_n is repeated for $n \in \{1, \ldots, N\}$.

In general, given a directed acyclic graph consisting of vertices $V := \{1, \ldots, N\}$ and directed edges E, each vertex n represents a random variable X_n and the joint probability of $X_1, \ldots, X_N$ is

$$P(x_1, \ldots, x_N) = \prod_{n=1}^{N} P(x_n|x_{\mathrm{pa}(n)}), \tag{8}$$

where $\mathrm{pa}(n) := \{m : (m, n) \in E\}$ denotes the set of parents of vertex n, i.e. the set of all vertices pointing to the vertex n. A probabilistic model is defined by specifying each conditional probability term $P(x_n|x_{\mathrm{pa}(n)})$.

Observed variables We denote that a variable is observed by shading the corresponding vertex in the graph. In Figure 1b, x_5 is the observed variable. It means that in this model of $P(x_{1:5})$, we have data for x_5 and are interested in $P(x_{1:4}|x_5)$, or more generally any subset of $x_{1:4}$ given x_5.

Hyperparameters. We denote that a variable is a deterministic one by a small filled circle. This variable is usually called a *hyperparameter* of the model and is usually written on the right of the conditioning bar after a semicolon. In Figure 1c, α is the hyperparameter of a model whose joint is written $P(x_{1:5}|\alpha)$. Since x_5 is shaded, we are interested in $P(x_{1:4}|x_5; \alpha)$.

Plate notation. If an indexed variable is repeated many times, we use can illustrate it by including it in a rectangle with the end index in the bottom-left corner. In Figure 1d, the graph represents a model of $P(x, y_{1:N})$ with the factorisation $P(x, y_{1:N}) = P(x) \prod_{n=1}^{N} P(y_n|x)$.

2.1.1 Conditional independence

Directed graphical models allow us to read off conditional independence relationships between variables in the model using a graph-based algorithm. We say "a is conditionally independent of b given c" if $p(a, b|c) = p(a|c)p(b|c)$ and denote this by $a \perp\!\!\!\perp b|c$. Independence is just a special case in which we don't condition on anything. We say "a is independent of b" if $p(a, b) = p(a)p(b)$ and denote this by $a \perp\!\!\!\perp b$.

Before introducing the D-separation algorithm for detecting conditional independencies in a general graph, let's consider the following simple cases for three-variable models.

Tail-to-tail. Consider the model in Figure 2a and 2e which has the factorisation $p(a,b,c) = p(c)p(a|c)p(b|c)$. $p(a,b) = \sum_c p(a,b,c) = \sum_c p(c)p(a|c)p(b|c)$ which in general doesn't equal $p(a)p(b)$ (Try it yourself on a toy model!). Hence $a \not\perp\!\!\!\perp b$.

However, if we condition on c, $p(a,b|c) = p(a,b,c)/p(c) = p(c)p(a|c)p(b|c)/p(c) = p(a|c)p(b|c)$. Hence $a \perp\!\!\!\perp b|c$.

We can think of observing c as blocking the path from a to b: when it's observed a and b are independent, when it's not, they are dependent.

Head-to-tail. Consider the model in Figure 2c and 2g which has the factorisation $p(a,b,c) = p(a)p(c|a)p(b|c)$. $p(a,b) = \sum_c p(a,b,c) = \sum_c p(a)p(c|a)p(b|c) = p(a)\sum_c p(c|a)p(b|c)$ which in general doesn't equal $p(a)p(b)$ (Try it!). Hence $a \not\perp\!\!\!\perp b$.

However, if we condition on c, $p(a,b|c) = p(a,b,c)/p(c) = p(a)p(c|a)p(b|c)/p(c) = p(a|c)p(b|c)$. Hence $a \perp\!\!\!\perp b|c$.

Similar to the tail-to-tail case, we can think of observing c as blocking the path from a to b.

Head-to-head. Consider the model in Figure 2c and 2g which has the factorisation $p(a,b,c) = p(a)p(b)p(c|a,b)$. $p(a,b) = \sum_c p(a,b,c) = \sum_c p(a)p(b)p(c|a,b) = p(a)p(b)\sum_c p(c|a,b) = p(a)p(b)$. Hence $a \perp\!\!\!\perp b$.

However, if we condition on c, $p(a,b|c) = p(a,b,c)/p(c) = p(a)p(b)p(c|a,b)/p(c)$ which in general doesn't equal $p(a|c)p(b|c)$ (Try it!). Hence $a \not\perp\!\!\!\perp b|c$.

Unlike the previous two cases, observing c unblocks the path: a and b are *independent* before conditioning on c but conditioning on it makes them *conditionally dependent*. This phenomenon is called *explaining away*.

2.1.2 D-separation

Consider a general directed graph in which A, B and C are arbitrary nonintersecting sets of nodes (whose union may be smaller than the complete set of nodes in the graph). We wish to ascertain whether a particular conditional independence statement $A \perp\!\!\!\perp B|C$ is implied by a given directed acyclic graph. To do so, we consider all possible paths from any node in A to any node in B. Any such path is said to be *blocked* if it includes a node such that either

- the arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set C, or
- the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in the set C.

If all paths are blocked, then A is said to be d-separated from B by C, and the joint distribution over all of the variables in the graph will satisfy $A \perp\!\!\!\perp B|C$.

The concept of d-separation is illustrated in Figure 2d and 2h. In Figure 2d, the path from a to b is not blocked by node f because it is a tail-to-tail node for this path and is not observed, nor is it blocked by node e because, although the latter is a head-to-head node, it has a descendant c in the conditioning set. Thus the conditional independence statement $a \perp\!\!\!\perp b|c$ does *not* follow from this graph. In Figure 2h, the path from a to b is blocked by node f because this is a tail-to-tail node that is observed, and so the conditional independence property $a \perp\!\!\!\perp b|f$ will be satisfied by any distribution that factorises according to this graph. Note that this path is also blocked by node e because e is a head-to-head node and neither it nor its descendant are in the conditioning set.

It should be noted that this section is more-or-less copied from (Bishop, 2006, 8.2.2).

2.2 Bayesian Linear Regression

Bayesian linear regression starts by modelling the joint over both the data and the parameters. The graphical model is shown in Figure 3.

The joint distribution corresponding to this graphical model factorises as follows:

$$p(y_{1:N}, \mathbf{w}|\mathbf{x}_{1:N}, \alpha, \sigma^2) = p(\mathbf{w}|\alpha)p(y_{1:N}|\mathbf{w}; \mathbf{x}_{1:N}, \sigma^2) \tag{9}$$

$$= p(\mathbf{w}|\alpha)\prod_{n=1}^{N} p(y_n|\mathbf{w}; \mathbf{x}_n, \sigma^2) \tag{10}$$

$$= \text{Normal}(\mathbf{w}|\mathbf{0}, \alpha\mathbf{I})\prod_{n=1}^{N} \text{Normal}(y_n|\mathbf{w}^T\mathbf{x}_n, \sigma^2). \tag{11}$$

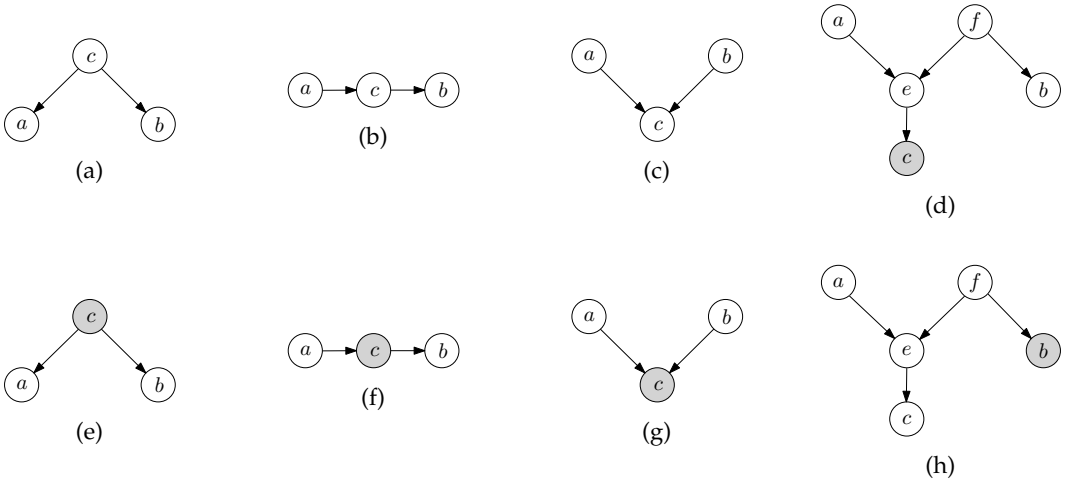

Figure 2: Simple directed graphical models to investigate independencies $a \perp\!\!\!\perp b$ and conditional independencies $a \perp\!\!\!\perp b|c$. (a, d) c is a tail-to-tail node, (b, e) c is a head-to-tail node, (c, f) c is a tail-to-tail node, (d, h) more complicated example illustrating the concept of d-separation. See text for details.

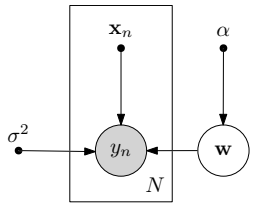

Figure 3: Graphical model for Bayesian Linear Regression.

By modelling the full joint, we can answer many interesting questions such as what is the posterior of $\mathbf{w}$ given the data? Using the sum and product rules and omitting hyperparameters to the right of the conditioning bar:

$$p(\mathbf{w}|y_{1:N}; \mathbf{x}_{1:N}) = \frac{p(\mathbf{w}) \prod_{n=1}^{N} p(y_n|\mathbf{w}; \mathbf{x}_n)}{\int p(\mathbf{w}) \prod_{n=1}^{N} p(y_n|\mathbf{w}; \mathbf{x}_n) \, \mathrm{d}\mathbf{w}} \tag{12}$$

$$\propto p(\mathbf{w}) \prod_{n=1}^{N} p(y_n|\mathbf{w}; \mathbf{x}_n). \tag{13}$$

Another question we might ask is: given the data and a new test point $\hat{\mathbf{x}}$, what is the posterior predictive distribution of $\hat{y}$? Again, we just use the sum and product rules:

$$p(\hat{y}|\hat{\mathbf{x}}, y_{1:N}; \mathbf{x}_{1:N}) = \int p(\hat{y}, \mathbf{w}|\hat{\mathbf{x}}, y_{1:N}; \mathbf{x}_{1:N}) \, \mathrm{d}\mathbf{w} \tag{14}$$

$$= \int p(\hat{y}|\mathbf{w}, \hat{\mathbf{x}}) p(\mathbf{w}|y_{1:N}; \mathbf{x}_{1:N}) \, \mathrm{d}\mathbf{w}. \tag{15}$$

These are just some of the interesting questions that we might ask. We will see in Section 3, how to answer these questions.

2.3 Hidden Markov Model

Hidden Markov Model (HMM) forms an important building block for richer time-series analysis models and have played a historically important role in control engineering, visual tracking, speech recognition, protein sequence modelling, and error decoding. Given a sequence of inputs $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$, HMMs model correlations between them by introducing latent or hidden state variables $x_1, x_2, \ldots, x_T$.

In particular, the joint distribution for a Hidden Markov Model factorises, according to Figure 4, as follows:

$$p(x_{1:T}, \mathbf{y}_{1:T}|\theta) = \prod_{t=1}^{T} P(x_t|x_{t-1}; \theta) p(\mathbf{y}_t|x_t; \theta) \tag{16}$$

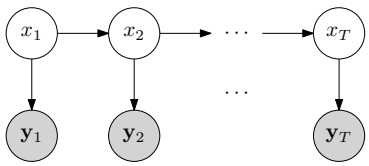

Figure 4: Graphical model the Hidden Markov Model.

Here, x_t denotes the hidden state of an HMM at time t. We assume that x_t can take discrete values in $\{1, \ldots, K\}$. $P(x_1|\theta)$ is simply some initial distribution over the K settings of the first hidden state. We call this distribution $\boldsymbol{\pi}$, represented by a $K \times 1$ vector. The state-transition probabilities $P(x_t|x_{t-1}; \theta)$ are captured by a $K \times K$ transition matrix A, with elements $A_{ij} := P(x_t = i|x_{t-1} = j; \theta)$. The observations in an HMM can be either continuous or discrete. For continuous observations $\mathbf{y}_t$, one can for example choose a Gaussian density; thus $p(\mathbf{y}_t|x_t = i; \theta)$ would be a different Gaussian for each choice if $i \in \{1, \ldots, K\}$. For discrete observations y_t, let us assume that it can take on values $\{1, \ldots, L\}$. In that case the output probabilities $P(y_t|x_t; \theta)$ can be captured by an $L \times K$ emission matrix E. The model parameters for a discrete-observation HMM are $\theta = (\boldsymbol{\pi}, A, E)$.

Questions we might want to answer are:

- What are the parameters that maximise the likelihood of observed data: $\arg\max_\theta p(\mathbf{y}_{1:T}|\theta)$?
- What is the distribution over the hidden states, given the observed states: $p(x_{1:T}|\mathbf{y}_{1:T}; \theta)$?
- What is the most likely sequence of hidden states, given the observed states: $\arg\max_{x_{1:T}} p(x_{1:T}|\mathbf{y}_{1:T}; \theta)$?
- What is the next point in the sequence: $p(\hat{\mathbf{y}}|\mathbf{y}_{1:T}; \theta)$?

2.4 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a probabilistic model used for modelling topics in a corpus of documents (documents is a set of words).

An annotated graphical model can be seen in Figure 5 below.

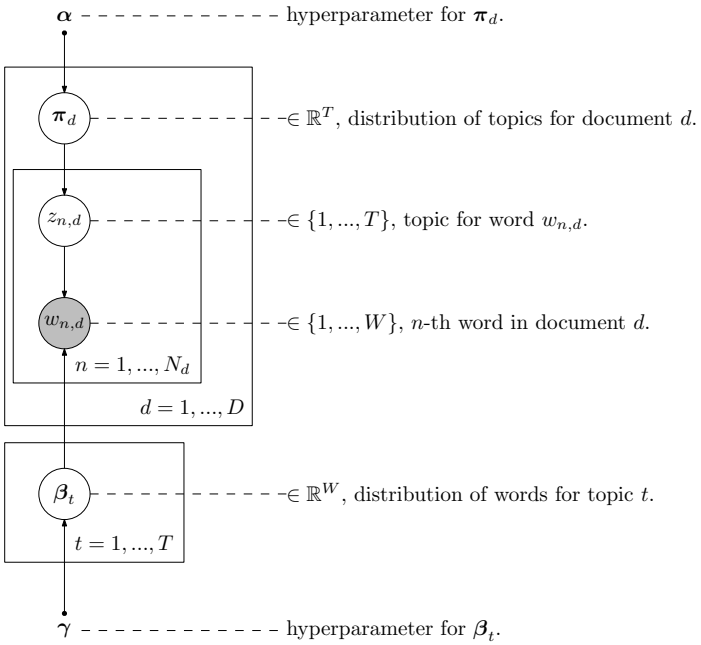

Figure 5: Graphical model for Latent Dirichlet allocation.

The joint probability is

$$p\left(\{\boldsymbol{\pi}_d\}, \{z_{n,d}\}, \{w_{n,d}\}, \{\beta_t\}|\boldsymbol{\alpha}, \boldsymbol{\gamma}\right) \tag{17}$$

$$= \left(\prod_d p(\boldsymbol{\pi}_d|\boldsymbol{\alpha})\right)\left(\prod_d \prod_n P(z_{n,d}|\{\boldsymbol{\pi}_\delta\})\right)\left(\prod_d \prod_n P(w_{n,d}|z_{n,d}, \{\boldsymbol{\beta}_t\})\right)\left(\prod_t p(\boldsymbol{\beta}_t|\boldsymbol{\gamma})\right) \tag{18}$$

where

$$p(\boldsymbol{\pi}_d|\boldsymbol{\alpha}) = \text{Dirichlet}(\boldsymbol{\pi}_d|\boldsymbol{\alpha}), \qquad d = 1, \ldots, D, \tag{19}$$

$$p(\boldsymbol{\beta}_t|\boldsymbol{\gamma}) = \text{Dirichlet}(\boldsymbol{\beta}_t|\boldsymbol{\gamma}), \qquad t = 1, \ldots, T, \tag{20}$$

$$P(z_{n,d}|\{\boldsymbol{\pi}_\delta\}) = \text{Categorical}(z_{n,d}|\boldsymbol{\pi}_d), \qquad n = 1, \ldots, N_d, d = 1, \ldots, D, \tag{21}$$

$$P(w_{n,d}|z_{n,d}, \{\boldsymbol{\beta}_t\}) = \text{Categorical}\left(w_{n,d}|\boldsymbol{\beta}_{z_{n,d}}\right), \qquad n = 1, \ldots, N_d, d = 1, \ldots, D. \tag{22}$$

Some of the questions we might want to answer are:

- How many topics are there in the corpus: $\arg\max_T P(\{w_{n,d}\}|\boldsymbol{\alpha}, \boldsymbol{\gamma}, T)$?
- What are the various topics of words: $P(\{z_{n,d}\}|\{w_{n,d}\}; \boldsymbol{\alpha}, \boldsymbol{\gamma})$?
- Which topics does this document belong to: $p(\boldsymbol{\pi}_d|\{w_{n,d}\}; \boldsymbol{\alpha}, \boldsymbol{\gamma})$?

2.5 Gaussian Mixture Model

Gaussian Mixture Model (GMM) is a probabilistic model used for clustering data points $\mathbf{x}_{1:N}$. For each data point $\mathbf{x}_n$, there is a corresponding latent (hidden) discrete random variable z_n taking values in $\{1, \ldots, K\}$ which represents the cluster this data point comes from.

The graphical model can be seen in Figure 6 below.

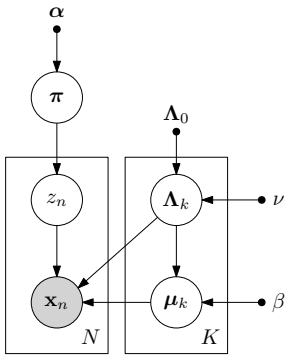

Figure 6: Graphical model for the Gaussian mixture model.

The joint probability is

$$p(\mathbf{x}_{1:N}, z_{1:N}, \boldsymbol{\Lambda}_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\pi}|\alpha, \boldsymbol{\Lambda}_0, \nu, \beta) = p(\boldsymbol{\pi}|\alpha)\left[\prod_{n=1}^{N} p(z_n|\boldsymbol{\pi})p(\mathbf{x}_n|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Lambda}_{z_n})\right]\left[\prod_{k=1}^{K} p(\boldsymbol{\Lambda}_k|\boldsymbol{\Lambda}_0, \nu)p(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k, \beta)\right] \tag{23}$$

where

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \text{Dirichlet}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \tag{24}$$

$$p(\boldsymbol{\Lambda}_k|\boldsymbol{\Lambda}_0, \nu) = \text{Wishart}(\boldsymbol{\Lambda}_k|\boldsymbol{\Lambda}_0, \nu) \tag{25}$$

$$p(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k, \beta) = \text{Normal}(\boldsymbol{\mu}_k|\mathbf{0}, (\beta\boldsymbol{\Lambda}_k)^{-1}) \tag{26}$$

$$P(z_n|\boldsymbol{\pi}) = \text{Categorical}(z_n|\boldsymbol{\pi}) \tag{27}$$

$$p(\mathbf{x}_n|z_n,\ \boldsymbol{\mu}_{1:K},\ \boldsymbol{\Lambda}_{1:K}) = \text{Normal}(\mathbf{x}_n|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Lambda}_{z_n}^{-1}). \tag{28}$$

Possible inference goals include: estimating candidate cluster centers and covariances; checking whether any two data points are in the same cluster; and estimating how many distinct clusters exist in the data.

2.6 Probabilistic Principal Component Analysis

Probabilistic Principal Component Analysis (Probabilistic PCA) is a probabilistic model used for dimensionality reduction. Given a set of high dimensional data points $\mathbf{y}_{1:N}$, we might want to know their representation in a lower-dimensional manifold. For each $\mathbf{y}_n$, we model its lower-dimensional representation $\mathbf{x}_n$ as a latent variable.

The graphical model for probabilistic PCA is in Figure 7 below.

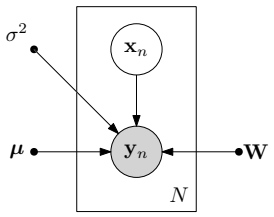

Figure 7: Graphical model for Probabilistic Principal Components Analysis.

The joint probability is

$$p(\mathbf{y}_{1:N}, \mathbf{x}_{1:N}|\mathbf{W}, \boldsymbol{\mu}, \sigma^2) = \prod_{n=1}^{N} p(\mathbf{x}_n)p(\mathbf{y}_n|\mathbf{x}_n; \mathbf{W}, \boldsymbol{\mu}, \sigma^2) \tag{29}$$

where

$$p(\mathbf{x}_n) = \text{Normal}(\mathbf{x}_n|\mathbf{0}, \mathbf{I}) \tag{30}$$

$$p(\mathbf{y}_n|\mathbf{x}_n; \mathbf{W}, \boldsymbol{\mu}, \sigma^2) = \text{Normal}(\mathbf{W}\mathbf{x}_n + \boldsymbol{\mu}, \sigma^2\mathbf{I}). \tag{31}$$

Possible inference goals might include

- Maximum likelihood learning of parameters: $\arg\max_{(\mathbf{W},\boldsymbol{\mu},\sigma^2)} p(\mathbf{y}_{1:N}|\mathbf{W}, \boldsymbol{\mu}, \sigma^2)$.
- Dimensionality reduction: $p(\mathbf{x}_{1:N}|\mathbf{y}_{1:N}; \mathbf{W}, \boldsymbol{\mu}, \sigma^2)$.

2.7 Undirected Graphical Models and Factor Graphs

Other graphical model that is commonly used is the undirected graphical models (or Markov fields). In this case, the edges don't correspond to the conditional probability distributions. Instead, they represent "compatibility functions." Conditional independence relationships can also be inspected from undirected graphical models although not using the D-separation algorithm. Factor graphs generalise directed and undirected graphical models. A factor graph consists of two types of nodes: factors and variables. An edge in a factor graph is undirected and is always between one factor node and one variable node. Each factor node corresponds to a factor (that needn't be a probability distribution) in the joint factorisation. This factor is a function of variables corresponding to the variable nodes connected to that factor node. We will not consider these, and other, graphical models in this course.

3 Inference, Learning, Monte Carlo Integration, Basic Sampling

In general, given a probabilistic model $p(x, y|\theta)$ of latents x, observed values y and hyperparameters θ, we might be interested in the following quantities.

Maximum likelihood/posterior learning of parameters θ: $\text{maximize}_\theta\, p(y|\theta)$ or $\text{maximize}_\theta\, p(y|\theta)p(\theta)$.

Note that in a fully Bayesian approach, we would place a prior over θ and treat it as another latent variable over which we would do posterior inference. However, this is not always possible. Maximum likelihood and maximum a-posteriori are point estimates of the posterior. This estimator is accurate in the limit of number of observed data going to infinity when the posterior becomes peaked on the maximum likelihood point.

These point estimates can be plugged in to the likelihood model to approximate the posterior predictive:

Posterior. $p(x|y;\theta) = \frac{p(x,y|\theta)}{\int p(x,y|\theta)\,\mathrm{d}x}$.

By inspecting the posterior over the latents, we can learn the hidden structure of our data under our model.

Marginal likelihood. $p(y|\theta) = \int p(x, y|\theta)\,\mathrm{d}x$.

The marginal likelihood (or the evidence) is often used for model selection. Assuming there are two models that we are considering, m_1 and m_2, we would like to know their probabilities given the data:

$$P(m_i|y) = \frac{p(y|m_i)P(m_i)}{\sum_j p(y|m_j)P(m_j)}. \tag{32}$$

Here, the denominator is independent of the choice of m_i and so we are only interested in the numerator, $p(y|m_i)P(m_i)$. We often use the marginal likelihood or evidence $p(y|m_i)$ to approximate this quantity. This is the basis of *Bayes factors* in which one looks at the ratio $p(y|m_1)/p(y|m_2)$ in order to choose a model.

Posterior expectations. $\mathbb{E}_{x\sim p(x|y;\theta)}[f(x)] = \int f(x)p(x|y;\theta)\,\mathrm{d}x$.

In many cases, we want to do something with our posterior. For example we might want to know the expected utility of an agent under the posterior distribution for some utility function f. Or we can design f to be some risk whose expected value we want to later minimize.

Posterior predictive. $p(\hat{y}|y;\theta) = \int p(\hat{y}|x;\theta)p(x|y;\theta)\,\mathrm{d}x$.

We would like to predict the next data point under our model while incorporating our posterior uncertainty about the latents given past data.

3.1 Inference by Analytic Integration: Dirichlet-Categorical Model

Here, we will consider a simple probabilistic model to model categorical data in which inferential questions can be answered by *analytic integration*. Consider a data set consisting of N data points $x_{1:N}$ which take on the values from $\{1,\dots,K\}$. Let our model consist of a latent variable $\boldsymbol{\theta}$ representing parameters of the Categorical distribution so that the likelihood is $P(x_n|\boldsymbol{\theta}) = \mathrm{Categorical}(x_n|\boldsymbol{\theta}) = \theta_{x_n}$. Let the prior for $\boldsymbol{\theta}$ follow the Dirichlet distribution with hyperparameters $\boldsymbol{\alpha} \in \mathbb{R}^K$. Hence, we can write the full joint of the latent variable and the data as follows:

$$p(\boldsymbol{\theta}, x_{1:N}) = \overbrace{p(\boldsymbol{\theta})}^{\text{prior}} \overbrace{\prod_{n=1}^{N} P(x_n|\boldsymbol{\theta})}^{\text{likelihood}} \tag{33}$$

$$= \mathrm{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{n=1}^{N} \mathrm{Categorical}(x_n|\boldsymbol{\theta}) \tag{34}$$

$$= \left(\frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1}\right) \prod_{n=1}^{N} \theta_{x_n}. \tag{35}$$

Posterior. We can derive the posterior as follows:

$$p(\boldsymbol{\theta}|x_{1:N}) = p(x_{1:N}, \boldsymbol{\theta})/p(x_{1:N}) \qquad \text{(Bayes' rule)} \tag{36}$$

$$\propto p(x_{1:N}, \boldsymbol{\theta}) \qquad \text{(the two quantities are proportional w.r.t. } \boldsymbol{\theta}) \tag{37}$$

$$= \left(\frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1}\right) \prod_{n=1}^{N} \theta_{x_n} \qquad \text{(substitute from (35))} \tag{38}$$

$$= \left(\frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1}\right) \prod_{k=1}^{K} \theta_k^{n_k} \qquad \text{(where we let } n_k := \sum_n \mathbb{1}(x_n = k)) \tag{39}$$

$$\propto \prod_k \theta_k^{n_k} \prod_k \theta_k^{\alpha_k - 1} \qquad \text{(the two quantities are proportional w.r.t. } \boldsymbol{\theta}) \tag{40}$$

$$= \prod_k \theta_k^{\alpha_k + n_k - 1} \tag{41}$$

$$\propto \mathrm{Dirichlet}\left(\boldsymbol{\theta}|\boldsymbol{\alpha} + (n_1,\dots,n_K)^T\right) \qquad \text{(the two quantities are proportional w.r.t. } \boldsymbol{\theta}). \tag{42}$$

We know that $p(\boldsymbol{\theta}|x_{1:N}) = \mathrm{Dirichlet}\left(\boldsymbol{\theta}|\boldsymbol{\alpha} + (n_1,\dots,n_K)^T\right)/Z$ for some proportionality constant Z independent of $\boldsymbol{\theta}$. Given the fact that $\int p(\boldsymbol{\theta}|x_{1:N})\,\mathrm{d}\boldsymbol{\theta} = \int \mathrm{Dirichlet}\left(\boldsymbol{\theta}|\boldsymbol{\alpha} + (n_1,\dots,n_K)^T\right)\,\mathrm{d}\boldsymbol{\theta} = 1$, we conclude that $Z = 1$ and $p(\boldsymbol{\theta}|x_{1:N}) = \mathrm{Dirichlet}\left(\boldsymbol{\theta}|\boldsymbol{\alpha} + (n_1,\dots,n_K)^T\right)$.

Effectively, we start with a $\mathrm{Dirichlet}(\boldsymbol{\alpha})$ prior belief for $\boldsymbol{\theta}$ and after observing data $x_{1:N}$, we update our beliefs about $\boldsymbol{\theta}$ to $\mathrm{Dirichlet}\left(\boldsymbol{\theta}|\boldsymbol{\alpha} + (n_1,\dots,n_K)^T\right)$. The vector $\boldsymbol{\alpha}$ is also known as the "pseudocounts".

Posterior predictive. Let $\boldsymbol{\alpha}' := \boldsymbol{\alpha} + (n_1, \ldots, n_K)^T$.

$$
\begin{aligned}
P(\hat{x} \mid x_{1:N}) &= \int_{\boldsymbol{\theta}} p(\hat{x}, \boldsymbol{\theta} \mid x_{1:N}) \, \mathrm{d}\boldsymbol{\theta} && \text{(sum rule)} && (43)\\
&= \int_{\boldsymbol{\theta}} p(\hat{x} \mid \boldsymbol{\theta}, x_{1:N}) p(\boldsymbol{\theta} \mid x_{1:N}) \, \mathrm{d}\boldsymbol{\theta} && \text{(product rule)} && (44)\\
&= \int_{\boldsymbol{\theta}} p(\hat{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid x_{1:N}) \, \mathrm{d}\boldsymbol{\theta} && \text{(conditional independence } \hat{x} \perp\!\!\!\perp x_{1:N} | \boldsymbol{\theta}) && (45)\\
&= \int_{\boldsymbol{\theta}} \mathrm{Categorical}(\hat{x} \mid \boldsymbol{\theta}) \, \mathrm{Dirichlet}(\boldsymbol{\theta} \mid \boldsymbol{\alpha}') \, \mathrm{d}\boldsymbol{\theta} && \text{(from the model and from (42))} && (46)\\
&= \int_{\boldsymbol{\theta}} \theta_{\hat{x}} \, \mathrm{Dirichlet}(\boldsymbol{\theta} \mid \boldsymbol{\alpha}') \, \mathrm{d}\boldsymbol{\theta} && && (47)\\
&= \mathbb{E}_{\boldsymbol{\theta} \sim \mathrm{Dirichlet}(\boldsymbol{\alpha}')}[\theta_{\hat{x}}] && \text{(the } \hat{x}\text{th element of the mean)} && (48)\\
&= \frac{\alpha'_{\hat{x}}}{\sum_k \alpha'_k} && \text{(read off tables)} && (49)\\
&= \mathrm{Categorical}\left(\hat{x} \middle| \frac{\boldsymbol{\alpha}'}{\sum_k \alpha'_k}\right). && && (50)
\end{aligned}
$$

That is to say, our belief about the next data point $\hat{x}$ under this model given the data $x_{1:N}$ follows a Categorical distribution with the normalised updated pseudocounts as parameters.

Marginal likelihood.

$$
\begin{aligned}
p(x_{1:N}) &= \frac{p(\boldsymbol{\theta}, x_{1:N})}{p(\boldsymbol{\theta}|x_{1:N})} && \text{(Bayes' rule)} && (51)\\
&= \frac{\mathrm{Dirichlet}(\boldsymbol{\theta}|\alpha) \prod_k \theta_k^{n_k}}{\mathrm{Dirichlet}\left(\boldsymbol{\theta}|\alpha + (n_1, \ldots, n_K)^T\right)} && \text{(substitute from the model and (42))} && (52)\\
&= \frac{\left[\frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \left(\prod_k \theta_k^{\alpha_k - 1}\right)\right] \prod_k \theta_k^{n_k}}{\left[\frac{\Gamma\left(N + \sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k + n_k)} \prod_k \theta_k^{\alpha_k + n_k - 1}\right]} && \text{(substitute expressions for Dirichlet density)} && (53)\\
&= \frac{\Gamma\left(\sum_k \alpha_k\right) \prod_k \Gamma(\alpha_k + n_k)}{\Gamma\left(N + \sum_k \alpha_k\right) \prod_k \Gamma(\alpha_k)} && \text{(cancel terms).} && (54)
\end{aligned}
$$

3.1.1 Interpretation

One real-life scenario that can be modelled using the Dirichlet-Categorical model is the following. Let θ_k represent the proportion of customers of age between $10k$ and $10(k+1)$ with $K = 15$ that are served by an organisation. Then

- $\boldsymbol{\alpha}$ is the "prior" belief about what the (unnormalised) proportions should be,
- x_n is the age bracket of the nth observed customer,
- $\boldsymbol{\theta}|x_{1:N}$ is the customer age distribution after incorporating the actual age data $x_{1:N}$, represented with uncertainty (variance) that shrinks as $N \to \infty$,
- $\hat{x}|x_{1:N}$ allows you to predict the next customer's age whilst taking into account uncertainty about the value of $\boldsymbol{\theta}$.

Important. Learning, prediction, and inspection can all be expressed in terms of analytic and computational manipulation of probability (marginalisation and conditioning) within a model specified as a joint distribution of latents and observed values.

3.1.2 Conjugate pairs

Here, we have seen that modelling data points $x_{1:N}$ as a joint distribution consisting of the Dirichlet prior over the latent $\boldsymbol{\theta}$ and the Categorical likelihood for data x_n given $\boldsymbol{\theta}$ has allowed us to answer inference questions using analytic integration. The Dirichlet and Categorical distributions form a *conjugate pair* where the Dirichlet

distribution is a *conjugate prior* for the Categorical likelihood. Wikipedia page on conjugate priors[2] contains a list of conjugate pairs for modelling various types of data. Conjugate pairs are often used as a building block of a more complicated model so that we can partially integrate things out as part of the underlying inference algorithm (for example, see Section 5).

3.2 Harder Problem: Gaussian Mixture Model

Consider again the Gaussian Mixture Model introduced in Section 2.5. One of the inferential questions we might want to ask is whether two points come from the same cluster. We can formulate this as an expectation under the posterior as follows:

$$P(z_i = z_j|\mathbf{x}_{1:N}) = \sum_{z_i,z_j} P(z_i, z_j|\mathbf{x}_{1:N})\, \mathbb{1}(z_i = z_j) \tag{55}$$

$$= \sum_{z_{1:N}} P(z_{1:N}|\mathbf{x}_{1:N})\, \mathbb{1}(z_i = z_j) \tag{56}$$

$$= \int_{\boldsymbol{\pi}} \int_{\boldsymbol{\Lambda}_{1:K}} \int_{\boldsymbol{\mu}_{1:K}} \left[\sum_{z_{1:N}} p(z_{1:N}, \boldsymbol{\Lambda}_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\pi}|\mathbf{x}_{1:N})\, \mathbb{1}(z_i = z_j) \right] \mathrm{d}\boldsymbol{\mu}_{1:K}\, \mathrm{d}\boldsymbol{\Lambda}_{1:K}\, \mathrm{d}\boldsymbol{\pi} \tag{57}$$

$$= \mathbb{E}_{(z_{1:N},\theta_{1:K},\boldsymbol{\pi})\sim p(\cdot|\mathbf{x}_{1:N})}[\mathbb{1}(z_i = z_j)]. \tag{58}$$

This looks difficult, but actually, thanks to the carefully chosen priors, we are able to do most of the integration analytically. This is not always the case in which case we turn to approximate inference algorithms such as Monte Carlo integration.

In this problem, we have been able to formulate our question as finding an expectation of a test function (in this case $\mathbb{1}(z_i = z_j)$ under the posterior. In general, inference, prediction, and inspection can all be expressed as expectation of some test function f under some target distribution $p(x)$.

3.2.1 Aside: How to Build a Model for Your Own Problem?

Statistics, machine learning, and other fields consist of the exploration and characterisation of the space of models. Most models you see are "convenient" in that much integration is analytically possible and that inference, inspection, and prediction are all known to work well and reliably for a wide variety of problems. The design of new models involves equal parts knowledge of a problem domain, familiarity with statistical model building blocks, and mathematical/computational aesthetic. Often it is possible to creatively map your specific inference problem onto an existing model, saving time and effort.

3.3 Monte Carlo Integration

As we have seen in the previous examples, it is important to be able to find expectations of various functions of random variables distributed according to some target distribution. Let X be the random variable of interest, distributed according to P and f be the test function. We are interested in the expectation of $f(X)$. We can obtain a *Monte Carlo* estimator by sampling from P:

$$\mathbb{E}[f(X)] = \sum_{x'} f(x')P(x') \approx \frac{1}{N} \sum_{n=1}^{N} f(x^{(n)}) =: \hat{f}, \tag{59}$$

where $x^{(n)}, n = 1, \ldots, N$ are independent samples from P. We call $\hat{f}$ the Monte Carlo estimator of $\mathbb{E}[f(x)]$.

Claim 3.1 (Unbiasedness of the Monte Carlo estimator)**.** *The estimator $\hat{f}$ is unbiased, i.e. $\mathbb{E}[\hat{f}] = \mathbb{E}[f(X)]$.*

Proof.

$$\mathbb{E}[\hat{f}] = \mathbb{E}\left[\frac{1}{N} \sum_{n=1}^{N} f(x^{(n)})\right] \tag{60}$$

$$= \frac{1}{N} \sum_{n=1} \mathbb{E}[f(x^{(n)})] \qquad \text{(from linearity of expectations)} \tag{61}$$

$$= \frac{1}{N} \sum_{n=1} \mathbb{E}[f(X)] \qquad \text{(since } x^{(n)} \text{ is a sample from } P\text{)} \tag{62}$$

$$= \mathbb{E}[f(X)]. \tag{63}$$

[2] https://en.wikipedia.org/wiki/Conjugate_prior

□

Claim 3.2 (Variance of the Monte Carlo estimator)**.** *The variance of $\hat{f}$ is independent of the dimensionality of X and decreases at the rate of $1/N$.*

Proof.

$$\mathrm{Var}[\hat{f}] = \mathrm{Var}\left[\frac{1}{N}\sum_{n=1}^{N} f(x^{(n)})\right] \tag{64}$$

$$= \frac{1}{N^2}\sum_{n=1}^{N} \mathrm{Var}\left[f(x^{(n)})\right] \qquad \text{(see Appendix A)} \tag{65}$$

$$= \frac{1}{N^2}\sum_{n=1}^{N} \mathrm{Var}\left[f(X)\right] \qquad \text{(since } x^{(n)} \text{ is a sample from } P\text{)} \tag{66}$$

$$= \frac{1}{N}\mathrm{Var}[f(X)]. \tag{67}$$

□

If f is poorly behaved, N is small, or the $x^{(n)}$ are correlated, the variance of the estimator could be quite large. In sampling-based inference there is usually a difficult practical trade-off between the latter two.

In general, generating samples from complex distributions encountered in machine learning is not easy. Typically, most of the probability is concentrated in regions whose volume is a tiny fraction of the total. In Section 4, we will explore methods to search for these relevant regions in order to sample from this complex distribution.

3.4 Rejection Sampling

One correct, but extremely inefficient algorithm to sample from probability distributions is *rejection sampling*.

Our goal is to sample from a target distribution $p(x)$ whose density can be evaluated only up to a normalising constant Z_p, i.e. we can only evaluate $\tilde{p}(x) = Z_p p(x)$. [3]

Assume that we can sample from a proposal distribution q with a density $q(x)$ which can also only be evaluated up to a normalising constant Z_q, i.e. we can only evaluate $\tilde{q}(x) = Z_q q(x)$. Assume also, that we can bound $\tilde{p}(x)$ by $k\tilde{q}(x)$ for a finite k for all x in the support:

$$k\tilde{q}(x) \geq \tilde{p}(x) \text{ for all } x. \tag{68}$$

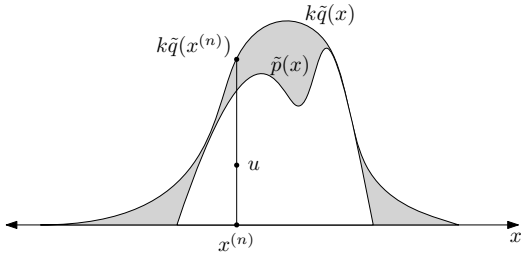

Figure 8: Illustration of rejection sampling.

The procedure that generates N samples $x^{(n)}, n = 1, \ldots, N$ from p is described in Algorithm 1 below.

Algorithm 1 Rejection sampling

1: $n \leftarrow 1$.
2: **while** $n \leq N$ **do**
3: Sample x from q.
4: Sample u from $\mathrm{Uniform}(0, k\tilde{q}(x))$.
5: **if** $u \leq \tilde{p}(x)$ **then**
6: $x^{(n)} \leftarrow x$.
7: $n \leftarrow n + 1$.
return $\{x^{(n)}\}_{1:N}$.

[3] This is the case when the target is the Bayesian posterior, in which we can't evaluate the posterior density $p(x|y)$ but can evaluate $\tilde{p}(x|y) = p(x, y) = Z_p p(x|y)$ where $Z_p = p(y)$ is the marginal likelihood.

3.4.1 Why it works?

Sampling $x \sim q$ and $u \sim \text{Uniform}(0, k\tilde{q}(x))$ yields a pair of values uniformly distributed in the grey region in Figure 8. If $u \leq \tilde{p}(x)$, then x is accepted, otherwise it is rejected and the process repeats until a sample is accepted. Accepted pairs are uniformly distributed in the white area; dropping u yields a sample distributed according to $\tilde{p}(x)$, and equivalently, $p(x)$. The efficiency of rejection sampling depends critically on the match between the proposal distribution and the distribution of interest.

3.4.2 Conditioning via Ancestral Sampling and Rejection

Consider a directed graphical model in which x are the latents and y are the observed variables. One can sample from the joint model $P(x, y)$ by successively sampling from random variables in the topological order of the corresponding nodes. In other words, start sampling from random variables/nodes that don't have any parents, then keep sampling ones which already have sampled parents (Exercise: Check why this can always be done!). This way of sampling from the joint $P(x, y)$ is called *ancestral sampling*.

We can combine ancestral sampling and rejection sampling to generate samples from the posterior $P(x|\hat{y})$ for a given observe $\hat{y}$ of interest.

$$\tilde{P}(x, y) = P(x, y)\, \mathbb{1}(y = \hat{y}) \tag{69}$$

$$= \begin{cases} P(x, \hat{y}) & \text{if } y = \hat{y} \\ 0 & \text{otherwise} \end{cases} \tag{70}$$

$$= \begin{cases} P(\hat{y})P(x|\hat{y}) & \text{if } y = \hat{y} \\ 0 & \text{otherwise.} \end{cases} \tag{71}$$

Here, we have set up $\tilde{P}(x, y)$ so that it is really an unnormalised version of $P(x|\hat{y})$ (normalising constant $Z_P = P(\hat{y}) = \sum_x P(x, \hat{y})$) assigning zero probability for all (x, y) in which $y \neq \hat{y}$.

The proposal distribution is $\tilde{Q}(x, y) = Q(x, y) := P(x, y)$. Sampling from Q is done using ancestral sampling. We can see that for $k = 1$,

$$k\tilde{Q}(x, y) = kP(x, y) \geq P(x, y)\, \mathbb{1}(y = \hat{y}) = \tilde{Q}(x, y) \tag{72}$$

for all (x, y).

Using these choices for $\tilde{P}$, $\tilde{Q}$ and k, the algorithm for rejection sampling 1 simplifies to Algorithm 2. Note that we have combined lines 4 and 5 in Algorithm 1 to line 4 in Algorithm 2 since they are equivalent for these choices of $\tilde{P}$, $\tilde{Q}$ and k.

Algorithm 2 Conditioning via Ancestral Sampling and Rejection

1: $n \leftarrow 1$.
2: **while** $n \leq N$ **do**
3: Sample (x, y) from $P(x, y)$ via ancestral sampling.
4: **if** $y = \hat{y}$ **then**
5: $x^{(n)} \leftarrow x$.
6: $n \leftarrow n + 1$.
return $\{x^{(n)}\}_{1:N}$.

Unless the prior and posterior are extremely well matched this will be an extremely inefficient sampler. It doesn't make sense to consider this algorithm for continuous observations since the probability of $\{y = \hat{y}\}$ which is the probability of accepting a sample is zero.

4 Markov Chain Monte Carlo

At a high level, algorithms such as rejection sampling don't include a searching element which is one of the reasons it doesn't work well in high dimensions. Markov Chain Monte Carlo (MCMC) is a family of algorithms that incorporate searching. An MCMC algorithm works by building a Markov Chain which converges to a unique invariant distribution (hopefully the same as the one we are after) and can be used to estimate expectations with respect to this distribution.

We first review Markov Chains and then present two most commonly used MCMC algorithms: Metropolis-Hastings and Gibbs sampling.

4.1 Markov Chains

Definition 4.1 (Markov Chain). *A first order*[4] *Markov Chain is a series of random variables $X^{(0)}, X^{(1)}, X^{(2)}, \dots$ for which the following conditional independence holds:*

$$P(x^{(n)}|x^{(1)}, \dots, x^{(n-1)}) = P(x^{(n)}|x^{(n-1)}) \qquad \text{for } n = 1, 2, \dots \tag{73}$$

A Markov Chain is specified by the *initial distribution* for $X^{(0)}$ and the *transition distribution* $T_n(x^{(n)} \to x^{(n+1)}) := P(x^{(n+1)}|x^{(n)})$.

Definition 4.2 (Homogeneous Markov Chains). *A Markov Chain is homogeneous if $T_n = T$ for all $n \in \{0, 1, 2, \dots\}$.*

Denote the probabilities of the marginal distribution of $X^{(n)}$ by p_n. p_n can be expressed as

$$p_n(x^{(n)}) := P(x^{(n)}) = \sum_{x^{(n-1)}} P(x^{(n-1)})P(x^{(n)}|x^{(n-1)}) = \sum_{x^{(n-1)}} P(x^{(n-1)})T(x^{(n-1)} \to x^{(n)}). \tag{74}$$

Definition 4.3 (Invariant distribution). *A distribution is invariant (or stationary) with respect to a Markov Chain if the transition distribution of that chain leaves that distribution unchanged. More formally, a distribution $\pi(x)$ is the invariant distribution of the Markov Chain with the transition T if, for all x:*

$$\pi(x) = \sum_{x'} \pi(x')T(x' \to x). \tag{75}$$

We are interested in constructing Markov Chains for which the distribution we wish to sample from is invariant.

Definition 4.4 (Detailed balance). *A transition distribution T is reversible if there exists a unique distribution π such that*

$$\pi(x)T(x \to x') = \pi(x')T(x' \to x). \tag{76}$$

This condition is called detailed balance.

Detailed balance implies that π is an invariant distribution since

$$\sum_{x'} \pi(x')T(x' \to x) = \sum_{x'} \pi(x)T(x \to x') = \pi(x) \sum_{x'} T(x \to x') = \pi(x). \tag{77}$$

For our purposes, it is not enough merely to find a Markov Chain with respect to which the distribution we wish to sample from is invariant. We also require the Markov chain to be *ergodic*—that the probabilities at time n, $P(x^{(n)})$ converge to this invariant distribution as $n \to \infty$, regardless of the choice of initial probabilities $P(x^{(0)})$. This property is called *ergodicity*.

Theorem 4.1 (Fundamental Theorem). *If a homogeneous Markov chain on a finite state space with transition distribution T has π as an invariant distribution and*

$$\nu = \min_x \min_{x':\pi(x')>0} T(x \to x')/\pi(x') > 0 \tag{78}$$

then

1. *that Markov chain is* ergodic, *i.e. for all x, regardless of the initial distribution $P(x^{(0)})$*

$$\lim_{n\to\infty} p_n(x) = \pi(x) \tag{79}$$

2. *If $f(x)$ is a real-valued function from the state space of the Markov chain, then the expectation of f with respect to the distribution p_n, written $\mathbb{E}_n[f] := \sum_x f(x)p_n(x)$, converges to its expectation with respect to π, written $\mathbb{E}[f] := \sum_x f(x)\pi(x)$. I.e.*

$$\lim_{n\to\infty} \mathbb{E}_n[f] = \mathbb{E}[f]. \tag{80}$$

We therefore have

- Detailed balance (equation (76)) as a sufficient condition for π being the invariant distribution, and

[4] First order refers to the fact that $X^{(n)}$ is only dependent on one previous random variable, $X^{(n-1)}$.

- A condition to ensure that T is ergodic, i.e. T avoids traps and can visit everywhere in our state space (equation (78)).

Note that there are alternative conditions such as *regularity* or [*irreducibility* and *aperiodicity*] which imply ergodicity.[5]

The theorem as stated guarantees only that at large times, the distribution will be close to the invariant distribution. It does not say how dependent the states at different times might be, and hence does not guarantee that the average value of a function over a long period of time converges to the function's expected value. In order to guarantee that we can use samples from a single realisation of a Markov chain to estimate expectations, a whole family of ergodic theorems has been developed.[6] The bottom line is: *We can approximate integrals in equation 59 using samples from single Markov chains even if they are dependent and the state space is not discrete.*

It is often convenient to construct the transition distribution from a set of base transition distributions, given by $B_1, \ldots, B_K$, each of which leaves the desired distribution invariant, but which may not be ergodic individually.

Proposition 4.1 (Mixture of Base Transitions)**.** *Let $B_1, \ldots, B_K$ be a set of base transition distributions each of which is invariant with respect to π. Let $\alpha_1, \ldots, \alpha_K$ be mixture probabilities for the base transition distributions where $\alpha_k > 0$ and $\sum_k \alpha_k = 1$. I.e. the transition distribution is given by:*

$$T(x \to x') = \sum_k \alpha_k B_k(x \to x'). \tag{81}$$

If π is invariant with respect to each of the B_k, then it is also invariant with respect to T. Furthermore, if each of the B_k satisfy detailed balance, T will as well.

Proof.

$$\begin{aligned}
\sum_{x'} \pi(x')T(x' \to x) &= \sum_{x'} \left(\pi(x') \sum_k \alpha_k B_k(x' \to x) \right) && (82)\\
&= \sum_k \left(\alpha_k \sum_{x'} \pi(x') B_k(x' \to x) \right) && (83)\\
&= \sum_k \alpha_k \pi(x) && (84)\\
&= \pi(x) \sum_k \alpha_k && (85)\\
&= \pi(x). && (86)
\end{aligned}$$

□

Proposition 4.2 (Sequence of Base Transitions)**.** *Let $B_1, \ldots, B_K$ be base transition distributions applied in sequence. If each B_k is invariant with respect to π, the resulting transition distribution obtained by applying these base distributions in sequence is also invariant with respect to π:*

Proof. Represent the transition distribution by a transition matrix $\mathbf{T}$ where $T_{ij} = T(i \to j)$ and the target distribution π by a row vector $\boldsymbol{\pi}$ where $\pi_i = \pi(i)$. Similarly, represent the base distributions by matrices $\mathbf{B}_1, \ldots, \mathbf{B}_K$. The transition matrix is $\mathbf{T} = \mathbf{B}_1 \cdots \mathbf{B}_K$. Since each B_k is invariant, we have $\boldsymbol{\pi}\mathbf{B}_k = \boldsymbol{\pi}$ for each k and hence

$$\boldsymbol{\pi}\mathbf{T} = \boldsymbol{\pi}(\mathbf{B}_1 \cdots \mathbf{B}_K) = \boldsymbol{\pi}(\mathbf{B}_2 \cdots \mathbf{B}_K) = \cdots = \boldsymbol{\pi}\mathbf{B}_K = \boldsymbol{\pi}, \tag{87}$$

which means that T is invariant with respect to π. □

4.2 Metropolis-Hastings Algorithm

The Metropolis Hastings (MH) algorithm generates a Markov chain a particular stationary distribution. Assume we can sample from a proposal distribution $q(\cdot|x) := q(x \to \cdot)$. Let π be the required distribution (stationary distribution for this Markov chain). Assume we can only evaluate q and π up to a multiplicative factor (i.e. we can only evaluate $\tilde{q}(x \to x') = Z_q q(x \to x')$ and $\tilde{\pi}(x) = Z_p \pi(x)$). The MH algorithm is outlined in Algorithm 3.

[5] For a detailed explanation of these concepts, consult Neal (1993).

[6] For those interested: Roberts et al. (2004, Fact 5), Robert and Casella (2013, p. 241, Theorem 6.63), Meyn and Tweedie (2012, p. 433, Theorem 17.3.2).

Algorithm 3 Metropolis Hastings algorithm

1: Sample $x^{(0)}$ from an arbitrary probability distribution.
2: **for** $n = 1, \ldots, N$ **do**
3: Sample $x' \sim q(x^{(n-1)} \to \cdot)$.
4: Let the acceptance probability of x' be

$$a \leftarrow A(x^{(n-1)} \to x') = \min\left(1, \frac{\tilde{\pi}(x')\tilde{q}(x' \to x^{(n-1)})}{\tilde{\pi}(x^{(n-1)})\tilde{q}(x^{(n-1)} \to x')}\right). \tag{88}$$

5: Sample $u \sim \text{Uniform}(0, 1)$.
6: **if** $u < a$ **then**
7: $x^{(n)} \leftarrow x'$.
8: **else**
9: $x^{(n)} \leftarrow x^{(n-1)}$.

4.2.1 Why it works?

We need to prove that π is the unique stationary distribution of this Markov chain.

We can express the transition distribution as follows:

$$T(x \to x') = \begin{cases} q(x \to x')A(x \to x') & \text{if } x \neq x' \\ q(x \to x) + \sum_{x', x' \neq x} q(x \to x')(1 - A(x \to x')) & \text{if } x = x'. \end{cases} \tag{89}$$

To prove that π is a stationary distribution of this Markov chain, we make sure the detailed balance equation holds.

For $x \neq x'$, we have

$$\pi(x)T(x \to x') = \pi(x)q(x \to x') \min\left(1, \frac{\pi(x')q(x' \to x)}{\pi(x)q(x \to x')}\right) \tag{90}$$

$$= \min\left(\pi(x)q(x \to x'), \pi(x')q(x' \to x)\right) \tag{91}$$

$$= \pi(x')q(x' \to x) \min\left(1, \frac{\pi(x)q(x \to x')}{\pi(x')q(x' \to x)}\right) \tag{92}$$

$$= \pi(x')T(x' \to x) \tag{93}$$

For $x = x'$, the detailed balance equation $\pi(x)T(x \to x') = \pi(x')T(x' \to x)$ obviously holds.

Provided that q is chosen so that equation (78) holds, we obtain a general-purpose Markov chain for simulating from and thereby computing expectations with respect to arbitrary distributions π.

4.3 Gibbs Sampling

Assume we want to sample from $\pi(\mathbf{x}) := \pi(x_1, \ldots, x_D)$. We can only sample from the conditionals $\pi(x_i|\mathbf{x}_{-i})$ where $\mathbf{x}_{-i}$ denotes $x_{1:D}$ with the i^{th} component omitted, i.e. $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_D)$. The Gibbs sampling algorithm (4) is given below.

Algorithm 4 Gibbs sampling algorithm

1: Sample $\mathbf{x}^{(0)}$ from an arbitrary probability distribution.
2: **for** $n = 1, \ldots, N$ **do**
3: Sample $x_1^{(n)} \sim \pi(\cdot|x_2^{(n-1)}, x_3^{(n-1)}, \ldots, x_D^{(n-1)})$
4: Sample $x_2^{(n)} \sim \pi(\cdot|x_1^{(n)}, x_3^{(n-1)}, \ldots, x_D^{(n-1)})$
5: $\vdots$
6: Sample $x_D^{(n)} \sim \pi(\cdot|x_1^{(n)}, x_2^{(n)}, \ldots, x_{D-1}^{(n)})$

4.3.1 Why it works?

To show that this procedure leaves π as the invariant distribution, we can treat each of the conditionals as a base distribution. Due to Proposition 4.2, it is sufficient for us to show that each base distribution has π as the

invariant distribution.

$$\sum_{\mathbf{x}} \pi(\mathbf{x}) B_k(\mathbf{x} \to \mathbf{x}') = \sum_{\mathbf{x}} [\pi(x_k|\mathbf{x}_{-k})\pi(\mathbf{x}_{-k})] \cdot \left[\pi(x'_k|\mathbf{x}_{-k}) \prod_{i \neq k} \mathbb{1}(x_i = x'_i) \right] \tag{94}$$

$$= \pi(x'_k|\mathbf{x}'_{-k})\pi(\mathbf{x}'_{-k}) \sum_{x_k} \pi(x_k|\mathbf{x}'_{-k}) \tag{95}$$

$$= \pi(\mathbf{x}'). \tag{96}$$

Due to Proposition 4.1, the algorithm could have equally sampled from the mixture of base distributions instead of going through them in sequence.

We can also view Gibbs sampling as an instance of the MH algorithm. If the proposal of MH $q_k(\mathbf{x} \to \mathbf{x}')$ is set to be $\pi(x'_k|\mathbf{x}_{-k}) \cdot \prod_{i \neq k} \mathbb{1}(x_i = x_k)$ the acceptance probability is one (shown below) and so it is equivalent to one sampling step in Gibbs sampling.

$$A(\mathbf{x} \to \mathbf{x}') = \min\left(1, \frac{\pi(\mathbf{x}')q_k(\mathbf{x}' \to \mathbf{x})}{\pi(\mathbf{x})q_k(\mathbf{x} \to \mathbf{x}')}\right) \tag{97}$$

$$= \min\left(1, \frac{\pi(\mathbf{x}')\pi(x_k|\mathbf{x}'_{-k}) \cdot \prod_{i \neq k} \mathbb{1}(x'_i = x'_k)}{\pi(\mathbf{x})\pi(x'_k|\mathbf{x}_{-k}) \cdot \prod_{i \neq k} \mathbb{1}(x_i = x_k)}\right) \tag{98}$$

$$= \min\left(1, \frac{\pi(x'_k|\mathbf{x}'_{-k})\pi(\mathbf{x}'_{-k})\pi(x_k|\mathbf{x}'_{-k}) \prod_{i \neq k} \mathbb{1}(x'_i = x'_k)}{\pi(x_k|\mathbf{x}_{-k})\pi(\mathbf{x}_{-k})\pi(x'_k|\mathbf{x}_{-k}) \prod_{i \neq k} \mathbb{1}(x_i = x_k)}\right) \tag{99}$$

$$= 1 \tag{100}$$

5 Gibbs Sampler for Gaussian Mixture Models

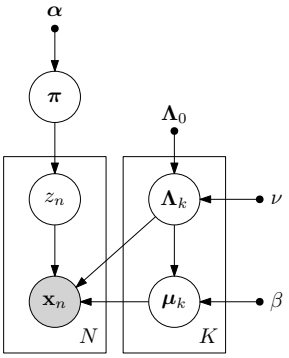

Figure 9: Graphical model for the Gaussian mixture model.

This section is adapted from internal notes written by Brooks Paige[7] and Tom Rainforth[8].

Recall that the joint probability is

$$p(\mathbf{x}_{1:N}, z_{1:N}, \boldsymbol{\Lambda}_{1:K}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\pi}|\alpha, \boldsymbol{\Lambda}_0, \nu, \beta) = p(\boldsymbol{\pi}|\alpha) \left[\prod_{n=1}^{N} p(z_n|\boldsymbol{\pi}) p(\mathbf{x}_n|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Lambda}_{z_n})\right] \left[\prod_{k=1}^{K} p(\boldsymbol{\Lambda}_k|\boldsymbol{\Lambda}_0, \nu) p(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k, \beta)\right] \tag{101}$$

where

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \text{Dirichlet}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \tag{102}$$

$$p(\boldsymbol{\Lambda}_k|\boldsymbol{\Lambda}_0, \nu) = \text{Wishart}(\boldsymbol{\Lambda}_k|\boldsymbol{\Lambda}_0, \nu) \tag{103}$$

$$p(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k, \beta) = \text{Normal}(\boldsymbol{\mu}_k|\mathbf{0}, (\beta\boldsymbol{\Lambda}_k)^{-1}) \tag{104}$$

$$P(z_n|\boldsymbol{\pi}) = \text{Categorical}(z_n|\boldsymbol{\pi}) \tag{105}$$

$$p(\mathbf{x}_n|z_n,\ \boldsymbol{\mu}_{1:K},\ \boldsymbol{\Lambda}_{1:K}) = \text{Normal}(\mathbf{x}_n|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Lambda}_{z_n}^{-1}). \tag{106}$$

Possible inference goals include: estimating candidate cluster centers and covariances; checking whether any two data points are in the same cluster; and estimating how many distinct clusters exist in the data.

[7] brooks@robots.ox.ac.uk

[8] twgr@robots.ox.ac.uk

5.1 Gibbs sampler

A basic Gibbs sampler would sweep through each latent random variable in turn—all the z_n, all the $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$, and $\boldsymbol{\pi}$—and sample each conditioned on the current settings of all other random variables. However, this can mix poorly.

A more efficient Gibbs sampler sweeps through sampling all the z_n in turn, with the other random variables marginalised out analytically; our carefully-chosen priors are exactly those such that this analytic integration is possible. To be completely clear: in the "basic" Gibbs sampler, each z_n would be sampled from the full conditional distribution

$$z_n \sim p(z_n|\mathbf{z}_{-n}, \boldsymbol{\mu}_{1:K}, \boldsymbol{\Lambda}_{1:K}, \boldsymbol{\pi}, \mathbf{x}_{1:N}) \tag{107}$$

where $\mathbf{z}_{-n}$ is all elements of $\mathbf{z}$ except for the z_n under consideration. This, of course, requires values of each of the $\boldsymbol{\mu}_{1:K}, \boldsymbol{\Lambda}_{1:K}, \boldsymbol{\pi}$. The "collapsed" Gibbs sampler instead draws each from the conditional distribution

$$z_n \sim p(z_n|\mathbf{z}_{-n}, \mathbf{x}_{1:N}). \tag{108}$$

5.1.1 Useful known results for conjugate pairs

What does this distribution look like, though? We're going to take advantage of two known results Gelman et al. (2004). The first exploits Dirichlet-Multinomial conjugacy to express a predictive distribution in closed form, marginalizing over the latent dirichlet $\boldsymbol{\pi}$:

$$p(z_n = k|\mathbf{z}_{-n}, \alpha) = \int p(z_n = k|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\mathbf{z}_{-n}, \alpha) d\boldsymbol{\pi} = \frac{N_k^{(-n)} + \alpha}{N - 1 + K\alpha} \tag{109}$$

where $N_k^{(-n)}$ is the count of the number of elements of $\mathbf{z}_{-n}$ which are assigned to cluster k. A derivation of this result can be found on the Wikipedia page for the "Dirichlet-Multinomial distribution".

The second analytic integration uses Normal-Wishart conjugacy to marginalise over latent means and precisions. Let $\tilde{\mathbf{x}}_k^{(-n)} \subseteq \mathbf{x}_{-n}$ denote the set of points $\mathbf{x}_i$ such that $z_i = k$. Then we can marginalise out $\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k$ with

$$\begin{aligned} p(\mathbf{x}_n|z_n = k, \tilde{\mathbf{x}}_k^{(-n)}) &= \iint p(\mathbf{x}_n|z_n = k, \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) p(\boldsymbol{\mu}_k|\tilde{\mathbf{x}}_k^{(-n)}, \boldsymbol{\Lambda}_k, \beta) p(\boldsymbol{\Lambda}_k|\tilde{\mathbf{x}}_k^{(-n)}, \boldsymbol{\Lambda}_0, \nu) d\boldsymbol{\mu}_k d\boldsymbol{\Lambda}_k \\ &= t_{(\nu_k^* - D + 1)}\left(\mathbf{x}_n \middle| \boldsymbol{\mu}_k^*, \frac{\boldsymbol{\Lambda}_k^*(\beta_k^* + 1)}{\beta_k^*(\nu_k^* - D + 1)}\right) \end{aligned} \tag{110}$$

where D is the dimension of each data point $\mathbf{x}_n$, and the multivariate t-distribution is defined as

$$t_\nu(\mathbf{x}_n|\mathbf{m}, \mathbf{W}) = \frac{\Gamma((\nu + D)/2)}{\Gamma(\nu/2)\nu^{D/2}\pi^{D/2}} |\mathbf{W}|^{-1/2} \left(1 + \frac{1}{\nu}(\mathbf{x}_n - \mathbf{m})^\top \mathbf{W}^{-1}(\mathbf{x}_n - \mathbf{m})\right)^{-(\nu+D)/2}. \tag{111}$$

The parameters of that t-distribution are given by

$$\boldsymbol{\mu}_k^* = \frac{\beta\boldsymbol{\mu}_0 + N_k^{(-n)}\bar{\mathbf{x}}_k}{\beta + N_k^{(-n)}} \tag{112}$$

$$\boldsymbol{\Lambda}_k^* = \boldsymbol{\Lambda}_0 + \sum_{n \in \mathcal{A}_k^{(-n)}} (\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^\top + \frac{\beta N_k^{(-n)}}{\beta + N_k^{(-n)}} (\bar{\mathbf{x}}_k - \boldsymbol{\mu}_0)(\bar{\mathbf{x}}_k - \boldsymbol{\mu}_0)^\top \tag{113}$$

$$\beta_k^* = \beta + N_k^{(-n)} \tag{114}$$

$$\nu_k^* = \nu + N_k^{(-n)}. \tag{115}$$

(in our example model, note that $\boldsymbol{\mu}_0 = \mathbf{0}$.) Here we have introduced a little extra notation, defining index sets for the current cluster assignments, and per-cluster means, using

$$\mathcal{A}_k^{(-n)} := \{i : z_i = k, i \neq n\} \qquad \bar{\mathbf{x}}_k := \frac{1}{N_k^{(-n)}} \sum_{n \in \mathcal{A}_k^{(-n)}} \mathbf{x}_n. \tag{116}$$

5.1.2 Collapsed Gibbs sampler

Great; now let's collapse the Gibbs sampler. Recall, we want to sample from $p(z_n|\mathbf{z}_{-n}, \mathbf{x}_{1:N})$. For each $z_n = k$, this is proportional to the joint distribution of both $z_n = k$ and $\mathbf{x}_n$, with

$$p(z_n = k|\mathbf{z}_{-n}, \mathbf{x}_{1:N}) = \frac{p(z_n = k, \mathbf{x}_n|\mathbf{z}_{-n}, \mathbf{x}_{-n})}{\sum_{j=1}^{K} p(z_n = j, \mathbf{x}_n|\mathbf{z}_{-n}, \mathbf{x}_{-n})}. \tag{117}$$

So, if we can compute that numerator, we can write a collapsed Gibbs sampler. (To avoid clutter, we'll avoid writing out the hyperparameters $\alpha, \beta, \nu, \mathbf{\Lambda}_0$ to the right of the conditioning bar.) Using the results in the previous section, and the conditional independence structure of the model,

$$\begin{aligned}
p(z_n = k, \mathbf{x}_n|\mathbf{z}_{-n}, \mathbf{x}_{-n}) &= \iiint p(z_n = k, \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}_{1:K}, \mathbf{\Lambda}_{1:K}|\mathbf{z}_{-n}, \mathbf{x}_{-n}) d\boldsymbol{\pi} d\boldsymbol{\mu}_{1:K} d\mathbf{\Lambda}_{1:K} \\
&= \iiint p(z_n = k, \mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\mu}_k, \mathbf{\Lambda}_k|\mathbf{z}_{-n}, \mathbf{x}_{-n}) d\boldsymbol{\pi} d\boldsymbol{\mu}_k d\mathbf{\Lambda}_k \prod_{j \neq k} \underbrace{\iint p(\boldsymbol{\mu}_j, \mathbf{\Lambda}_j|\tilde{\mathbf{x}}_j^{(-n)}) d\boldsymbol{\mu}_j d\mathbf{\Lambda}_j}_{=1} \\
&= \iiint p(\mathbf{x}_n|z_n = k, \boldsymbol{\mu}_k, \mathbf{\Lambda}_k) p(z_n = k|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\mathbf{z}_{-n}) p(\boldsymbol{\mu}_k, \mathbf{\Lambda}_k|\tilde{\mathbf{x}}_k^{(-n)}) d\boldsymbol{\pi} d\boldsymbol{\mu}_k d\mathbf{\Lambda}_k \\
&= \frac{N_k^{(-n)} + \alpha}{N - 1 + K\alpha} \iint p(\mathbf{x}_n|z_n = k, \boldsymbol{\mu}_k, \mathbf{\Lambda}_k) p(\boldsymbol{\mu}_k, \mathbf{\Lambda}_k|\tilde{\mathbf{x}}_k^{(-n)}) d\boldsymbol{\mu}_k d\mathbf{\Lambda}_k \\
&= \left(\frac{N_k^{(-n)} + \alpha}{N - 1 + K\alpha} \right) \times t_{(\nu_k^* - D + 1)} \left(\mathbf{x}_n \middle| \boldsymbol{\mu}_k^*, \frac{\mathbf{\Lambda}_k^* (\beta_k^* + 1)}{\beta_k^* (\nu_k^* - D + 1)} \right).
\end{aligned} \tag{118}$$

Now, the Gibbs sampler can sweep through $\mathbf{z}$, and for each n evaluate equation 118 for each cluster id k and sample according to these (unnormalised) weights a new value of z_n.

References

Zoubin Ghahramani. Unsupervised learning. In *Advanced lectures on machine learning*, pages 72–112. Springer, 2004.

Christopher M Bishop. Pattern recognition and machine learning (information science and statistics). 2006.

Radford M Neal. Probabilistic inference using markov chain monte carlo methods. 1993.

Gareth O Roberts, Jeffrey S Rosenthal, et al. General state space markov chains and mcmc algorithms. *Probability Surveys*, 1:20–71, 2004.

Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.

A Miscellaneous

Random variables X and Y are independent if $p(x, y) = p(x, y)$.
Expectation of a random variable X is $\mathbb{E}[X] = \int x p(x) \, \mathrm{d}x$.
If X and Y are independent random variables

$$\begin{aligned}
\mathbb{E}[XY] &= \int \int xy p(x, y) \, \mathrm{d}x \, \mathrm{d}y && (119) \\
&= \int \int xy p(x) p(y) \, \mathrm{d}x \, \mathrm{d}y && (120) \\
&= \int x p(x) \, \mathrm{d}x \int y p(y) \, \mathrm{d}y && (121) \\
&= \mathbb{E}[X] \, \mathbb{E}[Y]. && (122)
\end{aligned}$$

Hence their covariance, defined by $\mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y]$ is zero.

If X and Y are independent random variables, then the variance of the sum is the sum of the variance:

$$\begin{aligned}
\operatorname{Var}[X+Y] &= \mathbb{E}[(X+Y)^2] - [\mathbb{E}[X+Y]]^2 && (123)\\
&= \mathbb{E}[X^2 + 2XY + Y^2] - \left[\mathbb{E}[X]^2 + 2\,\mathbb{E}[X]\,\mathbb{E}[Y] + \mathbb{E}[Y]^2\right] && (124)\\
&= \left(\mathbb{E}[X^2] - \mathbb{E}[X]^2\right) + \left(\mathbb{E}[Y^2] - \mathbb{E}[Y]^2\right) + 2\left(\mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y]\right) && (125)\\
&= \operatorname{Var}[X] + \operatorname{Var}[Y] + 2\left(\mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y]\right) && (126)\\
&= \operatorname{Var}[X] + \operatorname{Var}[Y]. && (127)
\end{aligned}$$

For a constant $a \in \mathbb{R}$ and a random variable X:

$$\begin{aligned}
\operatorname{Var}[aX] &= \mathbb{E}[(aX)^2] - [\mathbb{E}[aX]]^2 && (128)\\
&= a^2\,\mathbb{E}[X^2] - a^2\,\mathbb{E}[X]]^2 && (129)\\
&= a^2\left(\mathbb{E}[X^2] - \mathbb{E}[X]^2\right) && (130)\\
&= a^2\operatorname{Var}[X]. && (131)
\end{aligned}$$

B Probability distributions

Summarised in Table 1.

Distribution	Parameters	Support	PDF/PMF	Mean	Variance/Covariance
Bernoulli	$\theta \in [0,1]$	$x \in \{0,1\}$	$\begin{cases}\theta & \text{if } x=1 \\ 1-\theta & \text{if } x=0\end{cases}$	θ	$\theta(1-\theta)$
Beta	$\alpha, \beta > 0$	$x \in [0,1]$	$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
Binomial	$N \in \mathbb{N}, \theta \in [0,1]$	$x \in \{0,\dots,N\}$	$\binom{N}{x}\theta^x(1-\theta)^{N-x}$	$N\theta$	$N\theta(1-\theta)$
Gamma	$\alpha, \beta > 0$	$x > 0$	$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1}\exp(-\beta x)$	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^2}$
Categorical	$\boldsymbol{\theta} \in [0,1]^K, \sum_k \theta_k = 1$	$x \in \{1,\dots,K\}$	θ_x	N/A	N/A
Dirichlet	$\boldsymbol{\alpha} \in (0,\infty)^K$	$\mathbf{x} \in [0,1]^K, \sum_k x_k = 1$	$\frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma(\alpha_k)} \prod_k x_k^{\alpha_k-1}$	$\frac{\boldsymbol{\alpha}}{\sum_k \alpha_k}$	$\mathrm{Var}[x_k] = \frac{\alpha_k(\alpha_0-\alpha_k)}{\alpha_0^2(\alpha_0+1)}$
Multivariate Normal	$\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{\Sigma}$ positive semi-definite $d \times d$	$\mathbf{x} \in \mathbb{R}^d$	$\frac{1}{\sqrt{\lvert 2\pi\boldsymbol{\Sigma}\rvert}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$
Normal	$\mu \in \mathbb{R}$, $\sigma^2 > 0$	$x \in \mathbb{R}$	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	μ	σ^2
Poisson	$\lambda > 0$	$x \in \{0,1,2,\dots\}$	$\frac{\lambda^x}{x!}\exp(-\lambda)$	λ	λ

Table 1: Summary of common probability distributions

[a]

[a] Where $\Gamma(z) := \int_0^\infty x^{z-1}e^{-x}\,\mathrm{d}x$ is the *Gamma function*; more on `https://en.wikipedia.org/wiki/Gamma_function`.