Chapter 2

Nonparametric and Graphical Models

STATISTICAL methods play a central role in the design and analysis of machine vision systems. In this background chapter, we review several learning and inference techniques upon which our later contributions are based. We begin in Sec. 2.1 by describing exponential families of probability densities, emphasizing the roles of sufficiency and conjugacy in Bayesian learning. Sec. 2.2 then shows how graphs may be used to impose structure on exponential families. We contrast several types of graphical models, and provide results clarifying their underlying statistical assumptions.

To apply graphical models in practical applications, computationally efficient learning and inference algorithms are needed. Sec. 2.3 describes several variational methods which approximate intractable inference tasks via message–passing algorithms. In Sec. 2.4, we discuss a complementary class of Monte Carlo methods which use stochastic simulations to analyze complex models. In this thesis, we propose new inference algorithms which integrate variational and Monte Carlo methods in novel ways.

Finally, we conclude in Sec. 2.5 with an introduction to nonparametric methods for Bayesian learning. These infinite–dimensional models achieve greater robustness by avoiding restrictive assumptions about the data generation process. Despite this flexibility, variational and Monte Carlo methods can be adapted to allow tractable analysis of large, high–dimensional datasets.

■ 2.1 Exponential Families

An *exponential family* of probability distributions [15, 36, 311] is characterized by the values of certain *sufficient statistics*. Let x be a random variable taking values in some sample space $\mathcal{X}$, which may be either continuous or discrete. Given a set of statistics or *potentials* $\{\phi_a \mid a \in \mathcal{A}\}$, the corresponding exponential family of densities is given by

$$p(x \mid \theta) = \nu(x) \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \phi_a(x) - \Phi(\theta) \right\} \tag{2.1}$$

where $\theta \in \mathbb{R}^{|\mathcal{A}|}$ are the family's *natural* or *canonical* parameters, and $\nu(x)$ is a non-negative *reference measure*. In some applications, the parameters θ are set to fixed constants, while in other cases they are interpreted as latent random variables. The *log partition function* $\Phi(\theta)$ is defined to normalize $p(x \mid \theta)$ so that it integrates to one:

$$\Phi(\theta) = \log \int_{\mathcal{X}} \nu(x) \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \phi_a(x) \right\} dx \tag{2.2}$$

For discrete spaces, dx is taken to be counting measure, so that integrals become summations. This construction is valid when the canonical parameters θ belong to the set Θ for which the log partition function is finite:

$$\Theta \triangleq \left\{ \theta \in \mathbb{R}^{|\mathcal{A}|} \mid \Phi(\theta) < \infty \right\} \tag{2.3}$$

Because $\Phi(\theta)$ is a convex function (see Prop. 2.1.1), Θ is necessarily convex. If Θ is also open, the exponential family is said to be *regular*. Many classic probability distributions form regular exponential families, including the Bernoulli, Poisson, Gaussian, beta, and gamma densities [21, 107]. For example, for scalar Gaussian densities the sufficient statistics are $\{x, x^2\}$, $\nu(x) = 1$, and Θ constrains the variance to be positive.

Exponential families are typically parameterized so that no linear combination of the potentials $\{\phi_a \mid a \in \mathcal{A}\}$ is almost everywhere constant. In such a *minimal* representation,[1] there is a unique set of canonical parameters θ associated with each density in the family, whose *dimension* equals $d \triangleq |\mathcal{A}|$. Furthermore, the exponential family defines a d–dimensional Riemannian manifold, and the canonical parameters a coordinate system for that manifold. By characterizing the convex geometric structure of such manifolds, *information geometry* [6, 15, 52, 74, 305] provides a powerful framework for analyzing learning and inference algorithms. In particular, as we discuss in Sec. 2.3, results from *conjugate duality* [15, 311] underlie many algorithms used in this thesis.

In the following sections, we further explore the properties of exponential families, emphasizing results which guide the specification of sufficient statistics appropriate to particular learning problems. We then introduce a family of conjugate priors for the canonical parameters θ, and provide detailed computational methods for two exponential families (the normal–inverse–Wishart and Dirichlet–multinomial) used extensively in this thesis. For further discussion of the convex geometry underlying exponential families, see [6, 15, 36, 74, 311].

■ 2.1.1 Sufficient Statistics and Information Theory

In this section, we establish several results which motivate the use of exponential families, and clarify the notion of sufficiency. The following properties of the log partition function establish its central role in the study of exponential families:

[1] We note, however, that *overcomplete* representations play an important role in recent theoretical analyses of variational approaches to approximate inference [305, 306, 311].

Proposition 2.1.1. *The log partition function* $\Phi(\theta)$ *of eq.* (2.2) *is convex (strictly so for minimal representations) and continuously differentiable over its domain* Θ. *Its derivatives are the* cumulants *of the sufficient statistics* $\{\phi_a \mid a \in \mathcal{A}\}$, *so that*

$$\frac{\partial \Phi(\theta)}{\partial \theta_a} = \mathbb{E}_\theta[\phi_a(x)] \triangleq \int_{\mathcal{X}} \phi_a(x)\, p(x \mid \theta)\, dx \tag{2.4}$$

$$\frac{\partial^2 \Phi(\theta)}{\partial \theta_a \partial \theta_b} = \mathbb{E}_\theta[\phi_a(x)\, \phi_b(x)] - \mathbb{E}_\theta[\phi_a(x)]\, \mathbb{E}_\theta[\phi_b(x)] \tag{2.5}$$

Proof. For a detailed proof of this classic result, see [15, 36, 311]. The cumulant generating properties follow from the chain rule and algebraic manipulation. From eq. (2.5), $\nabla^2 \Phi(\theta)$ is a positive semi–definite covariance matrix, implying convexity of $\Phi(\theta)$. For minimal families, $\nabla^2 \Phi(\theta)$ must be positive definite, guaranteeing strict convexity. □

Due to this result, the log partition function is also known as the *cumulant generating function* of the exponential family. The convexity of $\Phi(\theta)$ has important implications for the geometry of exponential families [6, 15, 36, 74].

Entropy, Information, and Divergence

Concepts from information theory play a central role in the study of learning and inference in exponential families. Given a probability distribution $p(x)$ defined on a discrete space $\mathcal{X}$, Shannon's measure of *entropy* (in natural units, or *nats*) equals

$$H(p) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{2.6}$$

In such diverse fields as communications, signal processing, and statistical physics, entropy arises as a natural measure of the inherent uncertainty in a random variable [49]. The *differential entropy* extends this definition to continuous spaces:

$$H(p) = -\int_{\mathcal{X}} p(x) \log p(x)\, dx \tag{2.7}$$

In both discrete and continuous domains, the (differential) entropy $H(p)$ is concave, continuous, and maximal for uniform densities. However, while the discrete entropy is guaranteed to be non-negative, differential entropy is sometimes less than zero.

For problems of model selection and approximation, we need a measure of the distance between probability distributions. The *relative entropy* or *Kullback-Leibler (KL) divergence* between two probability distributions $p(x)$ and $q(x)$ equals

$$D(p \,||\, q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}\, dx \tag{2.8}$$

Important properties of the KL divergence follow from *Jensen's inequality* [49], which bounds the expectation of convex functions:

$$\mathbb{E}[f(x)] \geq f(\mathbb{E}[x]) \qquad \text{for any convex } f : \mathcal{X} \to \mathbb{R} \tag{2.9}$$

Applying Jensen's inequality to the logarithm of eq. (2.8), which is concave, it is easily shown that the KL divergence $D(p\,||\,q) \geq 0$, with $D(p\,||\,q) = 0$ if and only if $p(x) = q(x)$ almost everywhere. However, it is not a true distance metric because $D(p\,||\,q) \neq D(q\,||\,p)$. Given a target density $p(x)$ and an approximation $q(x)$, $D(p\,||\,q)$ can be motivated as the information gain achievable by using $p(x)$ in place of $q(x)$ [49]. Interestingly, the alternate KL divergence $D(q\,||\,p)$ also plays an important role in the development of variational methods for approximate inference (see Sec. 2.3).

An important special case arises when we consider the dependency between two random variables x and y. Let $p_{xy}(x, y)$ denote their joint distribution, $p_x(x)$ and $p_y(y)$ their corresponding marginals, and $\mathcal{X}$ and $\mathcal{Y}$ their sample spaces. The *mutual information* between x and y then equals

$$I(p_{xy}) \triangleq D(p_{xy}\,||\,p_x p_y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} p_{xy}(x, y) \log \frac{p_{xy}(x, y)}{p_x(x) p_y(y)}\, dy\, dx \tag{2.10}$$

$$= H(p_x) + H(p_y) - H(p_{xy}) \tag{2.11}$$

where eq. (2.11) follows from algebraic manipulation. The mutual information can be interpreted as the expected reduction in uncertainty about one random variable from observation of another [49].

Projections onto Exponential Families

In many cases, learning problems can be posed as a search for the best approximation of an empirically derived *target* density $\tilde{p}(x)$. As discussed in the previous section, the KL divergence $D(\tilde{p}\,||\,q)$ is a natural measure of the accuracy of an approximation $q(x)$. For exponential families, the optimal approximating density is elegantly characterized by the following *moment–matching* conditions:

Proposition 2.1.2. *Let $\tilde{p}$ denote a target probability density, and p_θ an exponential family. The approximating density minimizing $D(\tilde{p}\,||\,p_\theta)$ then has canonical parameters $\hat{\theta}$ chosen to match the expected values of that family's sufficient statistics:*

$$\mathbb{E}_{\hat{\theta}}[\phi_a(x)] = \int_{\mathcal{X}} \phi_a(x)\, \tilde{p}(x)\, dx \qquad\qquad a \in \mathcal{A} \tag{2.12}$$

For minimal families, these optimal parameters $\hat{\theta}$ are uniquely determined.

Proof. From the definition of KL divergence (eq. (2.8)), we have

$$\begin{aligned} D(\tilde{p}\,||\,p_\theta) &= \int_{\mathcal{X}} \tilde{p}(x) \log \frac{\tilde{p}(x)}{p(x \mid \theta)}\, dx \\ &= \int_{\mathcal{X}} \tilde{p}(x) \log \tilde{p}(x)\, dx - \int_{\mathcal{X}} \tilde{p}(x) \left[\log \nu(x) + \sum_{a \in \mathcal{A}} \theta_a \phi_a(x) - \Phi(\theta) \right] dx \\ &= -H(\tilde{p}) - \int_{\mathcal{X}} \tilde{p}(x) \log \nu(x)\; dx - \sum_{a \in \mathcal{A}} \theta_a \int_{\mathcal{X}} \phi_a(x)\, \tilde{p}(x)\, dx + \Phi(\theta) \end{aligned}$$

Taking derivatives with respect to θ_a and setting $\partial D(\tilde{p}\,||\,p_\theta)\,/\partial\theta_a = 0$, we then have

$$\frac{\partial\Phi(\theta)}{\partial\theta_a} = \int_{\mathcal{X}} \phi_a(x)\,\tilde{p}(x)\,dx \qquad\qquad a \in \mathcal{A}$$

Equation (2.12) follows from the cumulant generating properties of $\Phi(\theta)$ (eq. (2.4)). Because $\Phi(\theta)$ is strictly convex for minimal families (Prop. 2.1.1), the canonical parameters $\hat{\theta}$ satisfying eq. (2.12) achieve the unique global minimum of $D(\tilde{p}\,||\,p_\theta)$. □

In information geometry, the density satisfying eq. (2.12) is known as the *I–projection* of $\tilde{p}(x)$ onto the *e–flat manifold* defined by the exponential family's canonical parameters [6, 52]. Note that the optimal projection depends only the potential functions' expected values under $\tilde{p}(x)$, so that these statistics are *sufficient* to determine the closest approximation.

In many applications, rather than an explicit target density $\tilde{p}(x)$, we instead observe L independent samples $\{x^{(\ell)}\}_{\ell=1}^{L}$ from that density. In this situation, we define the *empirical density* of the samples as follows:

$$\tilde{p}(x) = \frac{1}{L}\sum_{\ell=1}^{L} \delta\big(x, x^{(\ell)}\big) \tag{2.13}$$

Here, $\delta\big(x, x^{(\ell)}\big)$ is the Dirac delta function for continuous $\mathcal{X}$, and the Kronecker delta for discrete $\mathcal{X}$. Specializing Prop. 2.1.2 to this case, we find a correspondence between information projection and *maximum likelihood (ML)* parameter estimation.

Proposition 2.1.3. *Let p_θ denote an exponential family with canonical parameters θ. Given L independent, identically distributed samples $\{x^{(\ell)}\}_{\ell=1}^{L}$, with empirical density $\tilde{p}(x)$ as in eq. (2.13), the maximum likelihood estimate $\hat{\theta}$ of the canonical parameters coincides with the empirical density's information projection:*

$$\hat{\theta} = \arg\max_{\theta} \sum_{\ell=1}^{L} \log p(x^{(\ell)} \mid \theta) = \arg\min_{\theta} D(\tilde{p}\,||\,p_\theta) \tag{2.14}$$

These optimal parameters are uniquely determined for minimal families, and characterized by the following moment matching conditions:

$$\mathbb{E}_{\hat{\theta}}[\phi_a(x)] = \frac{1}{L}\sum_{\ell=1}^{L} \phi_a(x^{(\ell)}) \qquad\qquad a \in \mathcal{A} \tag{2.15}$$

Proof. Expanding the KL divergence from $\tilde{p}(x)$ (eq. (2.13)), we have

$$\begin{aligned} D(\tilde{p}\,||\,p_\theta) &= \int_{\mathcal{X}} \tilde{p}(x) \log \tilde{p}(x)\, dx - \int_{\mathcal{X}} \tilde{p}(x) \log p(x \mid \theta)\; dx \\ &= -H(\tilde{p}) - \int_{\mathcal{X}} \frac{1}{L} \sum_{\ell=1}^{L} \delta\big(x, x^{(\ell)}\big) \log p(x \mid \theta)\; dx \\ &= -H(\tilde{p}) - \frac{1}{L} \sum_{\ell=1}^{L} \log p(x^{(\ell)} \mid \theta) \end{aligned}$$

Because $H(\tilde{p})$ does not depend on θ, the parameters minimizing $D(\tilde{p}\,||\,p_\theta)$ and maximizing the expected log–likelihood coincide, establishing eq. (2.14). The unique characterization of $\hat{\theta}$ via moment–matching (eq. (2.15)) then follows from Prop. 2.1.2. □

In principle, Prop. 2.1.2 and 2.1.3 suggest a straightforward procedure for learning exponential familes: estimate appropriate sufficient statistics, and then find corresponding canonical parameters via convex optimization [6, 15, 36, 52]. In practice, however, significant difficulties may arise. For example, practical applications often require *semi-supervised learning* from partially labeled training data, so that the needed statistics cannot be directly measured. Even when sufficient statistics are available, calculation of the corresponding parameters can be intractable in large, complex models.

These results also have important implications for the selection of appropriate exponential families. In particular, because the chosen statistics are sufficient for parameter estimation, the learned model *cannot* capture aspects of the target distribution neglected by these statistics. These concerns motivate our later development of *nonparametric* methods (see Sec. 2.5) which extend exponential families to learn richer, more flexible models.

Maximum Entropy Models

In the previous section, we argued that certain statistics are sufficient to characterize the best exponential family approximation of a given target density. The following theorem shows that if these statistics are the *only* available information about a target density, then the corresponding exponential family provides a natural model.

Theorem 2.1.1. *Consider a collection of statistics $\{\phi_a \mid a \in \mathcal{A}\}$, whose expectations with respect to some target density $\tilde{p}(x)$ are known:*

$$\int_{\mathcal{X}} \phi_a(x)\, \tilde{p}(x)\, dx = \mu_a \qquad\qquad a \in \mathcal{A} \tag{2.16}$$

The unique distribution $\hat{p}(x)$ maximizing the entropy $H(\hat{p})$, subject to these moment constraints, is then a member of the exponential family of eq. (2.1), with $\nu(x) = 1$ and canonical parameters $\hat{\theta}$ chosen so that $\mathbb{E}_{\hat{\theta}}[\phi_a(x)] = \mu_a$.

Proof. The general form of eq. (2.1) can be motivated by a Lagrangian formulation of this constrained optimization problem. Taking derivatives, the Lagrange multipliers become the exponential family's canonical parameters. Global optimality can then be verified via a bound based on the KL divergence [21, 49]. A related characterization of exponential families with reference measures $\nu(x) \neq 1$ is also possible [21]. □

Note that eq. (2.16) implicitly assumes the existence of *some* distribution satisfying the specified moment constraints. In general, verifying this feasibility can be extremely challenging [311], relating to classic moment inequality [25, 176] and covariance extension [92, 229] problems. Also, given insufficient moment constraints for non–compact continous spaces, the maximizing density may be improper and have infinite entropy.

Recall that the entropy measures the inherent uncertainty in a random variable. Thus, if the sufficient statistics of eq. (2.16) are the only available characterization of a target density, the corresponding exponential family is justified as the model which imposes the fewest additional assumptions about the data generation process.

■ 2.1.2 Learning with Prior Knowledge

The results of the previous sections show how exponential families use sufficient statistics to characterize the *likelihood* of observed training data. Frequently, however, we also have *prior* knowledge about the expected location, scale, concentration, or other features of the process generating the data. When learning from small datasets, consistent incorporation of prior knowledge can dramatically improve the accuracy and robustness of the resulting model.

In this section, we develop Bayesian methods for learning and inference which treat the "parameters" of exponential family densities as random variables. In addition to allowing easy incorporation of prior knowledge, this approach provides natural confidence estimates for models learned from noisy or sparse data. Furthermore, it leads to powerful methods for transferring knowledge among multiple related learning tasks. See Bernardo and Smith [21] for a more formal, comprehensive survey of this topic.

Analysis of Posterior Distributions

Given an exponential family $p(x \mid \theta)$ with canonical parameters θ, Bayesian analysis begins with a *prior distribution* $p(\theta \mid \lambda)$ capturing any available knowledge about the data generation process. This prior distribution is typically itself a member of a family of densities with *hyperparameters* λ. For the moment, we assume these hyperparameters are set to some fixed value based on our prior beliefs.

Given L independent, identically distributed observations $\{x^{(\ell)}\}_{\ell=1}^{L}$, two computations arise frequently in statistical analyses. Using Bayes' rule, the *posterior distribution*

of the canonical parameters can be written as follows:

$$p(\theta \mid x^{(1)}, \ldots, x^{(L)}, \lambda) = \frac{p(x^{(1)}, \ldots, x^{(L)} \mid \theta, \lambda)\, p(\theta \mid \lambda)}{\int_\Theta p(x^{(1)}, \ldots, x^{(L)} \mid \theta, \lambda)\, p(\theta \mid \lambda)\; d\theta} \tag{2.17}$$

$$\propto p(\theta \mid \lambda) \prod_{\ell=1}^{L} p(x^{(\ell)} \mid \theta) \tag{2.18}$$

The proportionality symbol of eq. (2.18) represents the constant needed to ensure integration to unity (in this case, the data likelihood of eq. (2.17)). Recall that, for minimal exponential families, the canonical parameters are uniquely associated with expectations of that family's sufficient statistics (Prop. 2.1.3). The posterior distribution of eq. (2.18) thus captures our knowledge about the statistics likely to be exhibited by future observations.

In many situations, statistical models are used primarily to predict future observations. Given L independent observations as before, the *predictive likelihood* of a new observation $\bar{x}$ equals

$$p(\bar{x} \mid x^{(1)}, \ldots, x^{(L)}, \lambda) = \int_\Theta p(\bar{x} \mid \theta)\, p(\theta \mid x^{(1)}, \ldots, x^{(L)}, \lambda)\; d\theta \tag{2.19}$$

where the posterior distribution over parameters is as in eq. (2.18). By averaging over our posterior uncertainty in the parameters θ, this approach leads to predictions which are typically more robust than those based on a single parameter estimate.

In principle, a fully Bayesian analysis should also place a prior distribution $p(\lambda)$ on the hyperparameters. In practice, however, computational considerations frequently motivate an *empirical Bayesian* approach [21, 75, 107] in which λ is estimated by maximizing the training data's marginal likelihood:

$$\hat{\lambda} = \arg\max_\lambda\, p(x^{(1)}, \ldots, x^{(L)} \mid \lambda) \tag{2.20}$$

$$= \arg\max_\lambda \int_\Theta p(\theta \mid \lambda) \prod_{\ell=1}^{L} p(x^{(\ell)} \mid \theta)\; d\theta \tag{2.21}$$

In situations where this optimization is intractable, cross–validation approaches which optimize the predictive likelihood of a held–out data set are often useful [21].

More generally, the predictive likelihood computation of eq. (2.19) is itself intractable for many practical models. In these cases, the parameters' posterior distribution (eq. (2.18)) is often approximated by a single *maximum a posteriori (MAP)* estimate:

$$\hat{\theta} = \arg\max_\theta\, p(\theta \mid x^{(1)}, \ldots, x^{(L)}, \lambda) \tag{2.22}$$

$$= \arg\max_\theta\, p(\theta \mid \lambda) \prod_{\ell=1}^{L} p(x^{(\ell)} \mid \theta) \tag{2.23}$$

This approach is best justified when the training set size L is very large, so that the posterior distribution of eq. (2.22) is tightly concentrated [21, 107]. Sometimes, however, MAP estimates are used with smaller datasets because they are the only computationally viable option.

Parametric and Predictive Sufficiency

When computing the posterior distributions and predictive likelihoods motivated in the previous section, it is very helpful to have compact ways of characterizing large datasets. For exponential families, the notions of sufficiency introduced in Sec. 2.1.1 can be extended to simplify learning with prior knowledge.

Theorem 2.1.2. *Let $p(x \mid \theta)$ denote an exponential family with canonical parameters θ, and $p(\theta \mid \lambda)$ a corresponding prior density. Given L independent, identically distributed samples $\{x^{(\ell)}\}_{\ell=1}^{L}$, consider the following statistics:*

$$\boldsymbol{\phi}(x^{(1)}, \dots, x^{(L)}) \triangleq \left\{ \frac{1}{L} \sum_{\ell=1}^{L} \phi_a(x^{(\ell)}) \;\middle|\; a \in \mathcal{A} \right\} \tag{2.24}$$

These empirical moments, along with the sample size L, are then said to be parametric sufficient *for the posterior distribution over canonical parameters, so that*

$$p(\theta \mid x^{(1)}, \dots, x^{(L)}, \lambda) = p(\theta \mid \boldsymbol{\phi}(x^{(1)}, \dots, x^{(L)}), L, \lambda) \tag{2.25}$$

Equivalently, they are predictive sufficient *for the likelihood of new data $\bar{x}$:*

$$p(\bar{x} \mid x^{(1)}, \dots, x^{(L)}, \lambda) = p(\bar{x} \mid \boldsymbol{\phi}(x^{(1)}, \dots, x^{(L)}), L, \lambda) \tag{2.26}$$

Proof. Parametric sufficiency follows from the Neyman factorization criterion, which is satisfied by any exponential family. The correspondence between parametric and predictive sufficiency can then be argued from eqs. (2.18, 2.19). For details, see Sec. 4.5 of Bernardo and Smith [21]. □

This theorem makes exponential families particularly attractive when learning from large datasets, due to the often dramatic compression provided by the statistics of eq. (2.24). It also emphasizes the importance of selecting appropriate sufficient statistics, since other features of the data cannot affect subsequent model predictions.

Analysis with Conjugate Priors

Theorem 2.1.2 shows that statistical predictions in exponential families are functions solely of the chosen sufficient statistics. However, it does not provide an explicit characterization of the posterior distribution over model parameters, or guarantee that the predictive likelihood can be computed tractably. In this section, we describe an expressive family of prior distributions which are also analytically tractable.

Let $p(x \mid \theta)$ denote a family of probability densities parameterized by θ. A family of prior densities $p(\theta \mid \lambda)$ is said to be *conjugate* to $p(x \mid \theta)$ if, for any observation x and hyperparameters λ, the posterior distribution $p(\theta \mid x, \lambda)$ remains in that family:

$$p(\theta \mid x, \lambda) \propto p(x \mid \theta)\, p(\theta \mid \lambda) \propto p\big(\theta \mid \bar{\lambda}\big) \tag{2.27}$$

In this case, the posterior distribution is compactly described by an updated set of hyperparameters $\bar{\lambda}$. For exponential families parameterized as in eq. (2.1), conjugate priors [21, 36] take the following general form:

$$p(\theta \mid \lambda) = \exp\left\{\sum_{a \in \mathcal{A}} \theta_a \lambda_0 \lambda_a - \lambda_0 \Phi(\theta) - \Omega(\lambda)\right\} \tag{2.28}$$

While this functional form duplicates the exponential family's, the interpretation is different: the density is over the space of parameters Θ, and determined by hyperparameters λ. The conjugate prior is *proper*, or normalizable, when the hyperparameters take values in the space Λ where the log normalization constant $\Omega(\lambda)$ is finite:

$$\Omega(\lambda) = \log \int_{\Theta} \exp\left\{\sum_{a \in \mathcal{A}} \theta_a \lambda_0 \lambda_a - \lambda_0 \Phi(\theta)\right\} d\theta \tag{2.29}$$

$$\Lambda \triangleq \Big\{\lambda \in \mathbb{R}^{|\mathcal{A}|+1} \mid \Omega(\lambda) < \infty\Big\} \tag{2.30}$$

Note that the dimension of the conjugate family's hyperparameters λ is one larger than the corresponding canonical parameters θ.

The following result verifies that the conjugate family of eq. (2.28) satisfies the definition of eq. (2.27), and provides an intuitive interpretation for the hyperparameters:

Proposition 2.1.4. *Let $p(x \mid \theta)$ denote an exponential family with canonical parameters θ, and $p(\theta \mid \lambda)$ a family of conjugate priors defined as in eq. (2.28). Given L independent samples $\{x^{(\ell)}\}_{\ell=1}^{L}$, the posterior distribution remains in the same family:*

$$p(\theta \mid x^{(1)}, \ldots, x^{(L)}, \lambda) = p\big(\theta \mid \bar{\lambda}\big) \tag{2.31}$$

$$\bar{\lambda}_0 = \lambda_0 + L \qquad \bar{\lambda}_a = \frac{\lambda_0 \lambda_a + \sum_{\ell=1}^{L} \phi_a(x^{(\ell)})}{\lambda_0 + L} \qquad a \in \mathcal{A} \tag{2.32}$$

Integrating over Θ, the log–likelihood of the observations can then be compactly written using the normalization constant of eq. (2.29):

$$\log p(x^{(1)}, \ldots, x^{(L)} \mid \lambda) = \Omega\big(\bar{\lambda}\big) - \Omega(\lambda) + \sum_{\ell=1}^{L} \log \nu(x^{(\ell)}) \tag{2.33}$$

Proof. Expanding the posterior distribution as in eq. (2.18), we have

$$\begin{aligned}
p(\theta \mid x^{(1)}, \ldots, x^{(L)}, \lambda) &\propto p(\theta \mid \lambda) \prod_{\ell=1}^{L} p(x^{(\ell)} \mid \theta) \\
&\propto \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \lambda_0 \lambda_a - \lambda_0 \Phi(\theta) \right\} \prod_{\ell=1}^{L} \nu(x^{(\ell)}) \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \phi_a(x^{(\ell)}) - \Phi(\theta) \right\} \\
&\propto \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \left(\lambda_0 \lambda_a + \sum_{\ell=1}^{L} \phi_a(x^{(\ell)}) \right) - (\lambda_0 + L)\, \Phi(\theta) \right\} \prod_{\ell=1}^{L} \nu(x^{(\ell)}) \\
&\propto \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \, (\lambda_0 + L) \left(\frac{\lambda_0 \lambda_a + \sum_{\ell=1}^{L} \phi_a(x^{(\ell)})}{\lambda_0 + L} \right) - (\lambda_0 + L)\, \Phi(\theta) \right\}
\end{aligned}$$

Note that the last line absorbs the reference measure terms, which are constant with respect to θ, into the proportionality constant. The posterior hyperparameters of eq. (2.32) can now be verified by comparison with eq. (2.28). Likelihoods are determined by the following integral over Θ:

$$\begin{aligned}
p(x^{(1)}, \ldots, x^{(L)} \mid \lambda) &= \int_{\Theta} p(\theta \mid \lambda) \prod_{\ell=1}^{L} p(x^{(\ell)} \mid \theta)\, d\theta \\
&= \int_{\Theta} \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \lambda_0 \lambda_a - \lambda_0 \Phi(\theta) - \Omega(\lambda) \right\} \prod_{\ell=1}^{L} \nu(x^{(\ell)}) \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \phi_a(x^{(\ell)}) - \Phi(\theta) \right\} d\theta \\
&= \exp\{-\Omega(\lambda)\} \int_{\Theta} \exp\left\{ \sum_{a \in \mathcal{A}} \theta_a \left(\lambda_0 \lambda_a + \sum_{\ell=1}^{L} \phi_a(x^{(\ell)}) \right) - (\lambda_0 + L) \Phi(\theta) \right\} d\theta \prod_{\ell=1}^{L} \nu(x^{(\ell)})
\end{aligned}$$

Identifying the second term as an unnormalized conjugate prior, with hyperparameters $\bar{\lambda}$, the log–likelihood of eq. (2.33) then follows from eq. (2.29). □

Note that the predictive likelihood $p(\bar{x} \mid x^{(1)}, \ldots, x^{(L)}, \lambda)$ of eq. (2.19) arises as a special case of Prop. 2.1.4, where eq. (2.33) is used to determine the likelihood of $\bar{x}$ given hyperparameters incorporating previous observations (eq. (2.32)). For many common exponential families, the log normalization constant $\Omega(\lambda)$ can be determined in closed form, and likelihoods are easily computed.

Examining eq. (2.32), we see that the posterior hyperparameters $\bar{\lambda}_a$ are a weighted average of the prior hyperparameters λ_a and the corresponding sufficient statistics of the observations. Conjugate priors are thus effectively described by a set of synthetic *pseudo–observations*, where λ_a is interpreted as the average of $\phi_a(x)$ with respect to this synthetic data. Confidence in these prior statistics is expressed via the effective size $\lambda_0 > 0$ of this synthetic dataset, which need not be integral. This interpretation

often makes it easy to select an appropriate conjugate prior, since hyperparameters correspond to sufficient statistics with intuitive meaning.

When the number of observations L is large relative to λ_0, the posterior distribution of eq. (2.31) is primarily determined by the observed sufficient statistics. Thus, while conjugate families do not always contain truly non–informative *reference priors* [21], sufficiently uninformative, or *vague*, conjugate priors can typically be constructed when desired. More often, however, we find the ability to tractably include informative prior knowledge to be very useful. In cases where conjugate priors cannot adequately capture prior beliefs, mixtures of conjugate priors are often effective [21].

In principle, Prop. 2.1.4 provides a framework for conjugate analysis with any exponential family. In practice, however, canonical parameters may not provide the most convenient, computationally efficient representation. The following sections examine two conjugate families used extensively in this thesis, and develop specialized learning and inference methods with practical advantages.

■ 2.1.3 Dirichlet Analysis of Multinomial Observations

Consider a random variable x taking one of K discrete, categorical values, so that $\mathcal{X} = \{1, \ldots, K\}$. Any probability mass function, or distribution, $p(x)$ is then parameterized by the probabilities $\pi_k \triangleq \Pr[x = k]$ of the K discrete outcomes:

$$p(x \mid \pi_1, \ldots, \pi_K) = \prod_{k=1}^{K} \pi_k^{\delta(x,k)} \qquad \delta(x,k) \triangleq \begin{cases} 1 & x = k \\ 0 & x \neq k \end{cases} \tag{2.34}$$

Given L observations $\{x^{(\ell)}\}_{\ell=1}^{L}$, the *multinomial* distribution [21, 107, 229] gives the total probability of all possible length L discrete sequences taking those values:

$$p(x^{(1)}, \ldots, x^{(L)} \mid \pi_1, \ldots, \pi_K) = \frac{L!}{\prod_k C_k!} \prod_{k=1}^{K} \pi_k^{C_k} \qquad C_k \triangleq \sum_{\ell=1}^{L} \delta(x^{(\ell)}, k) \tag{2.35}$$

When $K = 2$, this is known as the *binomial* distribution. Through comparison with eq. (2.1), we see that multinomial distributions define regular exponential families with sufficient statistics $\phi_k(x) = \delta(x, k)$ and canonical parameters $\theta_k = \log \pi_k$. In a minimal representation, only the first $(K-1)$ statistics are necessary. The multinomial distribution is valid when its parameters lie in the $(K-1)$–simplex:

$$\Pi_{K-1} \triangleq \left\{ (\pi_1, \ldots, \pi_K) \;\middle|\; \pi_k \geq 0,\ \sum_{k=1}^{K} \pi_k = 1 \right\} \tag{2.36}$$

$$= \left\{ (\pi_1, \ldots, \pi_{K-1}, 1 - \textstyle\sum_{k=1}^{K-1} \pi_k) \;\middle|\; \pi_k \geq 0,\ \sum_{k=1}^{K-1} \pi_k \leq 1 \right\} \tag{2.37}$$

Note that the minimal representation of eq. (2.37) implicitly defines π_K as the complement of the probabilities of the other $(K-1)$ categories.

Given L observations as in eq. (2.35), Prop. 2.1.3 shows that the maximum likelihood estimates of the multinomial parameters $\pi = (\pi_1, \ldots, \pi_K)$ equal the empirical frequencies of the discrete categories:

$$\hat{\pi} = \arg\max_{\pi} \sum_{\ell=1}^{L} \log p(x^{(\ell)} \mid \pi) = \left(\frac{C_1}{L}, \ldots, \frac{C_K}{L} \right) \tag{2.38}$$

However, when L is not much larger than K, the ML estimate may assign zero probability to some values, and produce misleading predictions. In the following section, we describe a widely used family of conjugate priors which is useful in these situations.

Dirichlet and Beta Distributions

The *Dirichlet* distribution [21, 107] is the conjugate prior for the multinomial exponential family. Adapting the general form of eq. (2.28), the Dirichlet distribution with hyperparameters $\alpha = (\alpha_1, \ldots, \alpha_K)$ can be written as follows:

$$p(\pi \mid \alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \qquad \alpha_k > 0 \tag{2.39}$$

Note that the Dirichlet distribution's normalization constant involves a ratio of gamma functions. By convention, the exponents are defined to equal $(\alpha_k - 1)$ so that the density's mean has the following simple form:

$$\mathbb{E}_{\alpha}[\pi_k] = \frac{\alpha_k}{\alpha_0} \qquad \alpha_0 \triangleq \sum_{k=1}^{K} \alpha_k \tag{2.40}$$

We use $\mathrm{Dir}(\alpha)$ to denote a Dirichlet density with hyperparameters α. Samples can be drawn from a Dirichlet distribution by normalizing a set of K independent gamma random variables [107].

Often, we have no prior knowledge distinguishing the categories, and the K hyperparameters are thus set symmetrically as $\alpha_k = \alpha_0/K$. The variance of the multinomial parameters then equals

$$\mathrm{Var}_{\alpha}[\pi_k] = \frac{K - 1}{K^2(\alpha_0 + 1)} \qquad \alpha_k = \frac{\alpha_0}{K} \tag{2.41}$$

See [107] for other moments of the Dirichlet distribution. Because the variance is inversely proportional to α_0, it is known as the *precision* parameter. With a minor abuse of notation, we sometimes use $\mathrm{Dir}(\alpha_0)$ to denote this symmetric prior.

When $K = 2$, the Dirichlet distribution is equivalent to the *beta* distribution [107]. Denoting the beta density's two hyperparameters by α and β, let $\pi \sim \mathrm{Beta}(\alpha, \beta)$ indicate that

$$p(\pi \mid \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\,\Gamma(\beta)} \pi^{\alpha - 1}(1 - \pi)^{\beta - 1} \qquad \alpha, \beta > 0 \tag{2.42}$$

Note that by convention, samples from the beta density are the probability $\pi \in [0, 1]$ of the first category, while the two–dimensional Dirichlet distribution is equivalently expressed in terms of the probability vector $(\pi, 1 - \pi)$ (see eq. (2.39)). As in eqs. (2.40) and (2.41), the beta density's hyperparameters can be interpreted as setting the prior mean and variance of the binomial parameter π.

In Fig. 2.1, we illustrate several beta distributions. When $\alpha = \beta = 1$, it assigns equal prior probability to all possible binomial parameters π. Larger hyperparameters (smaller variances) lead to unimodal priors concentrated on the chosen mean. We also show examples of Dirichlet distributions on $K = 3$ multinomial categories, using the minimal 2–simplex representation of eq. (2.37). As with the beta density, setting $\alpha_k = 1$ ($\alpha_0 = K$) defines a uniform prior on the simplex, while larger precisions lead to unimodal priors. Interestingly, smaller values of the hyperparameters ($\alpha_k < 1$) favor *sparse* multinomial distributions which assign most of their probability mass to a subset of the categories.

When analyzing multinomial data, it is sometimes useful to consider *aggregate* distributions defined by combining a subset of the categories. If $\pi \sim \mathrm{Dir}(\alpha)$, the multinomial parameters attained by aggregation are also Dirichlet [107]. For example, combining the first two categories, we have

$$(\pi_1 + \pi_2, \pi_3, \ldots, \pi_K) \sim \mathrm{Dir}(\alpha_1 + \alpha_2, \alpha_3, \ldots, \alpha_K) \tag{2.43}$$

More generally, aggregation of any subset of the categories produces a Dirichlet distribution with hyperparameters summed as in eq. (2.43). In particular, the marginal distribution of any single component of a Dirichlet distribution follows a beta density:

$$\pi_k \sim \mathrm{Beta}(\alpha_k, \alpha_0 - \alpha_k) \tag{2.44}$$

This representation leads to an alternative, sequential procedure for drawing random Dirichlet samples [107, 147].

Conjugate Posteriors and Predictions

Consider a set of L observations $\{x^{(\ell)}\}_{\ell=1}^{L}$ from a multinomial distribution $p(x \mid \pi)$, with Dirichlet prior $p(\pi \mid \alpha)$. Via conjugacy, the posterior distribution is also Dirichlet:

$$\begin{aligned} p(\pi \mid x^{(1)}, \ldots, x^{(L)}, \alpha) &\propto p(\pi \mid \alpha)\, p(x^{(1)}, \ldots, x^{(L)} \mid \pi) \\ &\propto \prod_{k=1}^{K} \pi_k^{\alpha_k + C_k - 1} \propto \mathrm{Dir}(\alpha_1 + C_1, \ldots, \alpha_K + C_K) \end{aligned} \tag{2.45}$$

Here, C_k is the number of observations of category k, as in eq. (2.35). If L is sufficiently large, the mean of this posterior distribution (see eq. (2.40)) provides a useful summary statistic. We see that α_k is equivalent to a (possibly non–integral) number of pseudo–observations of category k, and the precision α_0 is the total size of the pseudo–dataset.

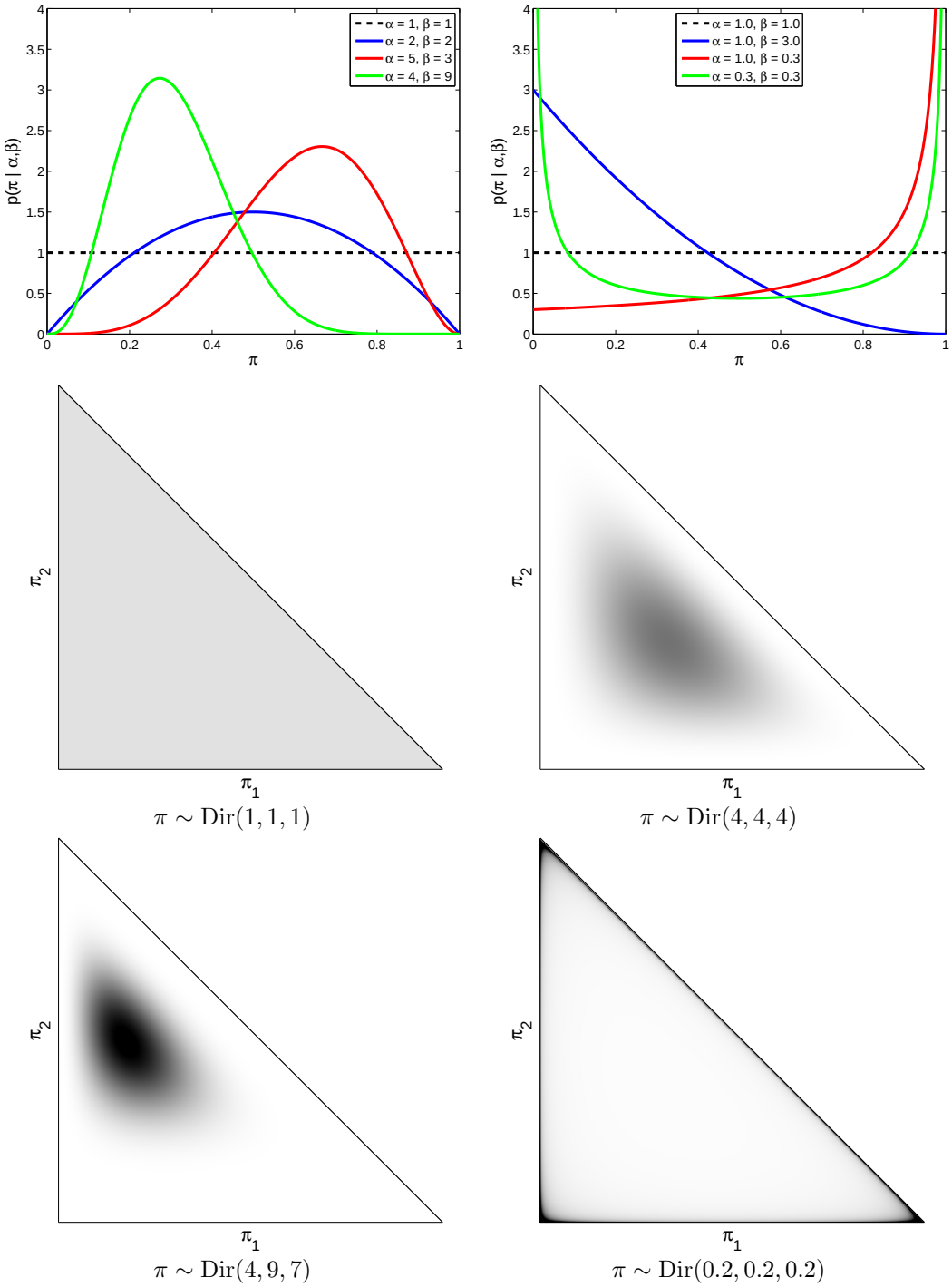


Figure 2.1. Examples of beta and Dirichlet distributions. *Top:* Beta densities with large hyperparameters are unimodal (left), while small values favor biased binomial distributions (right). *Bottom:* Dirichlet densities on $K = 3$ categories, visualized on the simplex $\Pi_2 = (\pi_1, \pi_2, 1 - \pi_1 - \pi_2)$. We show a uniform prior, an unbiased unimodal prior, a biased prior with larger precision α_0, and a prior favoring sparse multinomial distributions. Darker intensities indicate regions with higher probability.

As discussed previously, the predictive likelihood of future observations $\bar{x}$ (as in eq. (2.19)) is often of interest. Using the Dirichlet normalization constant of eq. (2.39) and cancelling terms, it can be shown that

$$p(\bar{x} = k \mid x^{(1)}, \ldots, x^{(L)}, \alpha) = \frac{C_k + \alpha_k}{L + \alpha_0} \tag{2.46}$$

Note that C_k is the number of times category k was observed in the previous L observations (excluding $\bar{x}$). Importantly, these observation counts provide easily updated sufficient statistics which allow rapid predictive likelihood evaluation. Comparing this prediction to that of eq. (2.38), we see that the raw frequencies underlying the ML estimate have been *smoothed* by the pseudo–counts contributed by the Dirichlet prior. More generally, Prop. 2.1.4 can be used to express the likelihood of multiple observations as a ratio gamma functions [123].

■ 2.1.4 Normal–Inverse–Wishart Analysis of Gaussian Observations

Consider a continuous–valued random variable x taking values in d–dimensional Euclidean space $\mathcal{X} = \mathbb{R}^d$. A *Gaussian* or *normal* distribution [21, 107, 229] with mean μ and covariance matrix Λ then has the following form:

$$p(x \mid \mu, \Lambda) = \frac{1}{(2\pi)^{d/2}|\Lambda|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Lambda^{-1}(x-\mu)\right\} \tag{2.47}$$

This distribution, which we denote by $\mathcal{N}(\mu, \Lambda)$, is normalizable if and only if Λ is positive definite. Given L independent Gaussian observations $\{x^{(\ell)}\}_{\ell=1}^L$, their joint likelihood is

$$p(x^{(1)}, \ldots, x^{(L)} \mid \mu, \Lambda) \propto |\Lambda|^{-L/2} \exp\left\{-\frac{1}{2}\sum_{\ell=1}^{L}(x^{(\ell)}-\mu)^T\Lambda^{-1}(x^{(\ell)}-\mu)\right\} \tag{2.48}$$

The maximum likelihood estimates of the Gaussian's parameters, based on this data, are the sample mean and covariance:

$$\hat{\mu} = \frac{1}{L}\sum_{\ell=1}^{L} x^{(\ell)} \qquad\qquad \hat{\Lambda} = \frac{1}{L}\sum_{\ell=1}^{L}(x^{(\ell)}-\hat{\mu})(x^{(\ell)}-\hat{\mu})^T \tag{2.49}$$

Expanding the quadratic form of eq. (2.47), we see that Gaussian densities define a regular exponential family, with canonical parameters proportional to the Gaussian's *information* parameterization $(\Lambda^{-1}, \Lambda^{-1}\mu)$. The sample mean and covariance, or equivalently sums of the observations and their outer products, provide sufficient statistics.

Gaussian Inference

Suppose that x and y are two jointly Gaussian random vectors, with distribution

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Lambda_x & \Lambda_{xy} \\ \Lambda_{yx} & \Lambda_y \end{bmatrix}\right) \tag{2.50}$$

Assuming a fixed covariance, the conjugate prior for a Gaussian's mean is another Gaussian. The conditional distribution of x given y is thus also Gaussian [107, 167, 229], with mean $\widehat{x}$ and covariance $\widehat{\Lambda}_x$ given by the *normal equations*:

$$\widehat{x} = \mu_x + \Lambda_{xy}{\Lambda_y}^{-1}(y - \mu_y) \tag{2.51}$$

$$\widehat{\Lambda}_x = \Lambda_x - \Lambda_{xy}{\Lambda_y}^{-1}\Lambda_{yx} \tag{2.52}$$

The conditional mean $\widehat{x}$ is the *linear least squares estimate* minimizing the mean squared error $\mathbb{E}\big[(x - \widehat{x})^2 \mid y\big]$, while the *error covariance* matrix $\widehat{\Lambda}_x$ measures the reliability of $\widehat{x}$. Note that for Gaussian densities, $\widehat{\Lambda}_x$ is not a function of the observed vector y, but does depend on the joint statistics of x and y.

In many problem domains, the observations y are naturally expressed as a noisy linear function of the latent variables x:

$$y = Cx + v \qquad\qquad v \sim \mathcal{N}(\mu_v, \Lambda_v) \tag{2.53}$$

Assuming x and v are independent, the normal equations then become

$$\widehat{x} = \mu_x + \Lambda_x C^T\big(C\Lambda_x C^T + \Lambda_v\big)^{-1}(y - (C\mu_x + \mu_v)) \tag{2.54}$$

$$\widehat{\Lambda}_x = \Lambda_x - \Lambda_x C^T\big(C\Lambda_x C^T + \Lambda_v\big)^{-1}C\Lambda_x \tag{2.55}$$

Often, these equations are more conveniently expressed in an alternative *information form*. Assuming Λ_x and Λ_v are both positive definite, the matrix inversion lemma [130] allows eqs. (2.54, 2.55) to be rewritten as follows:

$$\widehat{\Lambda}_x^{-1}\widehat{x} = {\Lambda_x}^{-1}\mu_x + C^T{\Lambda_v}^{-1}(y - \mu_v) \tag{2.56}$$

$$\widehat{\Lambda}_x^{-1} = {\Lambda_x}^{-1} + C^T{\Lambda_v}^{-1}C \tag{2.57}$$

This information form plays an important role in the development of tractable computational methods for Gaussian graphical models (see Sec. 2.2.2).

Normal–Inverse–Wishart Distributions

Any distribution satisfying certain spherical symmetries has a representation as a continuous mixture of Gaussian densities, for some prior on that Gaussian's covariance matrix [21, Sec. 4.4]. The conjugate prior for the covariance matrix of a Gaussian distribution with known mean is the *inverse–Wishart* distribution [107], a multivariate generalization of the scaled inverse–χ^2 density. The d–dimensional inverse–Wishart density, with covariance parameter Δ and ν degrees of freedom,[2] equals

$$p(\Lambda \mid \nu, \Delta) \propto |\Lambda|^{-\left(\frac{\nu+d+1}{2}\right)} \exp\left\{-\frac{1}{2}\operatorname{tr}(\nu\Delta\Lambda^{-1})\right\} \tag{2.58}$$

[2] In some texts [107], inverse–Wishart distributions are instead parameterized by a scale matrix $\nu\Delta$.

We denote this density by $\mathcal{W}(\nu, \Delta)$. An inverse–Wishart prior is proper when $\nu > d$, and skewed towards larger covariances, so that its mean and mode equal

$$\mathbb{E}_\nu[\Lambda] = \frac{\nu}{\nu - d - 1}\Delta \qquad \nu > d + 1 \tag{2.59}$$

$$\arg\max_{\Lambda} \mathcal{W}(\Lambda; \nu, \Delta) = \frac{\nu}{\nu + d + 1}\Delta \tag{2.60}$$

The degrees of freedom ν acts as a precision parameter, and can be interpreted as the size of a pseudo–dataset with sample covariance Δ. However, because the inverse–Wishart density is rotationally invariant, it cannot model situations in which the degree of prior knowledge varies across different covariance entries or subspaces. Inverse–Wishart samples can be drawn via appropriate transformations of standard Gaussian random variables [107].

If a multivariate Gaussian's mean and covariance are both uncertain, the *normal–inverse–Wishart* distribution [107] provides an appropriate conjugate prior. Following eq. (2.58), the covariance matrix is assigned an inverse–Wishart prior $\Lambda \sim \mathcal{W}(\nu, \Delta)$. Conditioned on Λ, the mean $\mu \sim \mathcal{N}(\vartheta, \Lambda/\kappa)$. Here, ϑ is the expected mean, for which we have κ pseudo–observations on the scale of observations $x \sim \mathcal{N}(\mu, \Lambda)$. The joint prior distribution, denoted by $\mathcal{NW}(\kappa, \vartheta, \nu, \Delta)$, then takes the following form:

$$p(\mu, \Lambda \mid \kappa, \vartheta, \nu, \Delta) \propto |\Lambda|^{-\left(\frac{\nu+d}{2}+1\right)} \exp\left\{-\frac{1}{2}\operatorname{tr}(\nu\Delta\Lambda^{-1}) - \frac{\kappa}{2}(\mu - \vartheta)^T\Lambda^{-1}(\mu - \vartheta)\right\} \tag{2.61}$$

Fig. 2.2 illustrates a normal–inverse–χ^2 density, the special case arising when $d = 1$. Note that the mean and variance are dependent, so that there is greater uncertainty in the mean value for larger underlying variances. This scaling is often, but not always, appropriate, and is necessary if conjugacy is desired [107]. Fig. 2.2 also shows several Gaussian distributions drawn from a two–dimensional normal–inverse–Wishart prior.

Conjugate Posteriors and Predictions

Consider a set of L observations $\{x^{(\ell)}\}_{\ell=1}^{L}$ from a multivariate Gaussian distribution $\mathcal{N}(\mu, \Lambda)$ with normal–inverse–Wishart prior $\mathcal{NW}(\kappa, \vartheta, \nu, \Delta)$. Via conjugacy, the posterior distribution $p\big(\mu, \Lambda \mid x^{(1)}, \ldots, x^{(\ell)}, \kappa, \vartheta, \nu, \Delta\big)$ is also normal–inverse–Wishart, and thus compactly described by a set of updated hyperparameters $\mathcal{NW}\big(\bar{\kappa}, \bar{\vartheta}, \bar{\nu}, \bar{\Delta}\big)$. Through manipulation of the quadratic form in eq. (2.61), it can be shown [107] that these posterior hyperparameters equal

$$\bar{\kappa}\bar{\vartheta} = \kappa\vartheta + \sum_{\ell=1}^{L} x^{(\ell)} \qquad \bar{\kappa} = \kappa + L \tag{2.62}$$

$$\bar{\nu}\bar{\Delta} = \nu\Delta + \sum_{\ell=1}^{L} x^{(\ell)}x^{(\ell)^T} + \kappa\vartheta\vartheta^T - \bar{\kappa}\bar{\vartheta}\bar{\vartheta}^T \qquad \bar{\nu} = \nu + L \tag{2.63}$$

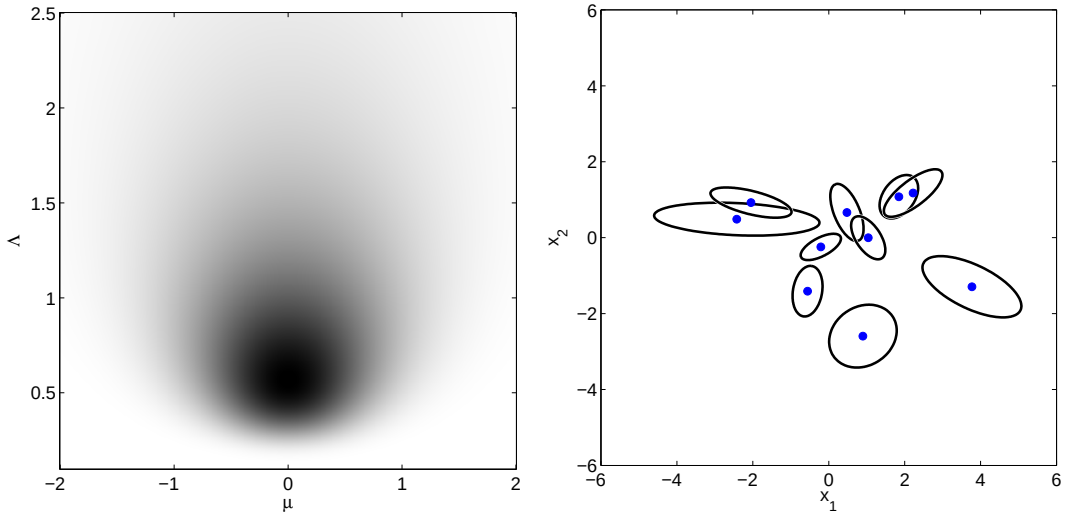


Figure 2.2. Examples of normal–inverse–Wishart distributions. *Left:* Joint probability density of a scalar normal–inverse–χ^2 distribution $(\mu, \Lambda) \sim \mathcal{NW}(2, 0, 4, 1)$. *Right:* Covariance ellipses corresponding to ten samples from a two–dimensional normal–inverse–Wishart distribution $(\mu, \Lambda) \sim \mathcal{NW}(0.3, 0, 4, I_2)$.

To efficiently represent these posterior parameters, we can cache the observations' sum (eq. (2.62)), and the *Cholesky decomposition* [63, 118] of the sum of observation outer products (eq. (2.63)). Cholesky decompositions are numerically robust, can be recursively updated as observations are added or removed, and allow fast likelihood evaluation through the solution of triangulated linear systems.

Integrating over the parameters of the normal–inverse–Wishart posterior distribution, the predictive likelihood of a new observation $\bar{x}$ is multivariate Student–t with $(\bar{\nu} - d + 1)$ degrees of freedom [107]. Assuming $\bar{\nu} > (d + 1)$, this posterior density has finite covariance, and can be approximated by a moment–matched Gaussian:

$$p(\bar{x} \mid x^{(1)}, \ldots, x^{(L)}, \kappa, \vartheta, \nu, \Delta) \approx \mathcal{N}\left(\bar{x}; \bar{\vartheta}, \frac{(\bar{\kappa} + 1)\bar{\nu}}{\bar{\kappa}(\bar{\nu} - d - 1)} \bar{\Delta}\right) \tag{2.64}$$

As illustrated in Fig. 2.3, Student–t distributions have heavier tails than Gaussians, due to integration over uncertainty in the true covariance. However, the KL divergence plot of Fig. 2.3 shows that, for small d, the Gaussian approximation is accurate unless $\bar{\nu}$ is very small. Examining eqs. (2.62, 2.63), we see that the predictive likelihood depends on regularized estimates of the mean and covariance of previous observations.

■ 2.2 Graphical Models

Many practical applications, including the computer vision tasks investigated in this thesis, involve very large collections of random variables. In these situations, direct application of the classic exponential families introduced in the previous section is typically infeasible. For example, a multinomial model of the joint distribution of

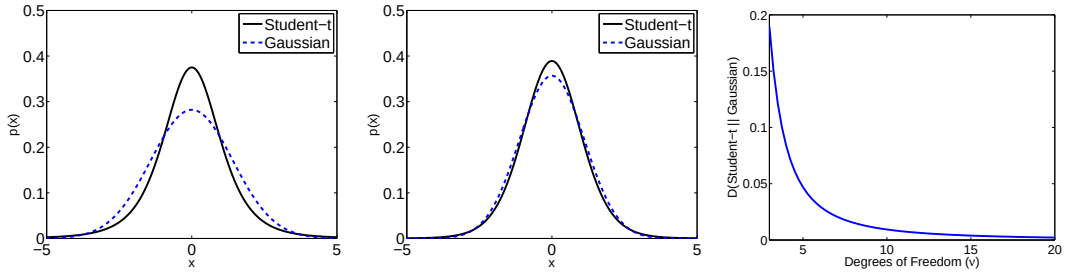


Figure 2.3. Approximation of Student–t predictive distributions by a Gaussian with moments matched as in eq. (2.64). We compare one–dimensional Gaussian and heavier–tailed Student–t densities with $\nu = 4$ (left) and $\nu = 10$ (center) degrees of freedom. For moderate ν, the Gaussian approximation becomes very accurate (see plot of KL divergence versus ν, right).

100 binary variables has $2^{100} \approx 10^{30}$ parameters. Even if such a density could be stored and manipulated, reliable parameter estimation would require an unrealistically massive dataset. Similarly, in fields such as image processing [85, 95, 189, 285] and oceanography [86, 330], estimation of random fields containing millions of continuous variables is not uncommon. However, explicit computations with large, unstructured covariance matrices are extremely difficult [63], typically requiring specialized, parallel hardware.

Probabilistic *graphical models* provide a powerful, flexible framework which addresses these concerns [40, 50, 159, 177, 231, 249, 311, 339]. Graphs are used to decompose multivariate, joint distributions into a set of local interactions among small subsets of variables. These local relationships produce conditional independencies which lead to efficient learning and inference algorithms. Moreover, their modular structure provides an intuitive language for expressing domain–specific knowledge about variable relationships, and facilitates the transfer of modeling advances to new applications.

In the following sections, we introduce and compare several different families of graphical models, including directed Bayesian networks, undirected Markov random fields, and factor graphs. We then relate these models to classic notions of exchangeability, motivating a family of *hierarchical* models used extensively in this thesis.

2.2.1 Brief Review of Graph Theory

We begin by reviewing definitions from graph theory which are useful in describing graphical models. For more detailed surveys of these concepts, see [50, 177].

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of *nodes* or *vertices* $\mathcal{V}$, and a corresponding set of *edges* $\mathcal{E}$. Each edge $(i, j) \in \mathcal{E}$ connects two distinct nodes $i, j \in \mathcal{V}$. For *directed graphs*, an edge (i, j) connects a *parent* vertex i to its *child* j, and is pictorially represented by an arrow (see Fig. 2.4(a)). The set of all parents $\Gamma(j)$ of node j is then given by

$$\Gamma(j) \triangleq \{i \in \mathcal{V} \mid (i, j) \in \mathcal{E}\} \tag{2.65}$$

In *undirected graphs*, an edge $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$, as depicted by an

arrowless line (see Fig. 2.4(c)). For such graphs, $\Gamma(j)$ are known as the *neighbors* of node j, since $i \in \Gamma(j)$ whenever $j \in \Gamma(i)$. It is also possible to define *chain graphs* which mix undirected and directed edges [37, 50, 177], but we do not use them in this thesis. Within any graph, a *clique* is a set of nodes for which all pairs are connected by an edge. If the entire graph forms a clique, it is said to be *complete*.

When describing the statistical properties of graphical models, the structural properties of the underlying graph play an important role. A *path* between nodes $i_0 \neq i_T$ is a sequence of distinct nodes $(i_0, i_1, \ldots, i_T)$ such that $(i_{\ell-1}, i_\ell) \in \mathcal{E}$ for $\ell = 1, \ldots, T$. A *cycle*, or loop,[3] is a path which starts and ends with the same node $i_0 = i_T$, and for which all internal nodes $(i_1, \ldots, i_{T-1})$ are distinct. If there is a path (in either direction) between every pair of nodes, $\mathcal{G}$ is *connected*. If an edge joins two non–consecutive vertices within some cycle, it is called a *chord*. When the undirected version of $\mathcal{G}$ (obtained by replacing all directed edges with undirected ones) has no cycles, the graph is *tree–structured*. Within any tree, a *leaf* node has at most one neighbor. Note that it is easy to construct acyclic, directed graphs which are *not* trees. For any graph, the *diameter* equals the number of edges in the longest path between any two nodes.

Hypergraphs extend graphs by introducing *hyperedges* connecting subsets with more than two vertices [177]. We denote a hypergraph by $\mathcal{H} = (\mathcal{V}, \mathcal{F})$, where $\mathcal{V}$ are vertices as before, and each hyperedge $f \in \mathcal{F}$ is some subset of those vertices ($f \subset \mathcal{V}$). Pictorially, we represent hypergraphs by *bipartite graphs* with circular nodes for each vertex $i \in \mathcal{V}$, and square nodes for each hyperedge $f \in \mathcal{F}$ (see Fig. 2.4(b)). Lines are then used to connect hyperedge nodes to their associated vertex set [175].

■ 2.2.2 Undirected Graphical Models

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ or hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{F})$, graphical models represent probability distributions by associating each node $i \in \mathcal{V}$ with a random variable $x_i \in \mathcal{X}_i$. The structure of the joint distribution $p(x)$, where $x \triangleq \{x_i \mid i \in \mathcal{V}\}$ takes values in the joint sample space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_N$, is then determined by the corresponding (hyper)edges. In this section, we introduce three closely related families of graphical models which use edges to encode local, probabilistic *constraints*.

Factor Graphs

Hypergraphs $\mathcal{H} = (\mathcal{V}, \mathcal{F})$ provide an intuitive means of describing probability distributions $p(x)$. For any $f \in \mathcal{F}$, let $x_f \triangleq \{x_i \mid i \in f\}$ denote the corresponding set of random variables. A *factor graph* then defines the joint distribution as a normalized product of local *potential functions* defined on these hyperedges:

$$p(x) \propto \prod_{f \in \mathcal{F}} \psi_f(x_f) \tag{2.66}$$

[3] In graph theoretic terminology, a loop is an edge connecting a node to itself [177]. However, as graphical models do not have self–connections, in this thesis we use the terms loop and cycle interchangeably, as is standard in the graphical inference literature [219, 319].

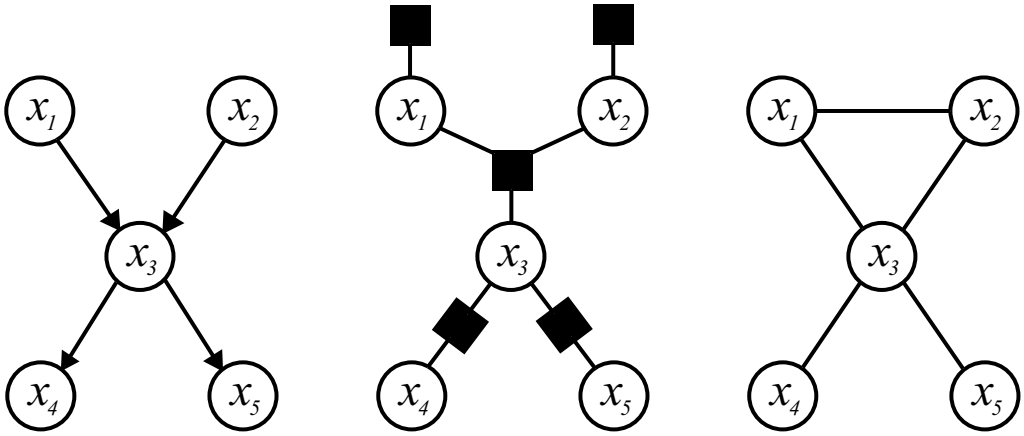


Figure 2.4. Three graphical representations of a distribution over five random variables (see [175]). (a) Directed graph $\mathcal{G}$ depicting a causal, generative process. (b) Factor graph expressing the factorization underlying $\mathcal{G}$. (c) A "moralized" undirected graph capturing the Markov structure of $\mathcal{G}$.

For example, in the factor graph of Fig. 2.5(c), there are 5 variable nodes, and the joint distribution has one potential for each of the 3 hyperedges:

$$p(x) \propto \psi_{123}(x_1, x_2, x_3)\, \psi_{234}(x_2, x_3, x_4)\, \psi_{35}(x_3, x_5)$$

Often, these potentials can be interpreted as local dependencies or constraints. Note, however, that $\psi_f(x_f)$ does *not* typically correspond to the marginal distribution $p_f(x_f)$, due to interactions with the graph's other potentials.

In many applications, factor graphs are used to impose structure on an exponential family of densities. In particular, suppose that each potential function is described by the following unnormalized exponential form:

$$\psi_f(x_f \mid \theta_f) = \nu_f(x_f) \exp\left\{ \sum_{a \in \mathcal{A}_f} \theta_{fa} \phi_{fa}(x_f) \right\} \tag{2.67}$$

Here, $\theta_f \triangleq \{\theta_{fa} \mid a \in \mathcal{A}_f\}$ are the canonical parameters of the *local* exponential family for hyperedge f. From eq. (2.66), the joint distribution can then be written as

$$p(x \mid \theta) = \left(\prod_{f \in \mathcal{F}} \nu_f(x_f) \right) \exp\left\{ \sum_{f \in \mathcal{F}} \sum_{a \in \mathcal{A}_f} \theta_{fa} \phi_{fa}(x_f) - \Phi(\theta) \right\} \tag{2.68}$$

Comparing to eq. (2.1), we see that factor graphs define regular exponential families [104, 311], with parameters $\theta = \{\theta_f \mid f \in \mathcal{F}\}$, whenever local potentials are chosen from such families. The results of Sec. 2.1 then show that *local* statistics, computed over the support of each hyperedge, are sufficient for learning from training data. This

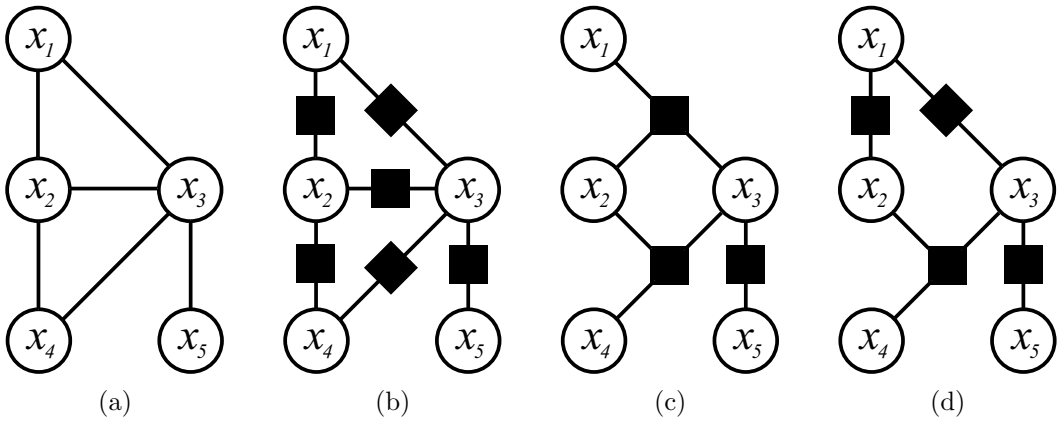


Figure 2.5. An undirected graphical model, and three factor graphs with equivalent Markov properties. (a) Undirected graph $\mathcal{G}$ representing five random variables. (b) Factor graph interpreting $\mathcal{G}$ as a pairwise MRF. (c) Factor graph corresponding to the maximal cliques of $\mathcal{G}$. (d) Another possible factorization which is Markov with respect to $\mathcal{G}$. In all cases, single-node factors are omitted for clarity.

guarantee can be extremely useful for large graphs with many variables. Note, however, that interactions among overlapping potential functions induce global dependencies in the parameters. Thus, as we discuss in more detail in Sec. 2.3, learning can be computationally difficult even when the potentials take simple forms.

Many widely used graphical models correspond to a particular choice of the exponential families in eq. (2.68). For example, any distribution on discrete spaces $\mathcal{X}_i = \{1, \dots, K_i\}$ can be expressed in terms of a set of *indicator* potential functions which enumerate all possible configurations of the variables within each factor [311]. Alternatively, jointly Gaussian random fields take potentials to be local quadratic functions. The graph structure of these *covariance selection* models is then expressed via an inverse covariance matrix which is *sparse*, with many entries equaling zero [64, 177, 268, 276].

The exponential family representation of eq. (2.68) is convenient for learning and parameter estimation. In many applications, however, a model has already been determined (perhaps via MAP estimation as in Sec. 2.1.2), and we are instead interested in *inference* problems. In such cases, we prefer the representation of eq. (2.66), since it highlights the factorization underlying efficient computational methods.

Markov Random Fields

Undirected graphical models, or *Markov random fields (MRFs)*, characterize distributions $p(x)$ via a set of implied conditional independencies. In this section, we describe these Markov properties, and relate them to an algebraic factorization similar to that underlying factor graphs.

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let f, g and h denote three disjoint subsets of $\mathcal{V}$. Set h is said to *separate* sets f and g if every path between f and g passes through some node in h. A stochastic process x is *globally Markov* with respect to $\mathcal{G}$ if x_f and

x_g are independent conditioned on the variables x_h in any separating set:

$$p(x_f, x_g \mid x_h) = p(x_f \mid x_h)\, p(x_g \mid x_h) \qquad \text{if } h \text{ separates } f \text{ from } g \tag{2.69}$$

This property generalizes temporal Markov processes, for which the past and future are independent conditioned on the present. For example, the undirected graph of Fig. 2.5(a) implies the following conditional independencies, among others:

$$p(x_1, x_2, x_5 \mid x_3, x_4) = p(x_1, x_2 \mid x_3, x_4)\, p(x_5 \mid x_3)$$
$$p(x_1, x_4, x_5 \mid x_2, x_3) = p(x_1 \mid x_2, x_3)\, p(x_4 \mid x_2, x_3)\, p(x_5 \mid x_3)$$

An important special case of eq. (2.69) guarantees that conditioned on its immediate neighbors, the random variable at any node is independent of the rest of the process:

$$p\big(x_i \mid x_{\mathcal{V}\setminus i}\big) = p\big(x_i \mid x_{\Gamma(i)}\big) \tag{2.70}$$

As we discuss in later sections, this *local Markov property* plays an important role in the design of efficient learning and inference algorithms.

The following theorem, due to Hammersley and Clifford, shows that Markov random fields are naturally parameterized via potential functions defined on the cliques of the corresponding undirected graph.

Theorem 2.2.1 (Hammersley-Clifford). *Let $\mathcal{C}$ denote the set of cliques of an undirected graph $\mathcal{G}$. A probability distribution defined as a normalized product of nonnegative potential functions on those cliques is then always Markov with respect to $\mathcal{G}$:*

$$p(x) \propto \prod_{c \in \mathcal{C}} \psi_c(x_c) \tag{2.71}$$

Conversely, any strictly positive density ($p(x) > 0$ for all x) which is Markov with respect to $\mathcal{G}$ can be represented in this factored form.

Proof. There are a variety of ways to prove this result; see [26, 35, 43] for examples and further discussion. For a degenerate Markov distribution which cannot be factored as in eq. (2.71), see Lauritzen [177]. □

Comparing eq. (2.71) to eq. (2.66), we see that Markov random fields can always be represented by a factor graph with one hyperedge for each of the graph's cliques [175, 339]. This representation is also known as the *clique hypergraph* corresponding to $\mathcal{G}$ [177]. Note that it is possible, but not necessary, to restrict this factorization to *maximal cliques* which are not a strict subset of any other clique (see Fig. 2.5(c)).

In practice, Markov properties are used in two complementary ways. If a stochastic process is known to satisfy certain conditional independencies, the Hammersley–Clifford Theorem then motivates models parameterized by local sufficient statistics. Conversely,

given any graphical model, the implied Markov properties can be exploited to design more efficient learning and inference algorithms.

While undirected graphs fully specify a probability density's Markov structure, they do not unambigously determine that density's factorization into potential functions. For example, in Fig. 2.5 we show three different factor graphs, all of which are Markov with respect to the same undirected graph. Because the differences among these factorizations have implications for learning and inference, the more detailed factor graph representation is often preferable [96, 98, 175].

Pairwise Markov Random Fields

In many applications, it is convenient to consider a restricted class of *pairwise Markov random fields*. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a pairwise MRF expresses the joint distribution as a product of potential functions defined on that graph's edges:

$$p(x) \propto \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} \psi_i(x_i) \tag{2.72}$$

Because pairs of neighboring nodes always define cliques, the Hammersley–Clifford Theorem guarantees that pairwise MRFs are Markov with respect to $\mathcal{G}$. The inclusion of single–node potentials $\psi_i(x_i)$ is not strictly necessary, but is often convenient. Pairwise MRFs containing only binary variables are known as *Ising models* in the statistical physics literature [337].

Fig. 2.5(b) shows the factor graph corresponding to a pairwise MRF, and contrasts it with models incorporating higher order cliques. To avoid ambiguities, in this thesis we only use undirected graphs to depict pairwise MRFs. For graphical models containing interactions among three or more variables, we instead use a factor graph representation which explicitly reveals the underlying factorization.

Many inference tasks can be posed as the estimation of a set of *latent* or *hidden* variables x based on noisy observations y. In such cases, pairwise MRFs are sometimes used to express the internal structure of the desired posterior distribution:

$$p(x \mid y) = \frac{p(x,y)}{p(y)} \propto \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} \psi_i(x_i, y) \tag{2.73}$$

Frequently, observations also decompose into local measurements $y = \{y_i \mid i \in \mathcal{V}\}$, so that $\psi_i(x_i, y) = \psi_i(x_i, y_i)$. Fig. 2.6 shows two examples of pairwise MRFs used widely in practice: a multiscale tree–structured graph [34, 41, 85, 86, 189, 330], and a nearest–neighbor grid [26, 95, 108, 196, 285]. In both cases, shaded nodes represent noisy local observations y_i.

■ 2.2.3 Directed Bayesian Networks

We now introduce a different family of graphical models derived from directed graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. As before, *Bayesian networks* associate each node $i \in \mathcal{V}$ with a random

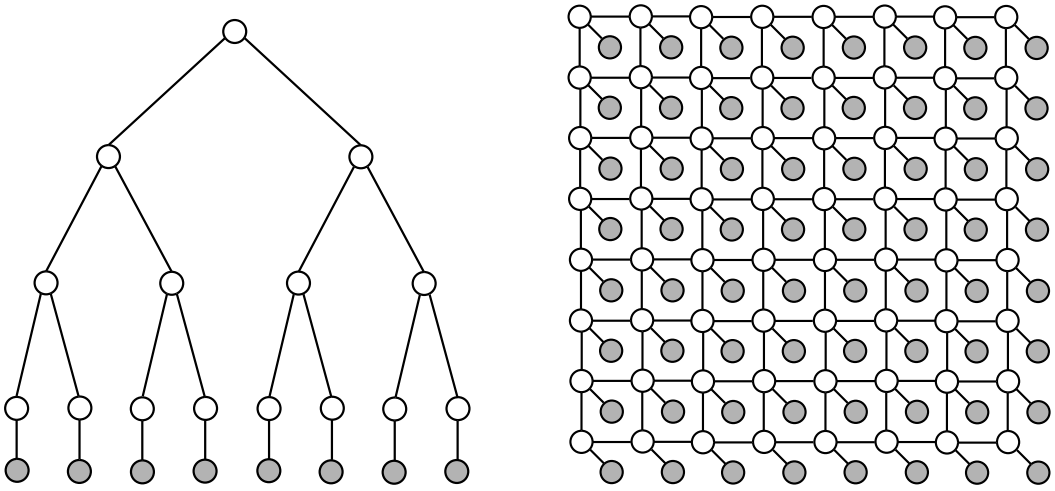


Figure 2.6. Sample pairwise Markov random fields, where open nodes represent hidden variables x_i and shaded nodes are local observations y_i. *Left:* A multiscale tree–structured graph, in which *coarse scale* nodes capture dependencies among an observed *fine scale* process. *Right:* A nearest–neighbor grid in which each hidden variable is connected to its four closest spatial neighbors.

variable x_i. However, in place of potential functions, directed models decompose $p(x)$ via the conditional density of each child node i given its parents $\Gamma(i)$:

$$p(x) = \prod_{i \in \mathcal{V}} p\big(x_i \mid x_{\Gamma(i)}\big) \tag{2.74}$$

For nodes i without parents ($\Gamma(i) = \emptyset$), we define $p\big(x_i \mid x_{\Gamma(i)}\big) = p(x_i)$. This factorization is consistent whenever $\mathcal{G}$ is a directed *acyclic* graph, so that its edges specify a valid partial ordering of the random variables [50, 128, 231]. For example, the directed graph of Fig. 2.4(a) implies the following conditional densities:

$$p(x) = p(x_1)\, p(x_2)\, p(x_3 \mid x_1, x_2)\, p(x_4 \mid x_3)\, p(x_5 \mid x_3)$$

Bayesian networks effectively define a *causal* generative process, beginning with nodes without parents and proceeding from parent to child throughout the graph. In contrast, direct sampling from undirected graphical models is often intractable (see Sec. 2.4).

The Markov properties of directed Bayesian networks are slightly different from those of undirected graphical models. In particular, a random variable x_i is conditionally independent of the remaining process given its parents $x_{\Gamma(i)}$, children $\{x_j \mid i \in \Gamma(j)\}$, and its *children's parents*. These relationships are captured by a corresponding *moral graph* in which parents are connected ("married") by additional undirected edges [50], as in Fig. 2.4(c). Although factor graphs can express this Markov structure (see Fig. 2.4(b)), doing so obscures the underlying causal, generative process. A directed generalization of factor graphs has been proposed [96, 98], but we focus on simpler

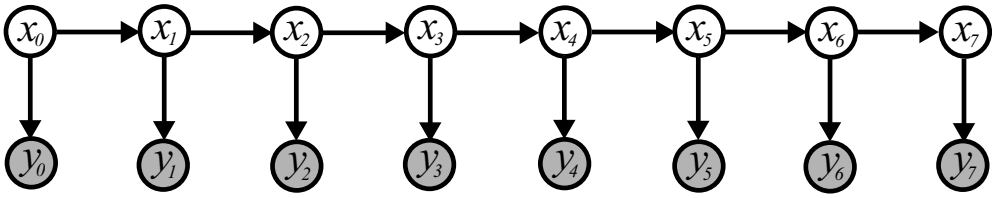


Figure 2.7. Directed graphical representation of a hidden Markov model (HMM) for $T = 8$ samples of a temporal process. The hidden states x_t capture dependencies among the observations y_t.

Bayesian network representations. For a discussion of transformations allowing conversion between undirected, directed, and factor graphs, see [175, 339].

In many applications, exponential families provide convenient parameterizations of the conditional densities composing a Bayesian network. In general, directed models define *curved* exponential families [74], because conditional densities with multiple parents may impose constraints on the set of achievable canonical parameters [104]. However, this subtlety does not arise in the particular models considered by this thesis.

Hidden Markov Models

Directed graphical models provide the basis for a family of *hidden Markov models (HMMs)* which are widely used to model temporal stochastic processes [8, 70, 163, 235]. Let $y = \{y_t\}_{t=0}^{T-1}$ denote observations of a temporal process collected at T discrete time points. We assume that each observation y_t is independently sampled conditioned on an underlying *hidden state* x_t. If we further assume that these states $x = \{x_t\}_{t=0}^{T-1}$ evolve according to a *first–order* temporal Markov process, the joint distribution equals

$$p(x, y) = p(x_0)\, p(y_0 \mid x_0) \prod_{t=1}^{T-1} p(x_t \mid x_{t-1})\, p(y_t \mid x_t) \tag{2.75}$$

Fig. 2.7 shows a directed graphical representation of this density. In later chapters, we extend this model to develop methods for visual tracking of articulated objects.

Historically, models equivalent to HMMs were independently developed in several different domains. For example, in speech recognition the hidden states typically take values on some finite, discrete set, and statistical methods are used to learn dynamics from speech waveforms [235]. In contrast, control theorists often use continuous *state space models* to characterize the position, velocity, and other properties of physical systems [8, 163]. Graphical models unify these disparate approaches, and allow advances in learning and inference methods to be transferred between domains [50, 159, 249].

■ 2.2.4 Model Specification via Exchangeability

In some applications, a graphical model's structure is determined by the physical data generation process. For example, HMMs (see Fig. 2.7) are often derived from a known

dynamical system, while grid–structured MRFs (see Fig. 2.6) can arise from the discretization of stochastic partial differential equations. For other learning tasks, however, the generative process may be unknown, or too complex to characterize explicitly. In this section, we show how simple assumptions about the *indistinguishability* of different observations lead naturally to a family of hierarchical, directed graphical models.

Consider a set of N random variables $\{x_i\}_{i=1}^N$. These variables are said to be *exchangeable* if every permutation, or reordering, of their indices has equal probability:

$$p(x_1, \ldots, x_N) = p\big(x_{\tau(1)}, \ldots, x_{\tau(N)}\big) \qquad \text{for any permutation } \tau(\cdot) \tag{2.76}$$

This expression formalizes the concept of an unordered collection of random variables, for which the chosen indices are purely notational. When no auxiliary information is available, this assumption is usually reasonable. Extending this definition, a sequence $\{x_i\}_{i=1}^\infty$ is *infinitely exchangeable* if every finite subsequence is exchangeable [21, 107].

As shown by the following theorem, exchangeable observations can *always* be represented via a prior distribution over some latent parameter space.

Theorem 2.2.2 (De Finetti). *For any infinitely exchangeable sequence of random variables $\{x_i\}_{i=1}^\infty$, $x_i \in \mathcal{X}$, there exists some space Θ, and corresponding density $p(\theta)$, such that the joint probability of any N observations has a mixture representation:*

$$p(x_1, x_2, \ldots, x_N) = \int_\Theta p(\theta) \prod_{i=1}^N p(x_i \mid \theta)\, d\theta \tag{2.77}$$

When $\mathcal{X}$ is a K–dimensional discrete space, Θ may be chosen as the $(K-1)$–simplex. For Euclidean $\mathcal{X}$, Θ is an infinite–dimensional space of probability measures.

Proof. De Finetti's original proof for binary $\mathcal{X}$ dates to the 1930's; see [127] for a simpler proof of that case, and [21, Sec. 4.5] for generalizations and additional references. □

Technically, the representation of eq. (2.77) is only guaranteed to exist when $\{x_i\}_{i=1}^N$ are part of an infinitely exchangeable sequence. However, for moderate N, the distortion induced by assuming infinite exchangeability, when only finite exchangeability is guaranteed, cannot be significant [21, Prop. 4.19].

De Finetti's theorem is often taken as a justification for Bayesian methods, since the infinite mixture representation of eq. (2.77) corresponds precisely with the marginal likelihood of eq. (2.21). We see that exchangeability does not imply independence of the observations, but *conditional independence* given a set of latent parameters θ. Note also that for continuous sample spaces, these parameters are infinite–dimensional, since there is no finite parameterization for the space of continuous densities. This motivates a class of nonparametric methods which we discuss further in Sec. 2.5.

When applying the representation of eq. (2.77), it is common to assume some family of prior distributions with hyperparameters λ, so that

$$p(x_1, \ldots, x_N, \theta \mid \lambda) = p(\theta \mid \lambda) \prod_{i=1}^N p(x_i \mid \theta) \tag{2.78}$$

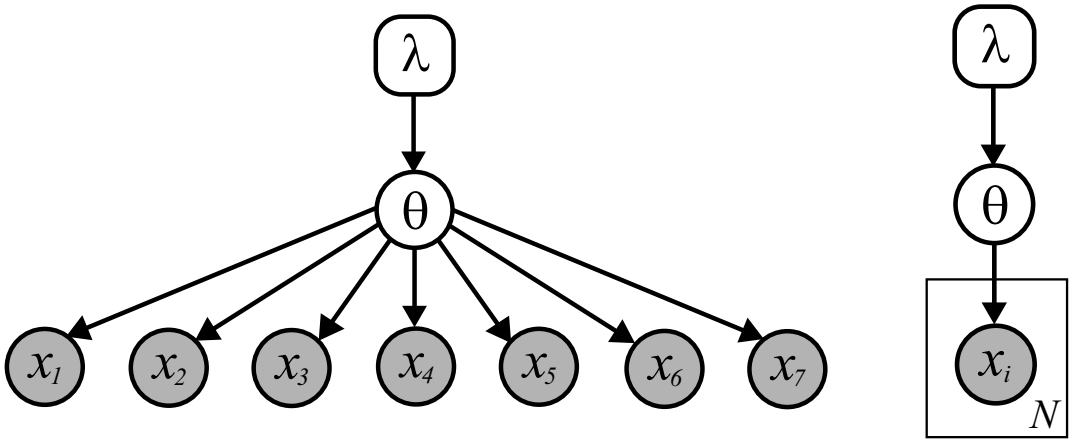


Figure 2.8. De Finetti's representation of N exchangeable random variables $\{x_i\}_{i=1}^N$ as a hierarchical model. Each observation is independently sampled from a density with parameters θ, which are in turn assigned a prior distribution with hyperparameters λ. *Left:* Explicit model for $N = 7$ variables. *Right:* Compact plate representation of the N–fold replication of the observations x_i.

This generative process can be described by the Bayesian network of Fig. 2.8, where *plates* are used to compactly denote replicated variables [37, 159]. In Bayesian statistics, this is known as a *hierarchical model* [21, 107] due to the layering by which observations depend on parameters, which are in turn related to hyperparameters. Note that we have explicitly included the parameters θ and hyperparameters λ (depicted by a rounded box) in the graphical structure. While not strictly necessary, this approach is often useful in learning problems where the parameters are of particular interest [50].

Finite Exponential Family Mixtures

Standard exponential family densities can be too inflexible to accurately describe many complex, multimodal datasets. In these situations, data are often modeled via a finite *mixture distribution* [107, 203, 239, 249]. A K component mixture model takes the following general form:

$$p(x \mid \pi, \theta_1, \ldots, \theta_K) = \sum_{k=1}^{K} \pi_k f(x \mid \theta_k) \qquad \pi \in \Pi_{K-1} \tag{2.79}$$

Each mixture component, or *cluster*, belongs to a parameterized family of probability densities $f(x \mid \theta)$, whose distribution we equivalently denote by $F(\theta)$. Each data point x_i is generated by independently selecting one of K clusters according to the multinomial distribution π, and then sampling from the chosen cluster's data distribution:

$$\begin{aligned} z_i &\sim \pi \\ x_i &\sim F(\theta_{z_i}) \end{aligned} \tag{2.80}$$

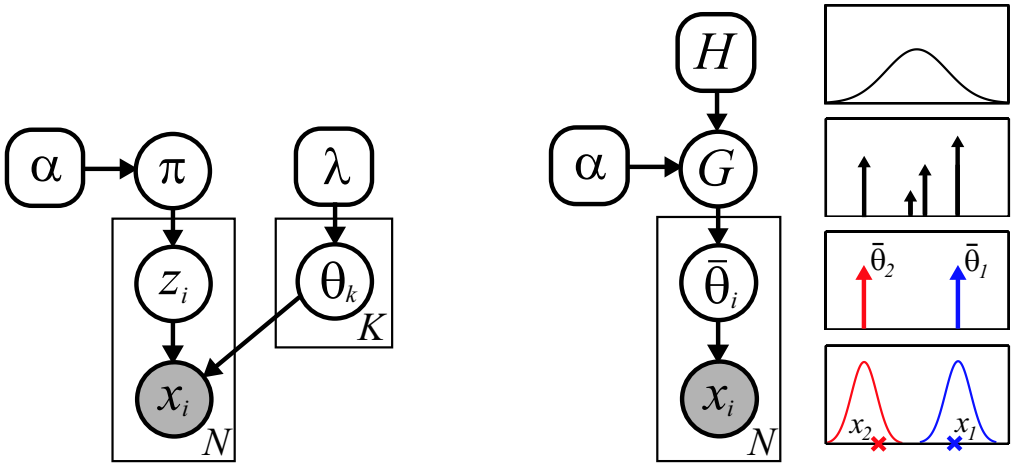


Figure 2.9. Directed graphical representations of a K component mixture model. Mixture weights $\pi \sim \mathrm{Dir}(\alpha)$, while cluster parameters are assigned independent priors $\theta_k \sim H(\lambda)$. *Left:* Indicator variable representation, in which $z_i \sim \pi$ is the cluster that generates $x_i \sim F(\theta_{z_i})$. *Right:* Alternative distributional form, in which G is a discrete distribution on Θ taking K distinct values. $\bar{\theta}_i \sim G$ are the parameters of the cluster that generates $x_i \sim F(\bar{\theta}_i)$. We illustrate with a mixture of $K = 4$ Gaussians, where cluster variances are known (bottom) and $H(\lambda)$ is a Gaussian prior on cluster means (top). Sampled cluster means $\bar{\theta}_1, \bar{\theta}_2$, and corresponding Gaussians, are shown for two observations x_1, x_2.

The unobserved *indicator variable* $z_i \in \{1, \ldots, K\}$ specifies the unique cluster associated with x_i. Mixture models are widely used for *unsupervised learning*, where clusters are used to discover subsets of the data with common attributes.

In most applications of mixture models, $f(x \mid \theta_k)$ is chosen to be an appropriate exponential family. For example, Euclidean observations are often modeled via Gaussian mixtures, so that the parameters $\theta_k = (\mu_k, \Lambda_k)$ specify each cluster's mean μ_k and covariance Λ_k. When learning mixtures from data, it is often useful to place an independent conjugate prior H, with hyperparameters λ, on each cluster's parameters:

$$\theta_k \sim H(\lambda) \qquad\qquad k = 1, \ldots, K \tag{2.81}$$

Similarly, in the absence of prior knowledge distinguishing the clusters, the mixture weights π can be assigned a symmetric Dirichlet prior with precision α:

$$\pi \sim \mathrm{Dir}\Big(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\Big) \tag{2.82}$$

Fig. 2.9 shows a directed graphical model summarizing this generative process. As in Fig. 2.8, plates are used to compactly denote the K cluster parameters $\{\theta_k\}_{k=1}^K$ and N data points $\{x_i\}_{i=1}^N$. In Fig. 2.10, we illustrate several two–dimensional Gaussian mixtures sampled from a conjugate, normal–inverse–Wishart prior.

Mixture models can equivalently be expressed in terms of a discrete distribution G

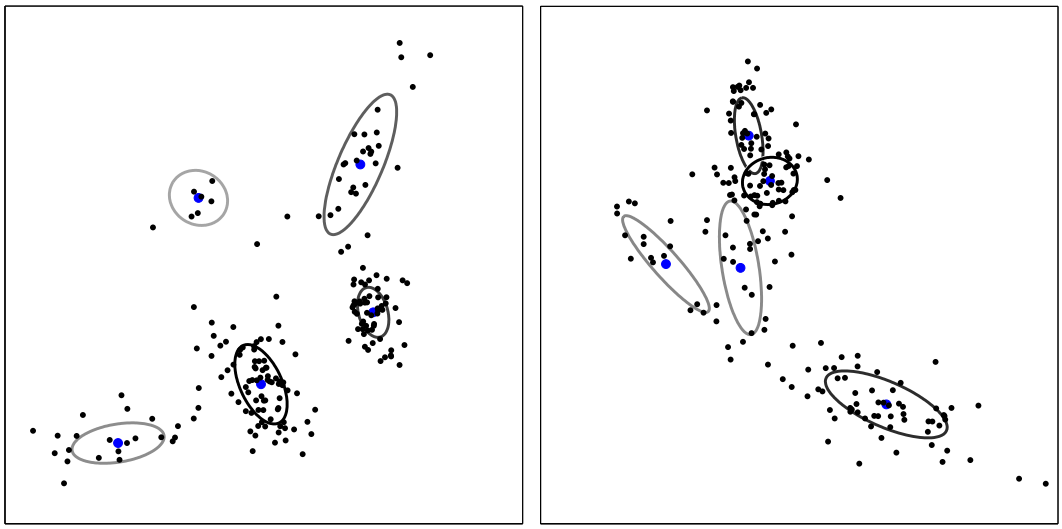


Figure 2.10. Two randomly sampled mixtures of $K = 5$ two–dimensional Gaussians. Mixture parameters are generated from conjugate, normal–inverse–Wishart priors. For each mixture, we plot one standard deviation covariance ellipses Λ_k with intensity proportional to their probability $\pi \sim \mathrm{Dir}(\alpha_0)$, $\alpha_0 = 10$. In each case, we also show $N = 200$ randomly sampled observations.

on the space Θ of cluster parameters:

$$G(\theta) = \sum_{k=1}^{K} \pi_k \delta(\theta, \theta_k) \qquad \begin{aligned} \pi &\sim \mathrm{Dir}(\alpha) \\ \theta_k &\sim H(\lambda) \qquad k = 1, \ldots, K \end{aligned} \tag{2.83}$$

We generate each data point x_i by sampling a set of parameters $\bar{\theta}_i$ from G:

$$\begin{aligned} \bar{\theta}_i &\sim G \\ x_i &\sim F\big(\bar{\theta}_i\big) \end{aligned} \tag{2.84}$$

This representation, which is statistically equivalent to the indicator variables used in eq. (2.80), plays an important role in later hierarchical extensions. Note that G can be seen as a discrete, K component approximation to the infinite–dimensional measure arising in De Finetti's Theorem. Fig. 2.9 shows a graphical representation of this alternative form, and illustrates the generative process for a simple one–dimensional Gaussian mixture with known variance.

The mixture models of Fig. 2.9 assume the number of clusters K to be a fixed, known constant. In general, determining an appropriate mixture size is a difficult problem, which has motivated a wide range of model selection procedures [46, 87, 203, 314]. In Sec. 2.5, we discuss an alternative nonparametric approach which controls complexity by placing prior distributions on *infinite* mixtures.

Analysis of Grouped Data: Latent Dirichlet Allocation

In many domains, there are several *groups* of data which are thought to be produced by related, but distinct, generative processes. For example, medical studies often combine data collected at multiple sites, which examine a common treatment but may have location–specific idiosyncrasies [75, 107]. In text analysis, the words composing a text corpus are typically separated into different documents [31, 123, 140, 289]. Similarly, computer vision systems like those developed in this thesis learn appearance models from visual features detected in different training images [14, 79, 81, 266, 280, 282].

While it is simplest to analyze each group independently, doing so neglects critical information when groups are individually ambiguous. Conversely, combining groups in a single exchangeable dataset may lead to inappropriately biased estimates, and obscures features distinguishing particular groups. By *sharing* random parameters among groups, hierarchical Bayesian models provide an elegant compromise [21, 107, 216]. Posterior dependencies between parameters then effectively transfer information between related experiments, documents, or objects. Estimates based on these distributions are "shrunk" together, so that groups share the strength of other datasets while retaining distinctive features. For example, the classic *James–Stein estimator*, which uniformly dominates the ML estimate of a multivariate Gaussian's mean, can be derived via an empirical Bayesian analysis of a particular hierarchical model [75].

Latent Dirichlet allocation (LDA) [31] extends mixture models (as in Fig. 2.9) to learn clusters describing several related sets of observations. Given J groups of data, let $\mathbf{x}_j = (x_{j1}, \ldots, x_{jN_j})$ denote the N_j data points in group j, and $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_J)$. LDA assumes that the data within each group are exchangeable, and independently sampled from one of K latent clusters with parameters $\{\theta_k\}_{k=1}^K$. Letting $\boldsymbol{\pi}_j \in \Pi_{K-1}$ denote the mixture weights for the j^{th} group, we have

$$p(x_{ji} \mid \boldsymbol{\pi}_j, \theta_1, \ldots, \theta_K) = \sum_{k=1}^{K} \pi_{jk} f(x_{ji} \mid \theta_k) \qquad i = 1, \ldots, N_j \tag{2.85}$$

Comparing to eq. (2.79), we see that for individual groups LDA is equivalent to a finite mixture model. LDA extends standard mixture models by sharing a common set of clusters among several related groups. These shared parameters θ_k allow learning algorithms to transfer information, while distinct mixture weights $\boldsymbol{\pi}_j$ capture the particular features of each group. Because the data within each group are always observed, an explicit generative model for N_j is unnecessary.

To complete this hierarchical model, we must assign a distribution to the mixture weights $\{\boldsymbol{\pi}_j\}_{j=1}^J$. LDA assumes groups have no distinguishing features beyond the data they contain, and are thus exchangeable. De Finetti's Theorem then implies that these mixture weights are independently sampled from some common prior distribution. For computational simplicity, LDA chooses a conjugate Dirichlet prior:

$$\boldsymbol{\pi}_j \sim \mathrm{Dir}(\alpha) \qquad j = 1, \ldots, J \tag{2.86}$$

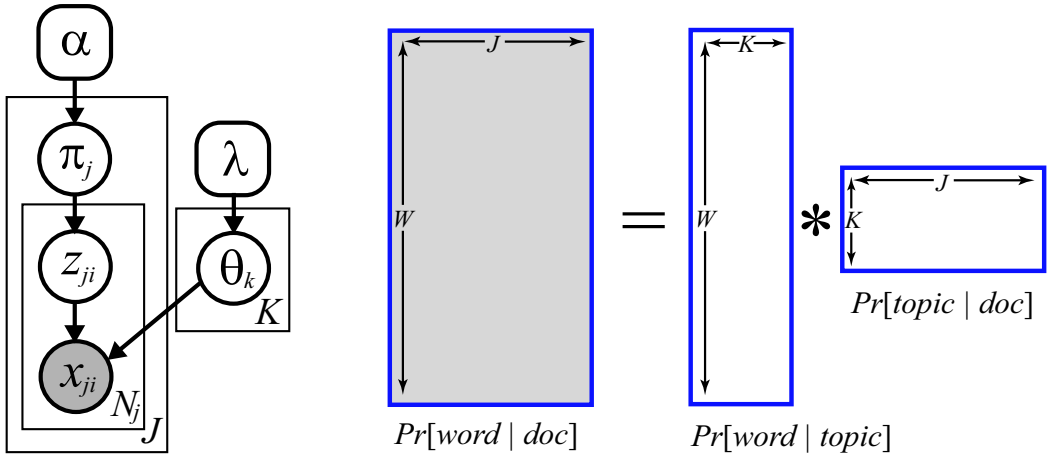


Figure 2.11. The latent Dirichlet allocation (LDA) model for sharing K clusters θ_k among J groups of exchangeable data $\mathbf{x}_j = (x_{j1}, \ldots, x_{jN_j})$. *Left:* LDA as a directed, hierarchical model. Each group's mixture weights $\boldsymbol{\pi}_j \sim \mathrm{Dir}(\alpha)$, while cluster parameters are assigned independent priors $\theta_k \sim H(\lambda)$. $z_{ji} \sim \boldsymbol{\pi}_j$ indicates the shared cluster that generates $x_{ji} \sim F(\theta_{z_{ji}})$. *Right:* When observations are one of W discrete words, LDA can be seen as a probabilistically constrained factorization of the matrix describing the bag of words composing each group, or document. The number K of latent clusters, or topics, determines the factorization's rank. The hyperparameters λ and α define Dirichlet priors for the columns of the word and topic distribution matrices, respectively.

The resulting hierarchical model is illustrated in Fig. 2.11. The Dirichlet hyperparameters α may be either chosen symmetrically (as in eq. (2.41)) to encode prior knowledge [123], or learned from training data in an empirical Bayesian fashion [31]. Often, robustness is improved by assigning conjugate priors $\theta_k \sim H(\lambda)$ to the cluster parameters, as in standard mixture models (see eq. (2.81)). The resulting model is said to be *partially exchangeable* [21], since observations are distinguished only by their associated group. As we demonstrate later, hierarchical graphical models provide a powerful framework for describing dependencies within richly structured datasets.

LDA was originally used to analyze text corpora, by associating groups with documents and data x_{ji} with individual words. The exchangeability assumption treats each document as a "bag of words," incorrectly ignoring the true sentence structure. By doing so, however, LDA leads to tractable algorithms which automatically learn *topics* (clusters) from large, unlabeled document collections [31, 123, 211]. These topics are alternatively known as *aspects*, and LDA as the generative aspect model [211].

For discrete data, LDA effectively determines a low–rank *factorization* of the matrix containing the frequency of each word in each document (see Fig. 2.11). As discussed in detail by Blei et. al. [31], LDA's globally consistent generative model provides conceptual and practical advantages over earlier factorization methods such as *latent semantic analysis* [140]. Importantly, however, LDA can also be generalized to continuous data by associating clusters with appropriate exponential families $F(\theta)$. For example, in later sections of this thesis we use Gaussian "topics" to model spatial data.

As with finite mixture models, the number of clusters or topics K used by LDA is a fixed constant. In practice, learning algorithms are sensitive to this parameter [31, 123], and computationally expensive cross–validation schemes are often needed. Motivated by this issue, Sec. 2.5.4 discusses the hierarchical Dirichlet process [289], a nonparametric generalization of LDA which automatically infers the number of topics needed to explain a given training corpus.

■ 2.2.5 Learning and Inference in Graphical Models

In most applications of graphical models, inference and learning can be posed in terms of a few canonical computational tasks. We divide the random variables composing the graphical model into three sets: observations y, latent or hidden variables x, and parameters θ. While the form of this parameterization differs for directed and undirected graphs, the objectives outlined below arise in both cases.

Inference Given Known Parameters

We begin by assuming the graph's parameters θ are fixed to known, constant values via some previous modeling procedure. The posterior distribution $p(x \mid y, \theta)$ then fully captures available information about the hidden variables x. However, for most realistic graphs the joint sample space $\mathcal{X}$ is far too large to characterize explicitly. For example, given N binary hidden variables, $|\mathcal{X}| = 2^N$. We must thus develop efficient methods to *infer* statistics summarizing this posterior density.

Given global observations y, the joint density $p(x \mid y, \theta)$ is often effectively summarized by the following *posterior marginal distributions:*

$$p(x_i \mid y, \theta) = \int_{\mathcal{X}_{\mathcal{V}\setminus i}} p(x \mid y, \theta)\, dx_{\mathcal{V}\setminus i} \qquad i \in \mathcal{V} \tag{2.87}$$

Here, $\mathcal{V} \setminus i$ denotes all nodes except that corresponding to x_i. The mean of this conditional density is the *Bayes' least squares estimate* [167, 229], while its mode is the *maximizer of the posterior marginals (MPM)* [196] minimizing the expected number of misclassified variables. In addition, the variance or entropy of $p(x_i \mid y, \theta)$ measure the posterior *uncertainty* in these estimates, which can be critical in practical applications [98, 231, 285, 330].

In some cases, hidden variables are instead inferred via a global MAP estimate:

$$\hat{x} = \arg\max_x p(x \mid y, \theta) \tag{2.88}$$

While MAP estimates desirably optimize the joint posterior probability [108], they do not directly provide confidence measures. Furthermore, when observations are noisy or ambiguous, MAP estimation is often less robust than the MPM criterion [196]. For these reasons, we focus primarily on the computation of posterior marginals.

Learning with Hidden Variables

Criteria for learning in graphical models directly generalize those proposed for exponential families in Sec. 2.1.2. Let $p(\theta \mid \lambda)$ denote a prior distribution, with hyperparameters λ, on the graphical model's parameters. In the simplest case, we use the given observations y to determine a single MAP parameter estimate:

$$\hat{\theta} = \arg\max_{\theta} p(\theta \mid y, \lambda) \tag{2.89}$$

$$= \arg\max_{\theta} p(\theta \mid \lambda) \int_{\mathcal{X}} p(x, y \mid \theta)\, dx \tag{2.90}$$

This optimization is complicated by a marginalization over hidden variables x, a difficulty which did not arise with fully observed exponential families (see eq. (2.23)). Inference problems analogous to the posterior marginal computation of eq. (2.87) thus also play a role when learning with hidden variables.

In many situations, the parameters themselves are of interest, and characterizations of their posterior uncertainty are useful. Given some decomposition $\theta = \{\theta_a \mid a \in \mathcal{A}\}$ of the joint parameter space, the posterior marginal distributions of these parameters, and the corresponding hidden variables, equal

$$p(\theta_a \mid y, \lambda) = \int_{\mathcal{X}} \int_{\Theta_{\mathcal{A}\setminus a}} p(x \mid y, \theta)\, p(\theta \mid y, \lambda)\, d\theta_{\mathcal{A}\setminus a}\, dx \qquad a \in \mathcal{A} \tag{2.91}$$

$$p(x_i \mid y, \lambda) = \int_{\Theta} \int_{\mathcal{X}_{\mathcal{V}\setminus i}} p(x \mid y, \theta)\, p(\theta \mid y, \lambda)\, dx_{\mathcal{V}\setminus i}\, d\theta \qquad i \in \mathcal{V} \tag{2.92}$$

Here, θ_a typically parameterizes an individual potential function in undirected graphs, or the conditional distribution of a single variable in directed graphs. Integrating over all parameters and hidden variables, we recover the observations' *marginal likelihood:*

$$p(y \mid \lambda) = \int_{\mathcal{X}} \int_{\Theta} p(x, y \mid \theta)\, p(\theta \mid \lambda)\, d\theta\, dx \tag{2.93}$$

The marginal likelihood is central to Bayesian approaches to model selection, where integration over parameters provides a form of Occam's razor penalizing overly complex models [154, 238]. It also arises in classification problems, for which posterior probabilities are used to determine the most likely explanation of the given observations. Furthermore, maximizing eq. (2.93) with respect to hyperparameters λ provides an empirical Bayesian estimate of the prior distribution (see eq. (2.21)).

Computational Issues

Unfortunately, for many graphical models arising in practice, exact solution of these learning and inference tasks is computationally intractable. Consider, for example, the posterior marginal computation of eq. (2.87). Given N variables, each taking one of K discrete states, this expression leads to a summation containing K^{N-1} terms, which

for arbitrary graphs is NP hard [45]. Optimization of the MAP criterion (eq. (2.88)) is equally challenging [258]. For continuous $\mathcal{X}$, we face a high–dimensional integration which is usually also intractable. A notable exception occurs when all variables are jointly Gaussian, so that linear algebraic connections allow exact inference in $\mathcal{O}(N^3)$ operations [63, 118]. However, even this computation may be extremely difficult for large graphs [285, 330]. Typically, learning problems are no more tractable, since they involve integrations like those arising in inference.

In the following sections, we discuss two general frameworks which provide approximate solutions to learning and inference tasks. We begin in Sec. 2.3 by outlining variational methods which pose these computations as deterministic optimization problems. In Sec. 2.4, we then describe a complementary family of Monte Carlo methods which explore posterior distributions via efficient numerical simulations.

■ 2.3 Variational Methods and Message Passing Algorithms

In this section, we introduce a class of deterministic approximations to the problems of learning and inference posed in Sec. 2.2.5. A *variational method* [98, 161, 251, 311] begins by expressing a statistical inference task as the solution to a mathematical optimization problem. By approximating or relaxing this objective function, one can derive computationally tractable algorithms which bound or approximate the statistics of interest. Often, these algorithms inherit the graphical model's local structure, and can be implemented via the calculation of *messages* passed between neighboring nodes.

We begin our development by considering the marginal log–likelihood of the observed variables y, integrating over hidden states x and parameters θ (see eq. (2.93)). Let $q(x, \theta)$ denote some approximation to the joint posterior density $p(x, \theta \mid y, \lambda)$. Via Jensen's inequality (see eq. (2.9)), any such approximation then provides a lower bound on the marginal likelihood:

$$\begin{aligned}
\log p(y \mid \lambda) &= \log \int_\Theta \int_{\mathcal{X}} p(x, y, \theta \mid \lambda)\, dx\, d\theta \\
&= \log \int_\Theta \int_{\mathcal{X}} q(x, \theta) \frac{p(x, y, \theta \mid \lambda)}{q(x, \theta)}\, dx\, d\theta \\
&\geq \int_\Theta \int_{\mathcal{X}} q(x, \theta) \log \frac{p(x, y, \theta \mid \lambda)}{q(x, \theta)}\, dx\, d\theta \qquad (2.94) \\
&= -D(q(x, \theta) \,||\, p(x, \theta \mid y, \lambda)) + \log p(y \mid \lambda) \qquad (2.95)
\end{aligned}$$

The final equality follows by using Bayes' rule to decompose $p(x, y, \theta \mid \lambda)$. Given some family of approximating densities $\mathcal{Q}$, the best lower bound is achieved by the distribution minimizing the KL divergence from the true posterior:

$$\hat{q}(x, \theta) = \arg\min_{q \in \mathcal{Q}} D(q(x, \theta) \,||\, p(x, \theta \mid y, \lambda)) \qquad (2.96)$$

Of course, if $\mathcal{Q}$ is unrestricted the optimum is trivially $\hat{q}(x, \theta) = p(x, \theta \mid y, \lambda)$. Variational methods instead choose $\mathcal{Q}$ to be a simpler density representation for which

computations are *tractable*.

The following sections explore two classes of variational methods. In Sec. 2.3.1, we discuss *mean field* methods which use tractable families $\mathcal{Q}$ to derive a simplifying decomposition of $D(q\,||\,p)$. This representation is used to develop iterative methods guaranteed to converge to a local optimum of eq. (2.96). Sec. 2.3.2 then describes *loopy belief propagation (BP)*, which uses properties of tree–structured graphical models to motivate intuitive approximations of $\mathcal{Q}$ and $D(q\,||\,p)$. While loopy BP leads to approximations, rather than bounds, on the marginal likelihood, it is often more accurate in practice. Importantly, for either method the optimizing density $\hat{q}(x, \theta)$ provides estimates of the posterior marginal densities motivated in Sec. 2.2.5.

For simplicity, we focus on algorithms which infer conditional marginal densities in pairwise Markov random fields. However, similar variational methods may also be derived for directed [161] and factor [98, 324] graphs. In Sec. 2.3.3, we then show how the *expectation–maximization (EM)* algorithm extends inference methods to learn parameters from partially labeled data.

■ 2.3.1 Mean Field Approximations

Given some fixed, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider a pairwise Markov random field as introduced in Sec. 2.2.2:

$$p(x \mid y) = \frac{1}{Z} \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} \psi_i(x_i, y) \tag{2.97}$$

$$= \exp\left\{ - \sum_{(i,j)\in\mathcal{E}} \phi_{ij}(x_i, x_j) - \sum_{i\in\mathcal{V}} \phi_i(x_i, y) - \Phi \right\} \tag{2.98}$$

Here, $\Phi = \log Z$ is the log partition function, and eq. (2.98) expresses the joint density via the negative logarithms of the potential functions:

$$\phi_{ij}(x_i, x_j) \triangleq -\log \psi_{ij}(x_i, x_j) \qquad \phi_i(x_i, y) \triangleq -\log \psi_i(x_i, y) \tag{2.99}$$

This representation is related to *Boltzmann's law* from statistical mechanics [337], which says that for a system in equilibrium at temperature T, a state x with energy $\phi(x)$ has probability $p(x) \propto \exp\{-\phi(x)/T\}$. For a pairwise MRF, the *energy* thus equals

$$\phi(x) = \sum_{(i,j)\in\mathcal{E}} \phi_{ij}(x_i, x_j) + \sum_{i\in\mathcal{V}} \phi_i(x_i, y) \tag{2.100}$$

We assume that the parameters θ defining the graph's potentials have been fixed by some previous modeling procedure, and do not denote them explicitly. Instead, we focus on estimating the posterior marginal densities $p(x_i \mid y)$ for all nodes $i \in \mathcal{V}$.

To develop the *mean field* method, we decompose the KL divergence (see eq. (2.96))

between an approximate posterior $q(x)$ and the target pairwise MRF as follows:

$$D(q \,||\, p) = \int_{\mathcal{X}} q(x) \log q(x) \; dx - \int_{\mathcal{X}} q(x) \log p(x \mid y) \; dx \tag{2.101}$$

$$= -H(q) + \int_{\mathcal{X}} \phi(x) \, q(x) \; dx + \Phi \tag{2.102}$$

The first term of this decomposition is the *negative entropy*, while by analogy with Boltzmann's law the second term is known as the *average energy*. Excluding the log partition function Φ, which is constant assuming fixed parameters, eq. (2.102) is sometimes called the *Gibbs free energy* [337]. Minimizing this free energy with respect to $q(x)$, we recover the true posterior of eq. (2.98). For an alternative interpretation of this relationship, in which the negative entropy arises as the conjugate dual of the log partition function, see [161, 311].

Naive Mean Field

Mean field methods are derived by choosing a restricted family of approximating densities $\mathcal{Q}$ for which minimization of eq. (2.102) is tractable. By appropriately parameterizing $\mathcal{Q}$, fixed points of this minimization also give estimates $q_i(x_i) \approx p(x_i \mid y)$ of the desired marginals. In the simplest case, the so–called *naive mean field* [98, 161, 311, 337] approximation takes $\mathcal{Q}$ to be the set of fully factorized densities:

$$q(x) = \prod_{i \in \mathcal{V}} q_i(x_i) \tag{2.103}$$

Recall that the joint entropy of a set of independent random variables equals the sum of their individual entropies [49]. Inserting the factorization of eq. (2.103) into the free energy of eq. (2.102) and simplifying, we then have

$$D(q \,||\, p) = -\sum_{i \in \mathcal{V}} H(q_i) + \sum_{i \in \mathcal{V}} \int_{\mathcal{X}_i} \phi_i(x_i, y) \, q_i(x_i) \; dx_i$$

$$\cdots + \sum_{(i,j) \in \mathcal{E}} \int_{\mathcal{X}_i} \int_{\mathcal{X}_j} \phi_{ij}(x_i, x_j) \, q_i(x_i) \, q_j(x_j) \; dx_j \, dx_i + \Phi \tag{2.104}$$

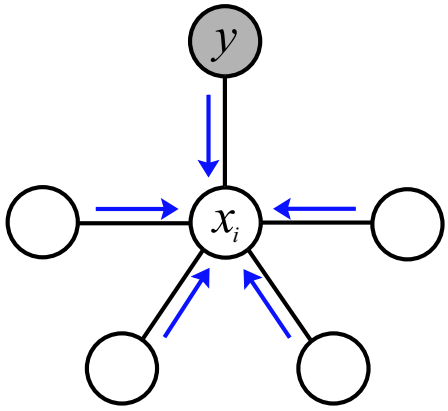
Here, we have used eq. (2.100) to decompose the average energy according to the pairwise MRF's graphical structure.

To minimize the mean field free energy of eq. (2.104), we construct a Lagrangian constraining each approximating marginal distribution to integrate to one:

$$\mathcal{L}(q, \gamma) = D(q \,||\, p) + \sum_{i \in \mathcal{V}} \gamma_i \left(1 - \int_{\mathcal{X}_i} q_i(x_i) \; dx_i \right) \tag{2.105}$$

Differentiating $\mathcal{L}(q, \gamma)$ with respect to $q_i(x_i)$ and simplifying, we find that the optimal marginals are related by the following fixed point equations:

$$\log q_i(x_i) = -\phi_i(x_i, y) - \sum_{j \in \Gamma(i)} \int_{\mathcal{X}_j} \phi_{ij}(x_i, x_j) \, q_j(x_j) \; dx_j + \bar{\gamma}_i \qquad i \in \mathcal{V} \tag{2.106}$$



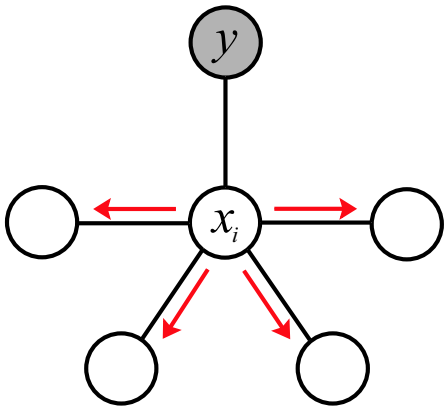


Figure 2.12. Message passing implementation of the naive mean field method. *Left:* Approximate marginal densities are determined from the normalized product of the local observation potential with messages sent from neighboring nodes. *Right:* Given an updated marginal estimate, new messages are calculated and transmitted to all neighbors.

Here, $\bar{\gamma}_i$ is a constant chosen to satisfy the marginalization constraint. Due to the pairwise relationships in the free energy of eq. (2.104), the marginal $q_i(x_i)$ at node i depends directly on the corresponding marginals at neighboring nodes $\Gamma(i)$. Thus, even though $\mathcal{Q}$ is fully factorized, the corresponding mean field solution desirably propagates information from local potentials throughout the graph.

To implement the mean field method, we must have a tractable representation for the marginal densities $q_i(x_i)$, and a corresponding algorithm for updating these marginals. Consider the following decomposition of the mean field fixed point equation (eq. (2.106)):

$$q_i(x_i) \propto \psi_i(x_i, y) \prod_{j \in \Gamma(i)} m_{ji}(x_i) \qquad i \in \mathcal{V} \tag{2.107}$$

$$m_{ji}(x_i) \propto \exp\left\{ -\int_{\mathcal{X}_j} \phi_{ij}(x_i, x_j)\, q_j(x_j)\, dx_j \right\} \qquad j \in \Gamma(i) \tag{2.108}$$

We interpret $m_{ji}(x_i)$ as a *message* sent from j to its neighboring node i. As illustrated in Fig. 2.12, mean field algorithms alternate between updating a local marginal estimate (eq. (2.107)), and using this new marginal to calculate an updated message for each neighbor (eq. (2.108)). If marginals are updated sequentially, the mean field algorithm is a form of coordinate descent which converges to a local minimum of the free energy (eq. (2.104)). Parallel updates are also possible, but do not guarantee convergence.

If $\mathcal{X}_i$ takes K discrete values, we can represent messages and marginals by K–dimensional vectors. The integration of eq. (2.108) then becomes a summation, allowing direct message computation in $\mathcal{O}(K^2)$ operations. For hidden variables defined on

continuous spaces $\mathcal{X}_i$, implementation of the mean field method is more complicated. In jointly Gaussian random fields, the integral message updates can be rewritten in terms of the posterior means [311], leading to an algorithm equivalent to the classic Gauss–Seidel iteration for linear systems [63]. More generally, for directed or undirected graphs where all potentials are defined by exponential families, the mean field marginals are finitely parameterized by the corresponding sufficient statistics [110]. From eq. (2.108), we see that messages then become exponentiated expectations of these statistics with respect to neighboring nodes. This approach can be extended to infer approximate marginal distributions for parameters θ_a (see eq. (2.91)) when all priors $p(\theta_a \mid \lambda_a)$ are conjugate [110, 331]. The VIBES software package exploits this flexibility, along with the local structure of message–passing updates, to automatically generate mean field inference code for directed graphical models [331].

While exponential families are somewhat flexible, many applications involve more complex, continuous potentials which lack sufficient statistics. In such cases, there is no finite representation for the marginal densities $q_i(x_i)$, and message updates are typically intractable. Sometimes, however, the mean field algorithm can be reasonably approximated by Monte Carlo methods which represent $q_i(x_i)$ via a collection of random samples [332]. We discuss these methods in more detail in Sec. 2.4.

Information Theoretic Interpretations

In information theory, the KL divergence $D(p \,||\, q)$ arises as a measure of the asymptotic inefficiency, or information loss [49], incurred by assuming that a stochastic process x has distribution $q(x)$ when its true distribution is $p(x \mid y)$. From this perspective, given an approximating family $\mathcal{Q}$, it seems more appropriate to minimize $D(p \,||\, q)$ over $q \in \mathcal{Q}$ rather than the "backwards" divergence $D(q \,||\, p)$ underlying mean field methods. Indeed, for fully factorized $\mathcal{Q}$ as in eq. (2.103), $D(p \,||\, q)$ has an intuitive form:

$$\begin{aligned} D(p \,||\, q) &= \int_{\mathcal{X}} p(x \mid y) \log p(x \mid y)\, dx - \int_{\mathcal{X}} p(x \mid y) \log \prod_{i \in \mathcal{V}} q_i(x_i)\, dx \\ &= -H(p) - \sum_{i \in \mathcal{V}} \int_{\mathcal{X}_i} p(x_i \mid y) \log q_i(x_i)\, dx_i \\ &= \sum_{i \in \mathcal{V}} H(p_i) - H(p) + \sum_{i \in \mathcal{V}} D(p_i \,||\, q_i) \end{aligned} \tag{2.109}$$

The first two terms, which do not depend on $q(x)$, capture the fundamental information loss incurred by *any* approximation neglecting depencies among the hidden variables. The last term is uniquely minimized by taking $q_i(x_i) = p(x_i \mid y)$, so that the true posterior marginals are exactly recovered. Interestingly, mean field methods can also be derived via a first–order Taylor series expansion of this divergence [166].

While the decomposition of eq. (2.109) shows that the marginals $p(x_i \mid y)$ provide an appropriate summary of $p(x \mid y)$, it does not provide a computational method for determining these marginals. Conversely, while mean field methods do *not* generally

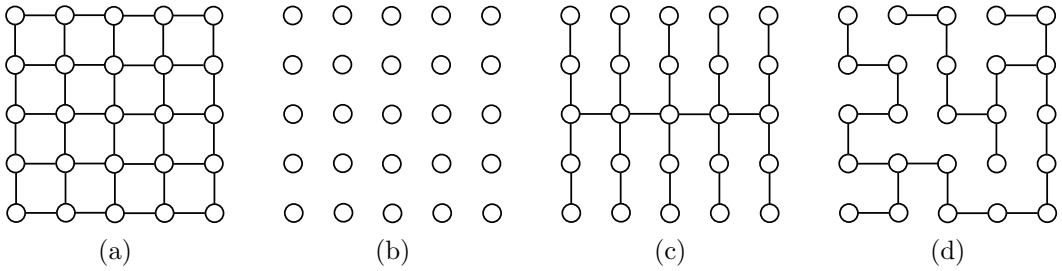


Figure 2.13. Tractable subgraphs underlying different variational methods for approximate inference. (a) Original nearest–neighbor grid (observation nodes not shown). (b) Fully factored model employed by the naive mean field method. (c) An embedded tree, as might be exploited by a structured mean field method. (d) Another of this grid's many embedded trees.

recover the true posterior marginals, minimization of $D(q \,||\, p)$ leads to tractable algorithms providing potentially useful approximations. Indeed, as we discuss in later sections, this variational approach provides a flexible framework for developing richer approximations with increased accuracy. See [161, 311] for an alternative motivation of mean field methods based on conjugate duality.

Structured Mean Field

Results from the statistical physics literature guarantee that, for certain densely connected models with sufficiently homogeneous potentials, the naive mean field approximation becomes exact as the number of variables N approaches infinity [337]. However, for sparse, irregular graphs like those considered by this thesis, its marginal estimates $q_i(x_i)$ can be extremely *overconfident*, underestimating the uncertainty of the true posterior $p(x_i \mid y)$. In addition, the mean field iteration of eqs. (2.107, 2.108) often gets stuck in local optima which differ substantially from the true posterior [98, 320]. Geometrically, these local optima arise because the set of pairwise marginals achievable via fully factorized densities is not convex [311].

Motivated by these issues, researchers have developed a variety of variational methods which extend and improve the naive mean field approximation [98, 161, 251, 311]. In particular, fully factorized approximations effectively remove all of the target graphical model's edges. However, one can also consider *structured mean field* methods based on approximating families which directly capture more of the original graph's structure (see Fig. 2.13). Optimization of these approximations is possible assuming exact inference in the chosen subgraphs is tractable [111, 252, 327, 335]. As we show in the following section, Markov chains and trees allow fast, exact recursive inference algorithms which form the basis for a variety of higher–order variational methods.

■ 2.3.2 Belief Propagation

As discussed in Sec. 2.2.5, direct solution of learning and inference problems arising in graphical models is typically intractable. Sometimes, however, global inference tasks

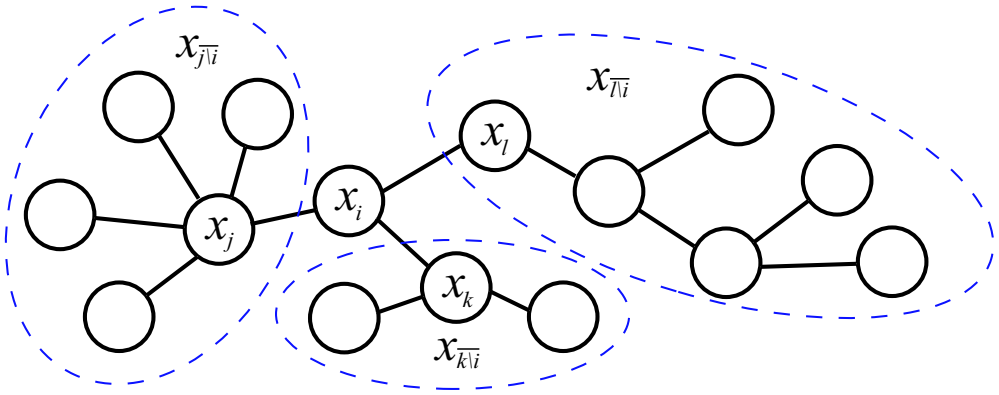


Figure 2.14. For a tree–structured graph, each node i partitions the graph into $|\Gamma(i)|$ disjoint subtrees. Conditioned on x_i, the variables $x_{\overline{j\setminus i}}$ in these subtrees are independent.

can be efficiently decomposed into a set of simpler, local computations. In particular, for tree–structured graphical models a generalization of dynamic programming known as *belief propagation (BP)* [178, 231, 255] recursively computes exact posterior marginals in linear time. In the following sections, we provide a brief derivation of BP, and discuss issues arising in its implementation. We then present a variational interpretation of BP which justifies extensions to graphs with cycles.

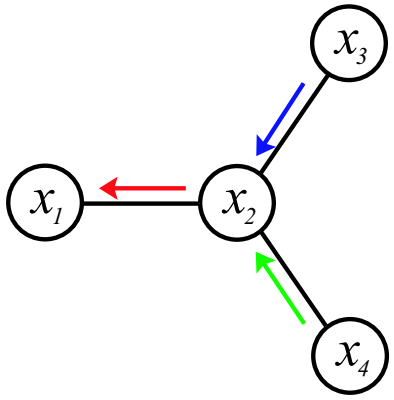
Message Passing in Trees

Consider a pairwise MRF, parameterized as in Sec. 2.3.1, whose underlying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is tree–structured. As shown in Fig. 2.14, any node $i \in \mathcal{V}$ divides such a tree into $|\Gamma(i)|$ disjoint subsets:

$$\overline{j \setminus i} \triangleq \{j\} \cup \{k \in \mathcal{V} \mid \text{no path from } k \to j \text{ intersects } i\} \tag{2.110}$$

By the Markov properties of $\mathcal{G}$, the variables $x_{\overline{j\setminus i}}$ in these sub–trees are conditionally independent given x_i. The BP algorithm exploits this structure to recursively decompose the computation of $p(x_i \mid y)$ into a series of simpler, local calculations.

From the Hammersley–Clifford Theorem, Markov properties are expressed through the *algebraic* structure of the pairwise MRF's factorization into clique potentials. As illustrated in Fig. 2.15, tree–structured graphs allow multi–dimensional integrals (or summations) to be decomposed into a series of simpler, one–dimensional integrals. As in dynamic programming [24, 90, 303], the overall integral can then be computed via a recursion involving messages sent between neighboring nodes. This decomposition is an instance of the same *distributive law* underlying a variety of other algorithms [4, 50, 255], including the fast Fourier transform. Critically, because messages are shared among similar decompositions associated with different nodes, BP efficiently and *simultaneously* computes the desired marginals for all nodes in the graph.



$$
\begin{aligned}
p(x_1) &\propto \iiint \psi_1(x_1)\psi_{12}(x_1,x_2)\psi_2(x_2)\psi_{23}(x_2,x_3)\psi_3(x_3)\psi_{24}(x_2,x_4)\psi_4(x_4)\,dx_4\,dx_3\,dx_2 \\
&\propto \psi_1(x_1)\iiint \psi_{12}(x_1,x_2)\psi_2(x_2)\psi_{23}(x_2,x_3)\psi_3(x_3)\psi_{24}(x_2,x_4)\psi_4(x_4)\,dx_4\,dx_3\,dx_2 \\
&\propto \psi_1(x_1)\int \psi_{12}(x_1,x_2)\psi_2(x_2)\left[\iint \psi_{23}(x_2,x_3)\psi_3(x_3)\psi_{24}(x_2,x_4)\psi_4(x_4)\,dx_4\,dx_3\right]dx_2 \\
&\propto \psi_1(x_1)\int \psi_{12}(x_1,x_2)\psi_2(x_2)\underbrace{\left[\int \psi_{23}(x_2,x_3)\psi_3(x_3)\,dx_3\right]}_{m_{32}(x_2)}\cdot\underbrace{\left[\int \psi_{24}(x_2,x_4)\psi_4(x_4)\,dx_4\right]}_{m_{42}(x_2)}dx_2 \\
&\qquad\qquad m_{21}(x_1) \propto \int \psi_{12}(x_1,x_2)\psi_2(x_2)m_{32}(x_2)\,m_{42}(x_2)\;dx_2
\end{aligned}
$$

Figure 2.15. Example derivation of the BP message passing recursion through repeated application of the distributive law. Because the joint distribution $p(x)$ factorizes as a product of pairwise clique potentials, the joint integral can be decomposed via messages $m_{ji}(x_i)$ sent between neighboring nodes.

To derive the BP algorithm, we begin by considering the clique potentials corresponding to particular subsets of the full graph:

$$\Psi_{\mathcal{A}}(x_{\mathcal{A}}) \triangleq \prod_{(i,j)\in\mathcal{E}(\mathcal{A})} \psi_{ij}(x_i,x_j) \prod_{i\in\mathcal{A}} \psi_i(x_i,y) \qquad \mathcal{A}\subset\mathcal{V} \tag{2.111}$$

Here, $\mathcal{E}(\mathcal{A}) \triangleq \{(i,j)\in\mathcal{E} \mid i,j\in\mathcal{A}\}$ are the edges contained in the *node–induced subgraph* [50] corresponding to $\mathcal{A}$. Using the partitions illustrated in Fig. 2.14, we can then write the marginal distribution of any node as follows:

$$p(x_i \mid y) \propto \int_{\mathcal{X}_{\mathcal{V}\setminus i}} \psi_i(x_i,y) \prod_{j\in\Gamma(i)} \psi_{ij}(x_i,x_j)\,\Psi_{\overline{j\setminus i}}(x_{\overline{j\setminus i}})\;dx_{\mathcal{V}\setminus i} \tag{2.112}$$

$$\propto \psi_i(x_i,y) \prod_{j\in\Gamma(i)} \int_{\mathcal{X}_{\overline{j\setminus i}}} \psi_{ij}(x_i,x_j)\,\Psi_{\overline{j\setminus i}}(x_{\overline{j\setminus i}})\;dx_{\overline{j\setminus i}} \tag{2.113}$$

To verify eq. (2.112), note that it simply regroups the pairwise MRF's potentials according to Fig. 2.14. Because the variables in the subgraphs separated by node i share no potentials, the joint integral then decomposes accordingly. Interpreting the integrals in eq. (2.113) as *messages* $m_{ji}(x_i)$ sent to node i from each of its neighbors, we have

$$p(x_i \mid y) \propto \psi_i(x_i, y) \prod_{j \in \Gamma(i)} m_{ji}(x_i) \tag{2.114}$$

The message $m_{ji}(x_i)$ is a function providing the value of the corresponding integral for each possible $x_i \in \mathcal{X}_i$. Note that in a graph with cycles, node i would not necessarily disjointly partition the potentials, so the decomposition of eqs. (2.112, 2.113) is invalid.

In some applications, the joint distributions $p(x_i, x_j \mid y)$ of pairs of nodes are also of interest [324]. In tree–structured graphs, neighboring nodes $(i, j) \in \mathcal{E}$ partition the global set of clique potentials as follows:

$$p(x \mid y) \propto \psi_{ij}(x_i, x_j)\, \psi_i(x_i, y)\, \psi_j(x_j, y) \prod_{\ell \in \Gamma(i) \backslash j} \Psi_{\overline{\ell \backslash i}}(x_{\overline{\ell \backslash i}}) \prod_{k \in \Gamma(j) \backslash i} \Psi_{\overline{k \backslash j}}(x_{\overline{k \backslash j}}) \tag{2.115}$$

The corresponding subgraphs are illustrated in Fig. 2.16. Applying this decomposition as in eq. (2.112), and integrating over all variables except x_i and x_j, we then have

$$p(x_i, x_j \mid y) \propto \psi_{ij}(x_i, x_j)\, \psi_i(x_i, y)\, \psi_j(x_j, y) \prod_{\ell \in \Gamma(i) \backslash j} m_{\ell i}(x_i) \prod_{k \in \Gamma(j) \backslash i} m_{kj}(x_j) \tag{2.116}$$

The messages decomposing this pairwise marginal density are defined *identically* to those used in eq. (2.114) to compute single–node marginals.

As defined in eq. (2.113), the messages may still be complex functions of large groups of variables. To derive an efficient recursive decomposition, we consider the marginalization constraint relating the single–node and pairwise marginal distributions:

$$p(x_i \mid y) = \int_{\mathcal{X}_j} p(x_i, x_j \mid y)\, dx_j \tag{2.117}$$

$$\begin{aligned} \psi_i(x_i, y) \prod_{\ell \in \Gamma(i)} m_{\ell i}(x_i) \propto \psi_i(x_i, y) \prod_{\ell \in \Gamma(i) \backslash j} m_{\ell i}(x_i) \\ \cdots \times \int_{\mathcal{X}_j} \psi_{ij}(x_i, x_j)\, \psi_j(x_j, y) \prod_{k \in \Gamma(j) \backslash i} m_{kj}(x_j)\, dx_j \end{aligned} \tag{2.118}$$

Note that all but one of the terms on the left hand side of eq. (2.118) have identical functions of x_i on the right hand side. Cancelling these terms, as illustrated graphically in Fig. 2.16 (see [339]), the marginalization constraint is always satisfied when the remaining message $m_{ji}(x_i)$ is defined as follows:

$$m_{ji}(x_i) \propto \int_{\mathcal{X}_j} \psi_{ij}(x_i, x_j)\, \psi_j(x_j, y) \prod_{k \in \Gamma(j) \backslash i} m_{kj}(x_j)\, dx_j \tag{2.119}$$

This recursion expresses one outgoing message from node j in terms of the other $(|\Gamma(j)| - 1)$ incoming messages. At the leaves of the tree, eq. (2.119) and our initial message definition (eq. (2.113)) coincide:

$$m_{ji}(x_i) \propto \int_{\mathcal{X}_j} \psi_{ij}(x_i, x_j)\, \psi_j(x_j, y)\, dx_j \qquad \Gamma(j) = \{i\} \tag{2.120}$$

Thus, by recursively computing the messages along every edge according to eq. (2.119), we may then easily find any single–node (eq. (2.114)) or pairwise (eq. (2.116)) marginal of interest. For more formal derivations of this algorithm, see [4, 255].

Fig. 2.16 summarizes the BP message update recursion, and the corresponding message products which provide marginal densities. These posterior marginals are sometimes called *beliefs*, by analogy with expert systems developed in the artificial intelligence community [50, 178, 231]. Anticipating later extensions of BP which only provide approximate posterior marginals, we denote the beliefs for individual and pairs of nodes by $q_i(x_i)$ and $q_{ij}(x_i, x_j)$, respectively. This form of the BP algorithm is due to Shafer and Shenoy [255], who emphasized the central role of factorization in recursive inference. Several other variants of BP have been proposed [50, 158, 306], including versions adapted to directed Bayesian networks [178, 231] and factor graphs [175, 324].

To implement the BP algorithm, a *schedule* by which the messages are updated must be selected. In tree–structured graphs, an appropriate ordering of these updates requires each message to be computed only once, so that all N marginals may be determined in $\mathcal{O}(N)$ operations. One possible efficient schedule chooses some node as the *root* of the tree. This induces a partial ordering of the nodes in *scale* according to their distance from the root (see Fig. 2.6). Messages are then computed in two stages: an upward sweep proceeding from leaves to the root, followed by a downward sweep propagating information from the root throughout the graph [34, 41, 330]. Alternatively, an efficient *decentralized* schedule begins by passing outward messages from all leaf nodes. Internal message $m_{ji}(x_i)$ is then computed once node j has received messages from all $(|\Gamma(j)| - 1)$ of its other neighbors [175].

One can also consider a *parallel* form of the BP algorithm, in which every node recomputes all outgoing messages at each iteration, based on messages received from its neighbors in the previous iteration [231]. After T iterations, local marginal estimates will then optimally incorporate information from all nodes within distance T [4]. Convergence to the optimal posterior marginals occurs once the number of iterations equals the tree's diameter (at most $(N-1)$). While parallel BP updates are typically inefficient on a serial computer, they are useful in distributed implementations [100, 245].

Representing and Updating Beliefs

As with the mean field algorithm, implementations of BP require a tractable representation of the beliefs, and corresponding computational methods for the message updates of eq. (2.119). In the simplest case, where each variable x_i takes one of K discrete values ($|\mathcal{X}_i| = K$), messages and marginals can be represented by K–dimensional vectors.

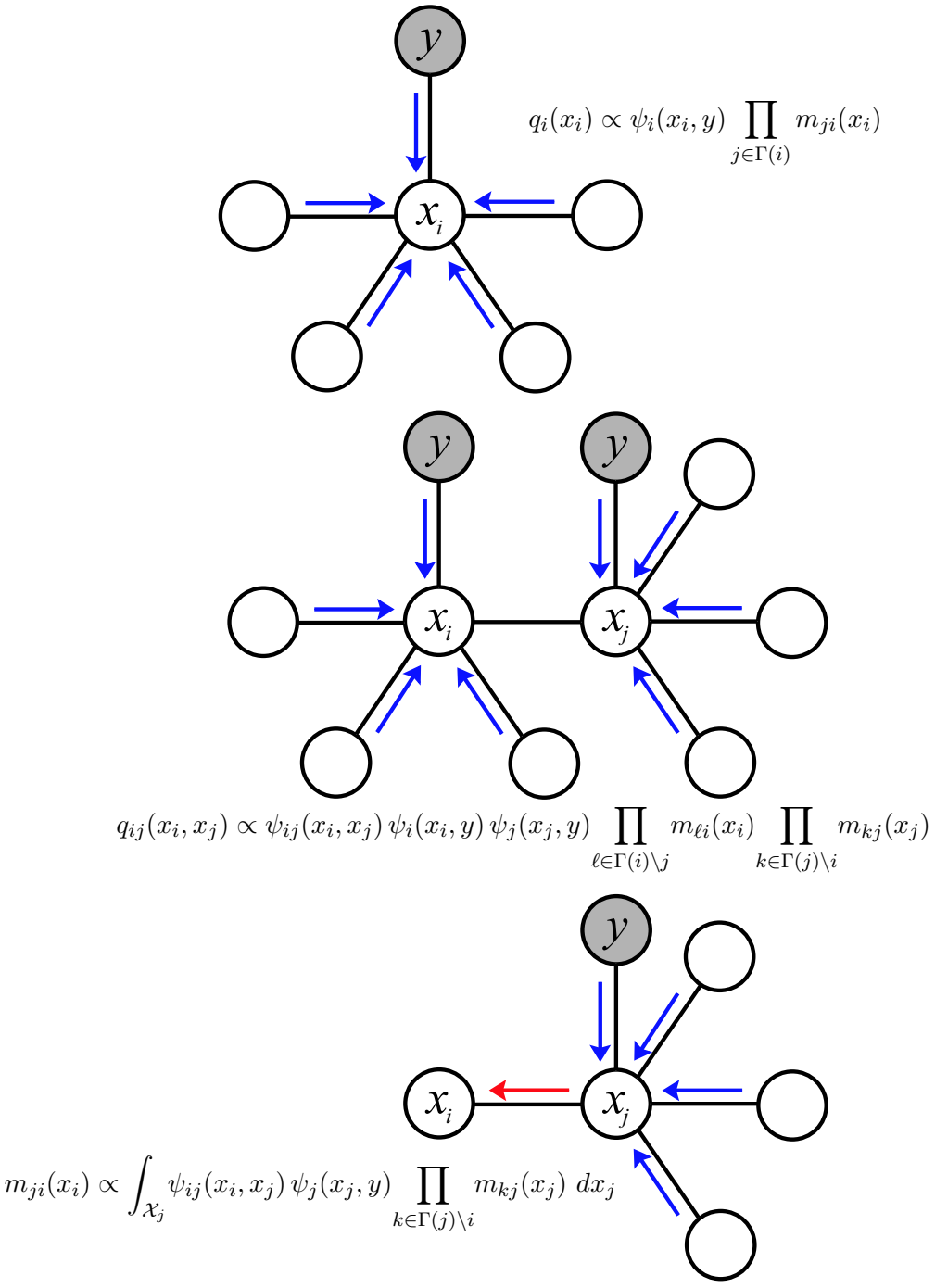


Figure 2.16. Message passing recursions underlying the BP algorithm. *Top:* Approximate marginal densities are determined from the normalized product of the local observation potential with messages sent from neighboring nodes. *Middle:* Pairwise marginal densities are derived from a similar message product. *Bottom:* A new outgoing message (red) is computed from all other incoming messages (blue).

The message update integral then becomes a matrix–vector product, which in general requires $\mathcal{O}(K^2)$ operations:

$$m_{ji}(x_i) \propto \sum_{x_j \in \mathcal{X}_j} \psi_{ij}(x_i, x_j)\, \psi_j(x_j, y) \prod_{k \in \Gamma(j) \setminus i} m_{kj}(x_j) \tag{2.121}$$

For an N node tree, BP can then compute all marginals in $\mathcal{O}(NK^2)$ operations, a dramatic savings versus the $\mathcal{O}(K^N)$ cost of brute–force summation. When the pairwise potentials $\psi_{ij}(x_i, x_j)$ are sufficiently regular, techniques such as FFTs can further reduce costs to $\mathcal{O}(K \log K)$, or $\mathcal{O}(K)$ with additional approximations [80]. By analogy with the form of eq. (2.121), BP is sometimes called the *sum–product* algorithm [175]. Specializing discrete BP to temporal HMMs (see Fig. 2.7), we recover the *forward–backward* algorithm, which is widely used for speech processing [235]. More generally, recursions equivalent to BP are often applied to multiscale discrete–state quadtree models arising in image processing [34, 330].

Inference in HMMs with continuous hidden variables has been extensively studied in the context of state space representations for dynamical systems [8, 164]. For linear systems with Gaussian dynamics and observation noise, the posterior distribution of the states is jointly Gaussian, and marginals are thus determined by their mean and covariance. In such models, BP is equivalent to *fixed–interval smoothing* algorithms which combine the Kalman filter with a complementary reverse–time recursion [8, 163, 164, 249]. These algorithms are readily generalized to any tree–structured graphical model with Gaussian potentials [41, 330]. In undirected Gaussian MRFs, BP messages are most easily updated in information form, via inverse covariance matrices [276, 321].

In contrast to the Gaussian case, continuous state space models containing non–linear or non–Gaussian interactions typically lead to message updates which lack a closed analytic form [8, 153]. Even in cases where all potentials are drawn from exponential families, the corresponding posterior densities may not have finite–dimensional sufficient statistics [326]. These difficulties have motivated a wide range of methods which approximate the true posterior by a tractable analytic form. For example, the *extended Kalman filter* fits a Gaussian posterior via a gradient–based linearization [8, 153], while the *unscented Kalman filter* uses a more accurate quadrature method [162]. More generally, given any exponential family, *expectation propagation (EP)* [135, 213] uses the moment matching conditions of Sec. 2.1.1 to approximate the beliefs produced by each message update. Note, however, that determining the sufficient statistics for such projections can itself be a challenging problem [344].

For many graphical models, the true posterior marginals are multimodal, or exhibit other features poorly approximated by standard exponential families. In some cases, a fixed K–point discretization leads to an effective histogram approximation of the true continuous beliefs [11, 80, 95, 169]. However, as K must in general grow exponentially with the dimension of $\mathcal{X}_i$, computation of the discrete messages underlying this approach can be extremely demanding. This has motivated approaches which use online message computations to *dynamically* discretize the belief space. In some cases,

deterministic rules are used to prune discretization grids [47, 48] or Gaussian mixture approximations [5, 94, 267]. Alternatively, Monte Carlo methods can be used to iteratively improve stochastic approximations to the true beliefs [9, 197, 224]. In particular, Chap. 3 describes and extends a family of *particle filters* [11, 70] which approximate messages and beliefs by a set of weighted samples.

Message Passing in Graphs with Cycles

Our earlier derivation of the BP algorithm assumed a tree–structured graph. The *junction tree* algorithm extends BP to allow exact inference in arbitrary graphs [178, 255]. Let $\mathcal{G}$ be an undirected graph (directed graphs are first moralized as in Fig. 2.4(c)). In the first of three stages, $\mathcal{G}$ is *triangulated* by adding edges so that all cycles of length four or greater contain a chord. Then, a tree is formed from the maximal cliques of the triangulated graph. Finally, a variant of BP performs exact inference on the resulting junction tree (for more details, see [4, 50, 158, 177]). The triangulation step ensures that any variables shared by two cliques are also members of other cliques along their connecting path. This *running intersection property* must be satisfied for local junction tree computations to produce globally consistent estimates. For many graphs, however, triangulation greatly increases the size of the resulting cliques. In such cases, the number of states associated with these cliques grows exponentially, and inference in the junction tree can become intractable [45].

For graphs in which exact inference is infeasible, we can still use the BP algorithm to develop improved variational methods. As mentioned in Sec. 2.3.1, one approach uses *embedded* trees (as in Fig. 2.13) to develop structured mean field bounds with increased accuracy [111, 252, 327]. In this thesis, we focus on an alternative method known as *loopy belief propagation* [231]. As summarized in Fig. 2.16, the BP algorithm proceeds entirely via a series of *local* message updates. Given a graph with cycles, loopy BP iterates a parallel form of these message updates. Remarkably, in many applications this seemingly heuristic method converges to beliefs which very closely approximate the true posterior marginals [101, 219].

The traditional dynamic programming derivation of BP provides no justification for loopy BP, other than the vague intuition that it should work well for graphs whose cycles are "long enough." In the following section, we provide a variational interpretation which places loopy BP on firmer conceptual ground. We then briefly survey known theoretical results and extensions.

Loopy BP and the Bethe Free Energy

Unsurprisingly, variational analyses of loopy BP are closely related to the Markov structure of tree–structured graphical models. The following proposition provides a local factorization which is valid for any tree–structured joint distribution, and derives a corresponding entropy decomposition.

Proposition 2.3.1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a tree–structured undirected graph. Any joint distribution $p(x)$ which is Markov with respect to $\mathcal{G}$ factorizes according to marginal distributions defined on the graph's nodes and edges:*

$$p(x) = \prod_{(i,j) \in \mathcal{E}} \frac{p_{ij}(x_i, x_j)}{p_i(x_i) p_j(x_j)} \prod_{i \in \mathcal{V}} p_i(x_i) \tag{2.122}$$

The joint entropy $H(p)$ then decomposes according to the graphical structure:

$$H(p) = \sum_{i \in \mathcal{V}} H(p_i) - \sum_{(i,j) \in \mathcal{E}} I(p_{ij}) \tag{2.123}$$

Proof. The factorization of eq. (2.122) is a special case of the junction tree decomposition, and can be formally verified using an induction argument [50, 177, 178]. In Markov chains, for example, it is easily derived from the standard representation via one–step transition probabilities. The entropy decomposition of eq. (2.123) then follows directly from the definitions of entropy (eq. (2.7)) and mutual information (eq. (2.11)). □

Interestingly, eq. (2.122) shows that the marginal distributions of tree–structured graphs can be inferred via a *reparameterization* operation which transforms arbitrary clique potentials (as in eq. (2.97)) to this particular canonical form [306].

Given any tree–structured undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider a pairwise MRF $p(x \mid y)$ parameterized as in eq. (2.98). Using the entropy decomposition of eq. (2.123), the KL divergence $D(q \,||\, p)$ from any tree–structured approximation $q(x)$ equals

$$\begin{aligned} D(q \,||\, p) = -\sum_{i \in \mathcal{V}} H(q_i) + \sum_{(i,j) \in \mathcal{E}} I(q_{ij}) + \sum_{i \in \mathcal{V}} \int_{\mathcal{X}_i} \phi_i(x_i, y)\, q_i(x_i)\, dx_i \\ \cdots + \sum_{(i,j) \in \mathcal{E}} \int_{\mathcal{X}_i} \int_{\mathcal{X}_j} \phi_{ij}(x_i, x_j)\, q_{ij}(x_i, x_j)\, dx_j\, dx_i + \Phi \end{aligned} \tag{2.124}$$

This divergence depends solely on the pairwise marginals $q_{ij}(x_i, x_j)$, not on other non–local aspects of $q(x)$. To arrive at the loopy BP algorithm, we assume that the KL divergence of eq. (2.124) is *approximately* correct even for graphs with cycles. The beliefs $q_i(x_i)$ and $q_{ij}(x_i, x_j)$ are then *pseudo–marginals*, which differ from the true marginals of $p(x \mid y)$. In statistical physics, this approximation is known as the *Bethe free energy* [337, 340]. Note that for pairwise MRFs, the average energy term can be exactly written in terms of pairwise marginals. The approximation thus involves incorrectly applying the tree–based entropy of eq. (2.123) to cyclic graphs.

As with our earlier mean field derivation, loopy BP is derived by using Lagrangian methods to minimize the Bethe free energy of eq. (2.124). First, each edge $(i, j) \in \mathcal{E}$ is associated with a set of Lagrange multipliers constraining $q_{ij}(x_i, x_j)$ to consistently marginalize to $q_i(x_i)$:

$$q_i(x_i) = \int_{\mathcal{X}_j} q_{ij}(x_i, x_j)\, dx_j \qquad \text{for all } x_i \in \mathcal{X}_i \tag{2.125}$$

Adding additional normalization constraints (as in eq. (2.105)) and taking derivatives, we recover a set of fixed point equations relating Lagrange multipliers and beliefs. Finally, as derived in detail by [340], the BP equations of Fig. 2.16 are *exactly* recovered by identifying messages as particular monotonic transformations of Lagrange multipliers.

The correspondence between loopy BP and the Bethe free energy has several important implications. First, the derivation sketched above shows that loopy BP fixed points correspond to stationary points of the Bethe free energy.[4] A more refined analysis shows that *stable* BP fixed points must be local minima [132]. Furthermore, because the Bethe free energy is bounded below, every graphical model has at least one BP fixed point [338, 340].

In general, the Bethe free energy is not convex, so there may be multiple BP solutions, and convergence is not guaranteed. However, for single cycles [133, 319] or graphs with sufficiently weak potentials [133, 143, 286], BP is guaranteed to have a single, unique global fixed point. In models where loopy BP exhibits instability, message schedules which pass messages along embedded chains or trees (as in Fig. 2.13), or step–size rules which damp message updates, can improve convergence [306]. Convergence dynamics are sometimes analyzed via the *computation tree* corresponding to the chosen message schedule [143, 155, 286, 319, 321]. Alternatively, *double–loop* algorithms have been developed which directly minimize the Bethe free energy at greater computational cost [134, 290, 341].

This derivation of loopy BP approximates the variational objective of eq. (2.96) in two ways. First, as mentioned earlier, the Bethe free energy (eq. (2.124)) uses an entropy approximation which is incorrect on graphs with cycles, and thus does not strictly bound the marginal likelihood. Second, the marginalization constraints of eq. (2.125) are insufficient to ensure that the estimated pseudo–marginals $\{q_{ij}(x_i, x_j) \mid (i,j) \in \mathcal{E}\}$ correspond to some valid global $q(x)$. For example, the constraint that every joint distribution has a positive definite covariance matrix is in general *not* implied by these marginalization conditions [311, 312]. Nevertheless, in many practical applications loopy BP produces accurate, effective belief estimates [101, 219, 320].

Theoretical Guarantees and Extensions

In the artificial intelligence community, the loopy BP algorithm was originally suggested by Pearl [231] (see [219] for a historical discussion). Then in 1993, *turbo codes* were independently discovered to achieve outstanding error–correction performance by coupling two randomly interleaved convolutional codes with an iterative decoder [23]. In the following years, the equivalence of this iterative approach and loopy BP was recognized [101, 201]. Graphical representations were then used to extend turbo (or sum–product) decoding to many other code families [175], rediscovering a class of *low density parity check (LDPC)* codes proposed in Gallager's 1960 doctoral thesis [102]. Subsequent refinements have led to long block–length codes which practically achieve

[4] Note that subtleties can arise with free energy analyses in graphical models containing hard constraints, for which potentials are not strictly positive [340].

the capacity of memoryless channels [22, 42]. This performance is theoretically understood through results which show that loopy BP becomes exact as cycles become arbitrarily long, and a corresponding *density evolution* algorithm which computes capacity thresholds for random code ensembles [244].

Inspired by its successes in iterative decoding, researchers have successfully applied loopy BP to a wide range of challenging learning and inference tasks [48, 95, 99, 219, 245, 283, 336]. Concurrently, the variational interpretation provided by the Bethe free energy has led to several important theoretical results and extensions. In particular, BP can be seen as a *reparameterization* algorithm which attempts to transform the given clique potentials into the canonical form of Prop. 2.3.1 [306]. Except in certain degenerate cases, this is impossible for graphs with cycles, and loopy BP will thus *not* provide exact posterior marginals. Interestingly, however, any loopy BP fixed point is consistent with respect to *every* tree embedded in the original graph (for examples, see Fig. 2.13). This analysis can be extended to bound the error in BP's approximate marginals [305, 306]. These results are stronger than those available for the mean field method, and support the empirical observation that loopy BP is typically more accurate and less prone to local optima [320].

Additional performance guarantees are available for Gaussian MRFs. If Gaussian BP converges, several different techniques can be used to guarantee exactness of the posterior means [155, 250, 306, 321]. However, the estimated variances are incorrect because correlations due to the graph's cycles are neglected. Intuitively, when all potentials are positively correlated or *attractive*, these variance estimates are over–confident [321]. Furthermore, convergence is guaranteed for a wide class of *walk–summable* models [155], or equivalently any graph whose pairwise potentials are normalizable.

More generally, variational interpretations of BP have led to the development of several extensions with improved accuracy. For example, the Bethe entropy of eq. (2.124) can be seen as the first terms of an expansion based on the Möbius inversion formula [125, 248]. Higher order terms directly account for relationships among larger groups of variables. Exploiting this, a *region graph* framework has been proposed which leads to better entropy approximations, and a corresponding family of *generalized belief propagation* algorithms [202, 338, 339, 340]. This approach generalizes the *Kikuchi free energies* [337] developed in the statistical physics community. The *expectation propagation* algorithm [135, 212, 213] provides a closely related method of incorporating higher–order dependencies (see [305] and [323] for unifying comparisons). In addition, a family of robust *reweighted* belief propagation algorithms have been derived from convex upper bounds on the log partition function [307, 310, 328, 329].

Finally, we note that the distributive structure underlying the BP algorithm can be generalized to any commutative semiring [4, 50, 255, 303]. In particular, a *max–product* variant of BP generalizes the Viterbi algorithm [90, 235] to efficiently compute optimal MAP estimates in tree–structured graphs [175, 231]. For graphs with cycles, there are some guarantees on max–product's accuracy [308, 322], and a reweighted extension can sometimes assure an optimal MAP solution [172, 309]. See [311] for an introduction

emphasizing variational interpretations of these methods.

■ 2.3.3 The Expectation Maximization Algorithm

In this section, we consider the MAP parameter estimation criterion motivated in Sec. 2.2.5. Given a model with parameters θ, and prior distribution $p(\theta \mid \lambda)$, we seek

$$\hat{\theta} = \arg\max_{\theta} p(\theta \mid y, \lambda) = \arg\max_{\theta} p(\theta \mid \lambda) \int_{\mathcal{X}} p(x, y \mid \theta)\ dx \tag{2.126}$$

As before, y are observations and x are latent variables. The *Expectation Maximization (EM)* algorithm [65] is an iterative parameter estimation scheme which tractably handles hidden or *missing* data x. We derive EM using the previously introduced variational framework, and discuss its application to learning in graphical models. For other introductions to the EM algorithm, see [98, 107, 161, 225].

As with other variational methods, the EM algorithm uses a distribution $q(x)$ over hidden variables to bound an otherwise intractable integral. Using Bayes' rule to expand the posterior distribution of eq. (2.126), we have

$$\log p(\theta \mid y, \lambda) = \log \int_{\mathcal{X}} p(x, y \mid \theta)\ dx + \log p(\theta \mid \lambda) - \log p(y \mid \lambda) \tag{2.127}$$

$$\geq \int_{\mathcal{X}} q(x) \log \frac{p(x, y \mid \theta)}{q(x)}\ dx + \log p(\theta \mid \lambda) - \log p(y \mid \lambda) \tag{2.128}$$

Here, we have applied Jensen's inequality as in our earlier variational bound on the marginal likelihood (eq. (2.94)). Regrouping terms and neglecting the final normalization constant, which does not depend on θ, we arrive at the following functional:

$$\mathcal{L}(q, \theta) = H(q) + \int_{\mathcal{X}} q(x) \log p(x, y \mid \theta)\ dx + \log p(\theta \mid \lambda) \tag{2.129}$$

Comparing to eq. (2.102), we see that $\mathcal{L}(q, \theta)$ equals a negative free energy [225] plus another term incorporating prior knowledge about the unknown parameters [107].

As in [225, 227], we derive the EM algorithm as a coordinate ascent iteration on $\mathcal{L}(q, \theta)$. In the expectation or *E–step*, the parameters θ are fixed and the optimal variational distribution $q(x)$ is determined. Then in the maximization or *M–step*, the lower bound defined by $q(x)$ is maximized with respect to the parameters:

$$q^{(t)} = \arg\max_{q} \mathcal{L}(q, \theta^{(t-1)}) \tag{2.130}$$

$$\theta^{(t)} = \arg\max_{\theta} \mathcal{L}(q^{(t)}, \theta) \tag{2.131}$$

It can be shown that the posterior probability of eq. (2.126) increases monotonically with each EM iteration, converging to some local maximum [65, 107, 225]. In the following sections, we discuss the implementation of these steps in greater detail.

Expectation Step

Fixing the parameters to some value $\theta^{(t-1)}$, provided either by the previous M–step or an initialization $\theta^{(0)}$, the E–step objective of eq. (2.130) becomes

$$q^{(t)} = \arg\max_q \left[H(q) + \int_{\mathcal{X}} q(x) \log p(x, y \mid \theta^{(t-1)}) \, dx \right] \tag{2.132}$$

Note the similarity of this equation to the variational objective underlying the mean field method (eq. (2.102)). Adding a Lagrange multiplier ensuring that $q(x)$ is properly normalized (as in eq. (2.105)) and taking derivatives, it is easily shown that

$$q^{(t)}(x) = p(x \mid y, \theta^{(t-1)}) \tag{2.133}$$

See [225] for a detailed derivation. We see that the E–step simply infers the posterior distribution of the hidden variables given the current parameters.

If $p(x, y \mid \theta)$ defines an exponential family, the expected values of that family's statistics are sufficient for the subsequent M–step. In graphical models, the E–step thus reduces to the problem of computing the posterior marginal distribution of each hidden variable (see Sec. 2.2.5). The variational derivation of the EM algorithm also justifies *incremental* E–steps, in which the expectations of only some variables are updated at each iteration [225]. In graphs where exact inference is intractable, mean field methods are commonly used to further bound the log–likelihood [161, 311, 331]. It is also tempting to use higher order variational methods, such as loopy BP, as approximate E–steps [98, 136]. In such cases, however, $\mathcal{L}(q, \theta)$ no longer strictly bounds the true posterior probability [311], and the resulting iteration may be unstable or inaccurate.

Maximization Step

Given the posterior distribution $q^{(t)}(x)$ determined in the previous E–step, the M–step objective of eq. (2.131) equals

$$\theta^{(t)} = \arg\max_\theta \left[\log p(\theta \mid \lambda) + \int_{\mathcal{X}} q^{(t)}(x) \log p(x, y \mid \theta) \, dx \right] \tag{2.134}$$

Up to an additive constant independent of θ, the likelihood term in eq. (2.134) equals $-D\big(q^{(t)} \,||\, p_\theta\big)$. If θ parameterizes an exponential family and the prior distribution is uninformative, Prop. 2.1.2 then shows that $\theta^{(t)}$ should be chosen to match the appropriate sufficient statistics of $q^{(t)}$. Similarly, conjugate priors $p(\theta \mid \lambda)$ are easily handled by appropriately biasing these statistics (see Prop. 2.1.4). More generally, partial M–steps can be used which increase, but do not maximize, the current likelihood bound [107, 225].

In directed Bayesian networks, the M–step can often be computed in closed form [37, 50, 98, 128]. Consider the following directed factorization:

$$p(x \mid \theta) = \prod_{i \in \mathcal{V}} p\big(x_i \mid x_{\Gamma(i)}, \theta_i\big) \tag{2.135}$$

Here, θ_i parameterizes the transition distribution for the i^{th} node, and we have not explicitly indicated which nodes correspond to observations y. If each transition is assigned a *meta independent* [50, 59] prior $p(\theta_i \mid \lambda_i)$, the objective of eq. (2.134) equals

$$\theta^{(t)} = \arg\max_{\theta} \sum_{i \in \mathcal{V}} \iint q^{(t)}(x_i, x_{\Gamma(i)}) \log p\big(x_i \mid x_{\Gamma(i)}, \theta_i\big)\, dx_i\, dx_{\Gamma(i)} + \log p(\theta_i \mid \lambda_i) \tag{2.136}$$

The parameters associated with different nodes are thus *decoupled*, and can be estimated independently. This optimization is similarly tractable for many models in which parameters are shared among multiple transition densities [235].

In undirected graphical models, parameter estimation is more challenging. Consider a factor graph parameterized as in eq. (2.68), and assume for simplicity that the reference measure $\nu(x) = 1$. Then, if each clique potential is assigned a meta independent prior $p(\theta_f \mid \lambda_f)$, the M–step objective equals

$$\theta^{(t)} = \arg\max_{\theta} \sum_{f \in \mathcal{F}} \left[\sum_{a \in \mathcal{A}_f} \theta_{fa} \int q^{(t)}(x_f) \phi_{fa}(x_f)\, dx_f + \log p(\theta_f \mid \lambda_f) \right] - \Phi(\theta) \tag{2.137}$$

In contrast with eq. (2.136), the log partition function $\Phi(\theta)$ induces non–local dependencies among the parameters. When the corresponding graph is decomposable or triangulated, junction tree representations can be used to efficiently estimate parameters [59, 177]. Otherwise, computationally demanding numerical methods are required, often implemented via one of several *iterative scaling* algorithms [53, 56, 62, 177, 227, 268, 290]. A recently proposed family of convex *upper* bounds on the log partition function can be used for approximate undirected parameter estimation [307, 310].

■ 2.4 Monte Carlo Methods

By using random samples to simulate probabilistic models, *Monte Carlo methods* [9, 107, 192] provide complementary solutions to the learning and inference tasks described in Sec. 2.2.5. In contrast with variational approaches, they are guaranteed to give arbitrarily precise estimates with sufficient computation. In practice, however, care must be taken to design efficient algorithms so that reliable, accurate estimates can be obtained at a tractable computational cost.

Let $p(x)$ denote some target density with sample space $\mathcal{X}$. Many inference tasks, including the calculation of marginal densities and sufficient statistics, can be expressed as the expected value $\mathbb{E}_p[f(x)]$ of an appropriately chosen function [9, 192]. Suppose that $p(x)$ is difficult to analyze explicitly, but that L independent samples $\{x^{(\ell)}\}_{\ell=1}^{L}$ are

available. The desired statistic can then be approximated as follows:

$$\mathbb{E}_p[f(x)] = \int_{\mathcal{X}} f(x)p(x)\,dx \tag{2.138}$$

$$\approx \frac{1}{L}\sum_{\ell=1}^{L} f(x^{(\ell)}) = \mathbb{E}_{\tilde{p}}[f(x)] \tag{2.139}$$

Here, $\tilde{p}(x)$ is the empirical density (see eq. (2.13)) corresponding to the L samples, as illustrated in Fig. 2.17(a). This estimate is unbiased, and converges to $\mathbb{E}_p[f(x)]$ almost surely as $L \to \infty$. Furthermore, its error is asymptotically Gaussian, with variance determined by $\mathbb{E}_p\left[f^2(x)\right]$ rather than the dimensionality of the sample space [9].

In graphical models, exact samples can be drawn from the posterior distribution $p(x \mid y)$ using a variant of the junction tree algorithm (see Sec. 2.3.2). First, some clique is chosen as the tree's root, and a sample is drawn from its corresponding marginal. The values of neighboring cliques are then recursively sampled from the appropriate conditional densities [50]. For many graphs, however, the junction tree's cliques are too large, and exact sampling is intractable. The following sections describe several Monte Carlo methods which allow approximate samples to be drawn more efficiently.

■ 2.4.1 Importance Sampling

Importance sampling provides an alternative to direct Monte Carlo approximation in cases where sampling from $p(x)$ is difficult. We assume that it is possible to evaluate $p(x) = \bar{p}(x)/Z$ up to some normalization constant Z. Let $q(x)$ denote a *proposal distribution* which is absolutely continuous with respect to $p(x)$, so that $p(\bar{x}) = 0$ whenever $q(\bar{x}) = 0$. The expectation of eq. (2.138) can then be rewritten as follows:

$$\mathbb{E}_p[f(x)] = \frac{\int_{\mathcal{X}} f(x)w(x)q(x)\,dx}{\int_{\mathcal{X}} w(x)q(x)\,dx} \qquad w(x) = \frac{\bar{p}(x)}{q(x)} \tag{2.140}$$

The denominator of eq. (2.140) implicitly defines the unknown normalization constant via the *weight function* $w(x)$. Given L independent samples $\{x^{(\ell)}\}_{\ell=1}^{L}$ from the proposal density $q(x)$, we approximate this expectation as

$$\mathbb{E}_p[f(x)] \approx \sum_{\ell=1}^{L} w^{(\ell)} f(x^{(\ell)}) \qquad w^{(\ell)} \triangleq \frac{w(x^{(\ell)})}{\sum_{m=1}^{L} w(x^{(m)})} \tag{2.141}$$

Importance sampling thus estimates the target expectation via a collection of *weighted* samples $\{(x^{(\ell)}, w^{(\ell)})\}_{\ell=1}^{L}$ from the proposal density $q(x)$. Under mild assumptions, this estimate is asymptotically consistent [9], and its variance is smallest when the proposal density $q(x) \propto |f(x)|p(x)$. Fig. 2.17 illustrates weighted samples drawn from two different importance approximations to a bimodal target distribution.

The practical effectiveness of importance sampling critically depends on the chosen importance density. When $q(x)$ assigns low probability to likely regions of the target sample space, importance estimates can be extremely inaccurate. For example,

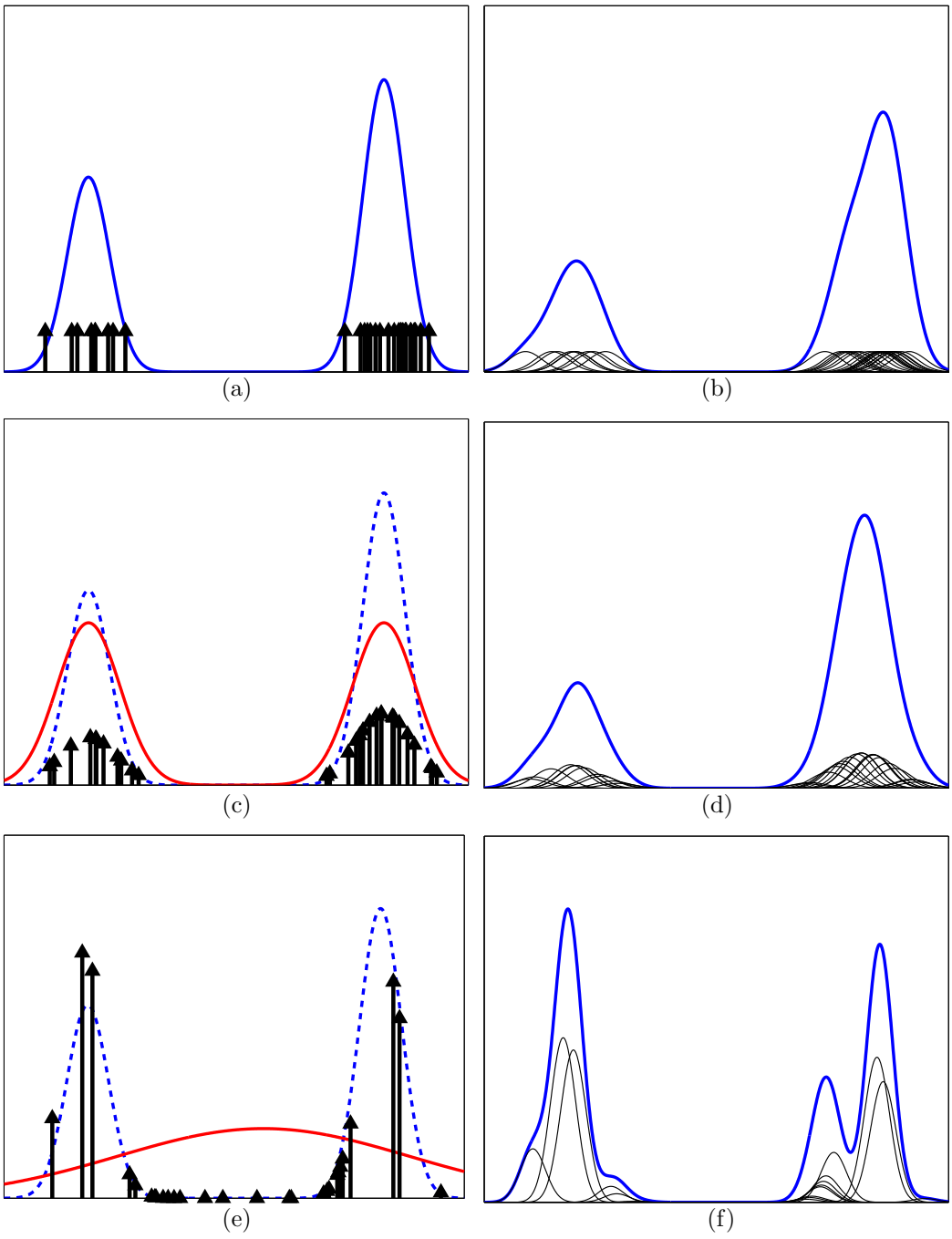


Figure 2.17. Monte Carlo estimates based on 30 samples (arrows) from one–dimensional proposal distributions (left column), and corresponding kernel density estimates (right column) constructed via likelihood cross–validation. (a) Target density (solid), and unweighted direct samples. (b) Kernel density (thick blue line) estimated from Gaussian kernels (thin black lines). (c) A mixture proposal distribution (solid) closely matched to the target density (dashed), and importance weighted samples. (d) Kernel density estimated from weighted Gaussian kernels. (e) A Gaussian proposal distribution (solid) with mean and variance matching the target density (dashed), and weighted samples. (f) Kernel density with artifacts from the Gaussian proposal's widely varying importance weights.

the poorly matched proposal distribution of Fig. 2.17(e) causes many samples to have negligible weight, greatly reducing the *effective* sample size. Heavy–tailed proposal distributions, which are more dispersed than the target density, typically provide greater robustness [107, 192]. For high–dimensional problems, however, designing good proposals is extremely challenging, since even minor discrepancies can produce widely varying importance weights. In graphical models, importance sampling is thus typically used as a building block within more sophisticated Monte Carlo methods.

■ 2.4.2 Kernel Density Estimation

In some applications of Monte Carlo methods, an explicit estimate $\hat{p}(x)$ of the target density $p(x)$ is desired, rather than a summary statistic as in eq. (2.138). *Nonparametric* density estimators avoid choosing a particular form for $\hat{p}(x)$, and allow the complexity of the estimated density to grow as more samples are observed. Given L independent samples $\{x^{(\ell)}\}_{\ell=1}^{L}$, the corresponding *kernel* or *Parzen window* density estimate [230, 263] can be written as follows:

$$\hat{p}(x) = \sum_{\ell=1}^{L} w^{(\ell)} \mathcal{N}(x; x^{(\ell)}, \Lambda) \tag{2.142}$$

This estimator uses a Gaussian kernel function to smooth the raw sample set, intuitively placing more probability mass in regions with many samples. Other kernel functions may also be considered [263], but we focus on the Gaussian case. If these samples are drawn from the target density $p(x)$, the weights are set uniformly to $w^{(\ell)} = 1/L$. More generally, they could come from an importance sampling scheme [220] as in eq. (2.141).

The kernel density estimate of eq. (2.142) depends on the bandwidth or covariance Λ of the Gaussian kernel function. There is an extensive literature on methods for automatic bandwidth selection [263]. For example, the simple "rule of thumb" method combines a robust covariance estimate with an asymptotic formula which assumes the target density is Gaussian. While fast to compute, it often oversmooths multimodal distributions. In such cases, more sophisticated cross–validation schemes can improve performance [263]. Fig. 2.17 illustrates kernel density estimates constructed from three different proposal distributions, with bandwidth automatically selected via likelihood cross–validation. Note that inaccurate importance densities produce less reliable density estimators (compare Fig. 2.17(d) and Fig. 2.17(f)).

■ 2.4.3 Gibbs Sampling

We now describe a family of iterative, *Markov chain Monte Carlo (MCMC)* methods which draw samples from an otherwise intractable target density $p(x)$. Starting from some initial global configuration $x^{(0)} \in \mathcal{X}$, subsequent states are determined via a first–order Markov process:

$$x^{(t)} \sim q(x \mid x^{(t-1)}) \qquad\qquad t = 1, 2, \ldots \tag{2.143}$$

The transition distribution $q(\cdot \mid \cdot)$ is designed so that the resulting Markov chain is irreducible and aperiodic, with $p(x)$ as its unique equilibrium distribution [9]. Thus, after many iterations T the state will be approximately distributed as $x^{(T)} \sim p(x)$, providing a sample from the desired target density.

The *Metropolis–Hastings* algorithm [9, 107] provides a flexible, general framework for constructing Markov chains with a desired equilibrium distribution $p(x)$. In this section, we describe the *Gibbs sampler* [106, 108, 196], a special case that is particularly well suited to state spaces with internal structure. Let $x = (x_1, \ldots, x_N)$ denote a decomposition of the joint sample space into N variables. Gibbs samplers assume that it is tractable to sample from the conditional distribution of one of these variables given the other $(N-1)$. At iteration t, a particular variable $i(t)$ is selected for resampling, and the rest are held constant:

$$x_i^{(t)} \sim p(x_i \mid x_j^{(t-1)}, j \neq i) \qquad i = i(t) \tag{2.144}$$

$$x_j^{(t)} = x_j^{(t-1)} \qquad j \neq i(t) \tag{2.145}$$

If these sampling updates are iterated so that all variables are resampled infinitely often, mild conditions ensure $x^{(t)}$ will converge to a sample from $p(x)$ as $t \to \infty$ [9, 108, 186]. Randomly permuting the order in which variables are resampled, rather than repeating a single fixed order, often improves the rate of convergence [246].

Although there exist polynomial bounds on the time required for some MCMC methods to *mix* to the target equilibrium distribution [9, 186], it can be difficult to guarantee or diagnose convergence in high–dimensional models [192]. In practice, it is often useful to run the sampler from several random initializations, and compare problem–dependent summary statistics. If slow mixing is observed, one can consider *blocked Gibbs samplers* which, rather than sampling individual variables, jointly resample small groups of variables which are thought to be strongly correlated [9, 185, 246].

For some models, Gibbs samplers are best implemented via *auxiliary variable* methods [9]. These algorithms are based on a joint distribution $p(x, z)$ which is designed to marginalize to the target density $p(x)$. In the simplest case, auxiliary variables z are chosen so that the following conditional densities are tractable:

$$x^{(t)} \sim p(x \mid z^{(t-1)}) \tag{2.146}$$

$$z^{(t)} \sim p(z \mid x^{(t)}) \tag{2.147}$$

More generally, eq. (2.146) may be replaced by several Gibbs sampling steps as in eqs. (2.144, 2.145). Any joint sample $(x^{(T)}, z^{(T)})$ from the resulting Markov chain then also provides an approximate sample $x^{(T)}$ from the target density of interest. Some auxiliary variable methods, such as the hybrid Monte Carlo algorithm [9, 107, 192], are designed to improve the convergence rate of the resulting Markov chain. Alternatively, auxiliary variable methods sometimes lead to tractable Gibbs samplers for models in which direct conditional densities lack simple forms [222]. Several algorithms developed in this thesis exploit this technique.

Sampling in Graphical Models

The Gibbs sampler's use of partitioned state spaces is ideally suited for inference in graphical models [98, 108, 196, 231]. For example, consider a pairwise MRF $p(x \mid y)$ parameterized as in eq. (2.97). By the Markov properties discussed in Sec. 2.2.2, the posterior distribution of x_i depends only on the values at neighboring nodes:

$$p\big(x_i \mid x_{\mathcal{V}\setminus i}, y\big) = p\big(x_i \mid x_{\Gamma(i)}, y\big) \propto \psi_i(x_i, y) \prod_{j \in \Gamma(i)} \psi_{ij}(x_i, x_j) \tag{2.148}$$

When the clique potentials are drawn from exponential families, it is typically easy to sample from this conditional density. Iterating such resampling as in eqs. (2.144, 2.145), we obtain a Gibbs sampler providing Monte Carlo estimates of the posterior marginals motivated in Sec. 2.2.5. Alternatively, the related *simulated annealing* method [9, 108] can be used to search for approximate MAP estimates.

Gibbs sampling is also used to estimate posterior distributions for model parameters θ (see eq. (2.91)). First, hidden variables are sampled given fixed parameters as in eq. (2.148). Then, conditioned on these hidden variables, conjugate priors $p(\theta \mid \lambda)$ typically allow individual parameters to be tractably resampled [37, 50, 106, 128]. Alternating between sampling $x^{(t)} \sim p\big(x \mid \theta^{(t-1)}, y\big)$ and $\theta^{(t)} \sim p\big(\theta \mid x^{(t)}, y, \lambda\big)$, we can estimate statistics of the joint posterior $p(x, \theta \mid y, \lambda)$. The BUGS software package uses this method to do Bayesian learning and inference in directed graphical models [115].

Gibbs Sampling for Finite Mixtures

To illustrate the Gibbs sampler, we consider a K–component exponential family mixture model, as introduced in Sec. 2.2.4 (see Fig. 2.9). While the data $x = \{x_i\}_{i=1}^N$ are directly observed, the latent cluster $z_i \in \{1, \dots, K\}$ associated with each data point is unknown. The simplest mixture model Gibbs sampler thus alternates between sampling cluster indicators $z = \{z_i\}_{i=1}^N$, mixture weights π, and cluster parameters $\{\theta_k\}_{k=1}^K$. We assume the hyperparameters α and λ are set to fixed, known constants.

Given fixed cluster weights and parameters, the indicator variables are conditionally independent. Let $z_{\setminus i}$ denote the set of all cluster assignments excluding z_i. Applying Bayes' rule to the generative model of eq. (2.79), we then have

$$p(z_i = k \mid z_{\setminus i}, x, \pi, \theta_1, \dots, \theta_K) = p(z_i = k \mid x_i, \pi, \theta_1, \dots, \theta_K) \tag{2.149}$$

$$\propto \pi_k f(x_i \mid \theta_k) \tag{2.150}$$

Here, the simplification of eq. (2.149) follows from the Markov properties of the directed graph in Fig. 2.9. By evaluating the likelihood of x_i with respect to each current cluster, we may thus resample z_i in $\mathcal{O}(K)$ operations.

As discussed in detail by [96], the mixture weights π and parameters $\{\theta_k\}_{k=1}^K$ are mutually independent conditioned on the indicator variables z:

$$p(\pi, \theta_1, \dots, \theta_K \mid z, x, \alpha, \lambda) = p(\pi \mid z, \alpha) \prod_{k=1}^{K} p(\theta_k \mid \{x_i \mid z_i = k\}, \lambda) \tag{2.151}$$

Given mixture weights $\pi^{(t-1)}$ and cluster parameters $\{\theta_k^{(t-1)}\}_{k=1}^K$ from the previous iteration, sample a new set of mixture parameters as follows:

1. Independently assign each of the N data points x_i to one of the K clusters by sampling the indicator variables $z = \{z_i\}_{i=1}^N$ from the following multinomial distributions:
$$z_i^{(t)} \sim \frac{1}{Z_i} \sum_{k=1}^K \pi_k^{(t-1)} f(x_i \mid \theta_k^{(t-1)})\, \delta(z_i, k) \qquad Z_i = \sum_{k=1}^K \pi_k^{(t-1)} f(x_i \mid \theta_k^{(t-1)})$$
2. Sample new mixture weights according to the following Dirichlet distribution:
$$\pi^{(t)} \sim \mathrm{Dir}(N_1 + \alpha/K, \ldots, N_K + \alpha/K) \qquad N_k = \sum_{i=1}^N \delta(z_i^{(t)}, k)$$
3. For each of the K clusters, independently sample new parameters from the conditional distribution implied by those observations currently assigned to that cluster:
$$\theta_k^{(t)} \sim p(\theta_k \mid \{x_i \mid z_i^{(t)} = k\}, \lambda)$$
When λ defines a conjugate prior, this posterior distribution is given by Prop. 2.1.4.

Algorithm 2.1. Direct Gibbs sampler for a K component exponential family mixture model, as defined in Fig. 2.9. Each iteration resamples the cluster assignments for all N observations $x = \{x_i\}_{i=1}^N$ once, and uses these updated assignments to choose new mixture parameters.

Assuming α is the precision of a symmetric Dirichlet prior, the posterior distribution of the mixture weights π is also Dirichlet (see eq. (2.45)), with hyperparameters determined by the number of observations N_k currently assigned to each cluster:

$$p(\pi \mid z, \alpha) = \mathrm{Dir}(N_1 + \alpha/K, \ldots, N_K + \alpha/K) \qquad N_k = \sum_{i=1}^N \delta(z_i, k) \tag{2.152}$$

Standard methods may then be used to sample new cluster weights [107]. Intuitively, eq. (2.151) shows that the posterior distribution of the k^{th} cluster's parameters θ_k depends only on those observations currently assigned to it. If λ parameterizes a conjugate prior, Prop. 2.1.4 provides a closed form for this posterior. For example, when clusters are Gaussian, $\theta_k = (\mu_k, \Lambda_k)$ follows a normal–inverse–Wishart density (see Sec. 2.1.4).

Algorithm 2.1 summarizes the Gibbs sampler implied by these conditional distributions. We initialize the mixture parameters according to their priors $\pi^{(0)} \sim \mathrm{Dir}(\alpha)$, $\theta_k^{(0)} \sim H(\lambda)$. At each iteration, $\mathcal{O}(NK)$ operations are needed to resample all N indicator variables. Note that because these indicators are mutually independent given known parameters, the order of this resampling is unimportant. To allow fast parameter resampling, we cache sufficient statistics (as in Thm. 2.1.2) of the data assigned to each cluster, and recursively update these statistics as assignments change.

In Fig. 2.18, we use the Gibbs sampler of Alg. 2.1 to fit a mixture of $K = 4$ two–dimensional Gaussians to $N = 300$ observations. Each Gaussian cluster is assigned a weakly informative normal–inverse–Wishart prior, so that the posterior distribution of $\theta_k = (\mu_k, \Lambda_k)$ can be determined as described in Sec. 2.1.4. The columns of Fig. 2.18

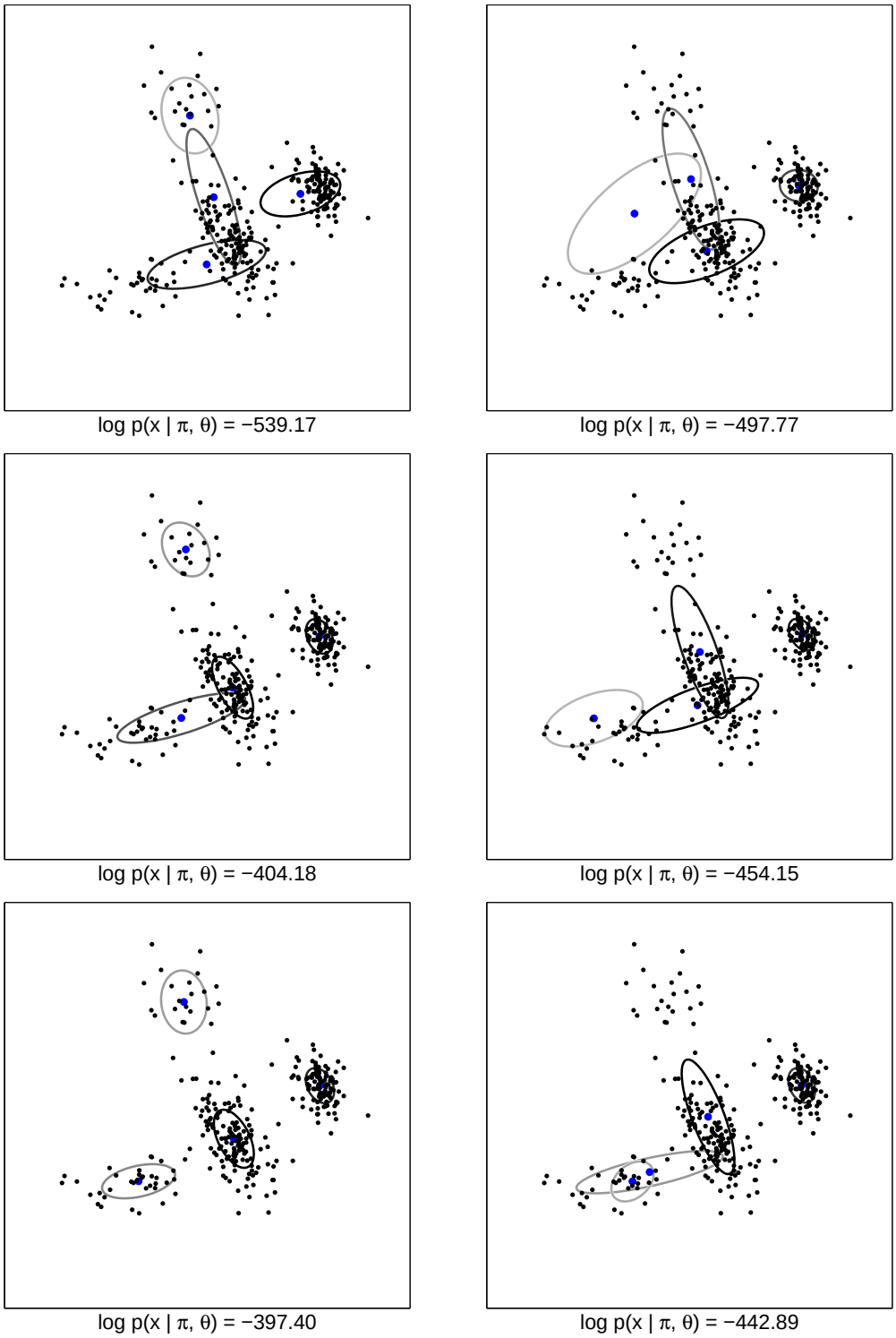


Figure 2.18. Learning a mixture of $K = 4$ Gaussians using the Gibbs sampler of Alg. 2.1. Columns show the current parameters after T=2 (top), T=10 (middle), and T=50 (bottom) iterations from two random initializations. Each plot is labeled by the current data log–likelihood.

compare two different random initializations. Because we use vague priors, the data log–likelihood provides a reasonable convergence measure:

$$\log p(x \mid \pi, \theta_1, \ldots, \theta_K) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k f(x_i \mid \theta_k) \right) \tag{2.153}$$

We see that the Gibbs sampler effectively implements a random walk, which gradually moves towards parameters with higher posterior probability. Although the induced Markov chain may converge quickly (left column), it sometimes remains trapped in locally optimal regions of the parameter space for many iterations (right column). Fig. 2.20 compares this behavior to a more sophisticated Rao–Blackwellized sampler developed in the following section.

■ 2.4.4 Rao–Blackwellized Sampling Schemes

In models which impose structured dependencies on multiple latent variables, we can often construct tractable Monte Carlo procedures which improve on the basic estimator of eq. (2.139). Let $p(x, z)$ denote a target distribution on two random variables $x \in \mathcal{X}$, $z \in \mathcal{Z}$. Given L independent samples $\{(x^{(\ell)}, z^{(\ell)})\}_{\ell=1}^{L}$ from this joint distribution, the simplest approximation of a statistic $f(x, z)$ equals

$$\mathbb{E}_p[f(x,z)] = \int_{\mathcal{Z}} \int_{\mathcal{X}} f(x,z) p(x,z)\, dx\, dz \tag{2.154}$$

$$\approx \frac{1}{L} \sum_{\ell=1}^{L} f(x^{(\ell)}, z^{(\ell)}) = \mathbb{E}_{\tilde{p}}[f(x,z)] \tag{2.155}$$

Suppose, however, that the conditional density $p(x \mid z)$ has a tractable analytic form. In this case, we can consider the following alternative estimator:

$$\mathbb{E}_p[f(x,z)] = \int_{\mathcal{Z}} \int_{\mathcal{X}} f(x,z) p(x \mid z)\, p(z)\, dx\, dz \tag{2.156}$$

$$= \int_{\mathcal{Z}} \left[\int_{\mathcal{X}} f(x,z) p(x \mid z)\ dx \right] p(z)\, dz \tag{2.157}$$

$$\approx \frac{1}{L} \sum_{\ell=1}^{L} \int_{\mathcal{X}} f(x, z^{(\ell)}) p(x \mid z^{(\ell)})\ dx = \mathbb{E}_{\tilde{p}}[\mathbb{E}_p[f(x,z) \mid z]] \tag{2.158}$$

The estimators of eqs. (2.155) and (2.158) are both unbiased, and converge to $\mathbb{E}_p[f(x, z)]$ almost surely as $L \to \infty$. Intuitively, however, the marginalized estimate of eq. (2.158) should be more reliable [9, 39, 106], because the underlying sample space $\mathcal{Z}$ is smaller than the original space $\mathcal{X} \times \mathcal{Z}$.

In classical statistics, the *Rao–Blackwell Theorem* [167, 242] establishes the importance of sufficient statistics in parameter estimation. In particular, it allows *minimum variance unbiased estimators* to be designed by conditioning simpler estimators with

respect to appropriate statistics. The Rao–Blackwell Theorem is derived from the following relationship between conditional and unconditional variance, which is also more broadly applicable.

Theorem 2.4.1 (Rao-Blackwell). *Let x and z be dependent random variables, and $f(x, z)$ a scalar statistic. Consider the marginalized statistic $\mathbb{E}_x[f(x, z) \mid z]$, which is a function solely of z. The unconditional variance $\mathrm{Var}_{xz}[f(x, z)]$ is then related to the variance of the marginalized statistic as follows:*

$$\mathrm{Var}_{xz}[f(x,z)] = \mathrm{Var}_z[\mathbb{E}_x[f(x,z) \mid z]] + \mathbb{E}_z[\mathrm{Var}_x[f(x,z) \mid z]] \tag{2.159}$$

$$\geq \mathrm{Var}_z[\mathbb{E}_x[f(x,z) \mid z]] \tag{2.160}$$

Proof. Using the iterated expectations [229, 242] induced by the conditional factorization $p(x, z) = p(x \mid z)\, p(z)$, the unconditional variance of $f(x, z)$ equals

$$\begin{aligned}\mathrm{Var}_{xz}[f(x,z)] &= \mathbb{E}_{xz}\left[f(x,z)^2\right] - \mathbb{E}_{xz}[f(x,z)]^2 \\ &= \mathbb{E}_z\left[\mathbb{E}_x\left[f(x,z)^2 \mid z\right]\right] - \mathbb{E}_z[\mathbb{E}_x[f(x,z) \mid z]]^2\end{aligned}$$

Subtracting and adding $\mathbb{E}_z[\mathbb{E}_x[f(x, z) \mid z]^2]$ and regrouping terms, we may then verify eq. (2.159). Equation (2.160) follows from the non–negativity of $\mathrm{Var}_x[f(x, z) \mid z]$. □

As established by eq. (2.160), analytic marginalization of some variables from a joint distribution *always* reduces the variance of later estimates. Applying this result, the so–called *Rao–Blackwellized* Monte Carlo estimator [9, 39] of eq. (2.158) has lower variance than the direct estimator of eq. (2.155). Intuitively, eq. (2.159) shows that marginalization of x is most useful when the average conditional variance of x is large.

Rao–Blackwellization also plays an important role in other, more sophisticated Monte Carlo methods. In particular, the variance inequality of Thm. 2.4.1 can be generalized to bound the variance of marginalized importance estimators (see Sec. 2.4.1). As we discuss in Chap. 3, this approach has been used to design Rao–Blackwellized improvements of standard particle filters [71, 73]. Similarly, Rao–Blackwellization may dramatically improve the efficiency and accuracy of Gibbs samplers [39, 106, 185]. In particular, for hierarchical models based on conjugate priors, Prop. 2.1.4 can often be used to integrate over latent parameters in closed form. Importantly, the variance reduction guaranteed by Thm. 2.4.1 generalizes to estimates based on the correlated samples produced by a Gibbs sampler [185].

Rao–Blackwellized Gibbs Sampling for Finite Mixtures

To illustrate the design of Rao–Blackwellized samplers, we revisit the mixture model Gibbs sampler summarized in Alg. 2.1. Given fixed cluster indicators z, we show that conjugate priors allow mixture weights π and parameters $\{\theta_k\}_{k=1}^K$ to be analytically marginalized. We may then directly determine the predictive distribution of z_i given the other cluster assignments $z_{\setminus i}$, and construct a more efficient sampler.

Consider the K–component exponential family mixture model of Fig. 2.9, and assume $H(\lambda)$ specifies a conjugate prior for the clusters θ_k. Integrating over the parameters π and $\{\theta_k\}_{k=1}^K$, the model's Markov structure implies the following factorization:

$$p(z_i \mid z_{\setminus i}, x, \alpha, \lambda) \propto p(z_i \mid z_{\setminus i}, \alpha)\, p(x_i \mid z, x_{\setminus i}, \lambda) \tag{2.161}$$

The first term arises from the marginalization of the mixture weights π. Because these weights have a symmetric Dirichlet prior, this predictive distribution is given by eq. (2.46) of Sec. 2.1.3, so that

$$p(z_i = k \mid z_{\setminus i}, \alpha) = \frac{N_k^{-i} + \alpha/K}{N - 1 + \alpha} \qquad N_k^{-i} = \sum_{j \neq i} \delta(z_j, k) \tag{2.162}$$

Note that N_k^{-i} counts the number of observations currently assigned to the k^{th} cluster excluding x_i, the datum whose assignment z_i is being resampled. Similarly, the likelihood term of eq. (2.161) depends on the current assignments $z_{\setminus i}$ as follows:

$$p(x_i \mid z_i = k, z_{\setminus i}, x_{\setminus i}, \lambda) = p(x_i \mid \{x_j \mid z_j = k, j \neq i\}, \lambda) \tag{2.163}$$

For each of the K possible values of z_i, eq. (2.163) equals the predictive likelihood (as in eq. (2.19)) of x_i given the other data currently assigned to that cluster. Because $H(\lambda)$ is conjugate to θ_k, these likelihoods can be analytically determined from Prop. 2.1.4. For example, Gaussian clusters lead to Student–t predictive distributions (see Sec. 2.1.4), which can usually be approximated by the moment–matched Gaussian of eq. (2.64).

Algorithm 2.2 provides one possible Rao–Blackwellized Gibbs sampler based on these predictive distributions. As with the direct Gibbs sampler of Alg. 2.1, $\mathcal{O}(NK)$ operations are required to resample N cluster assignments. To improve the Markov chain's convergence rate, each iteration resamples indicator variables in a different, randomly chosen order [246]. Fast predictive likelihood evaluation is achieved by caching the sufficient statistics $\boldsymbol{\phi}(x)$ (as in Thm. 2.1.2) associated with each cluster. When an observation x_i is reassigned, these statistics are easily updated by subtracting $\boldsymbol{\phi}(x_i)$ from the previous cluster $z_i^{(t-1)}$, and adding $\boldsymbol{\phi}(x_i)$ to the newly chosen cluster $z_i^{(t)}$. We initialize the sampler by sequentially choosing $z_i^{(0)}$ conditioned on $\{z_1^{(0)}, \ldots, z_{i-1}^{(0)}\}$.

In Fig. 2.19, we use the Rao–Blackwellized Gibbs sampler of Alg. 2.2 to fit a mixture of $K = 4$ two–dimensional Gaussians to $N = 300$ observations. Compared to the direct Gibbs sampler of Alg. 2.1 (tested on identical data in Fig. 2.18), the Rao–Blackwellized sampler has less random variation from iteration to iteration. Fig. 2.20 compares the data log–likelihoods (eq. (2.153)) produced by these two algorithms from 100 different random initializations. Typically, the Rao–Blackwellized sampler much more rapidly reaches parameters with high posterior probability. Intuitively, this happens because marginalized, predictive likelihoods implicitly update the model's parameters after every indicator reassignment, rather than once per iteration as in Alg. 2.1. However, the two samplers have similar worst case performance, and may occasionally remain in local

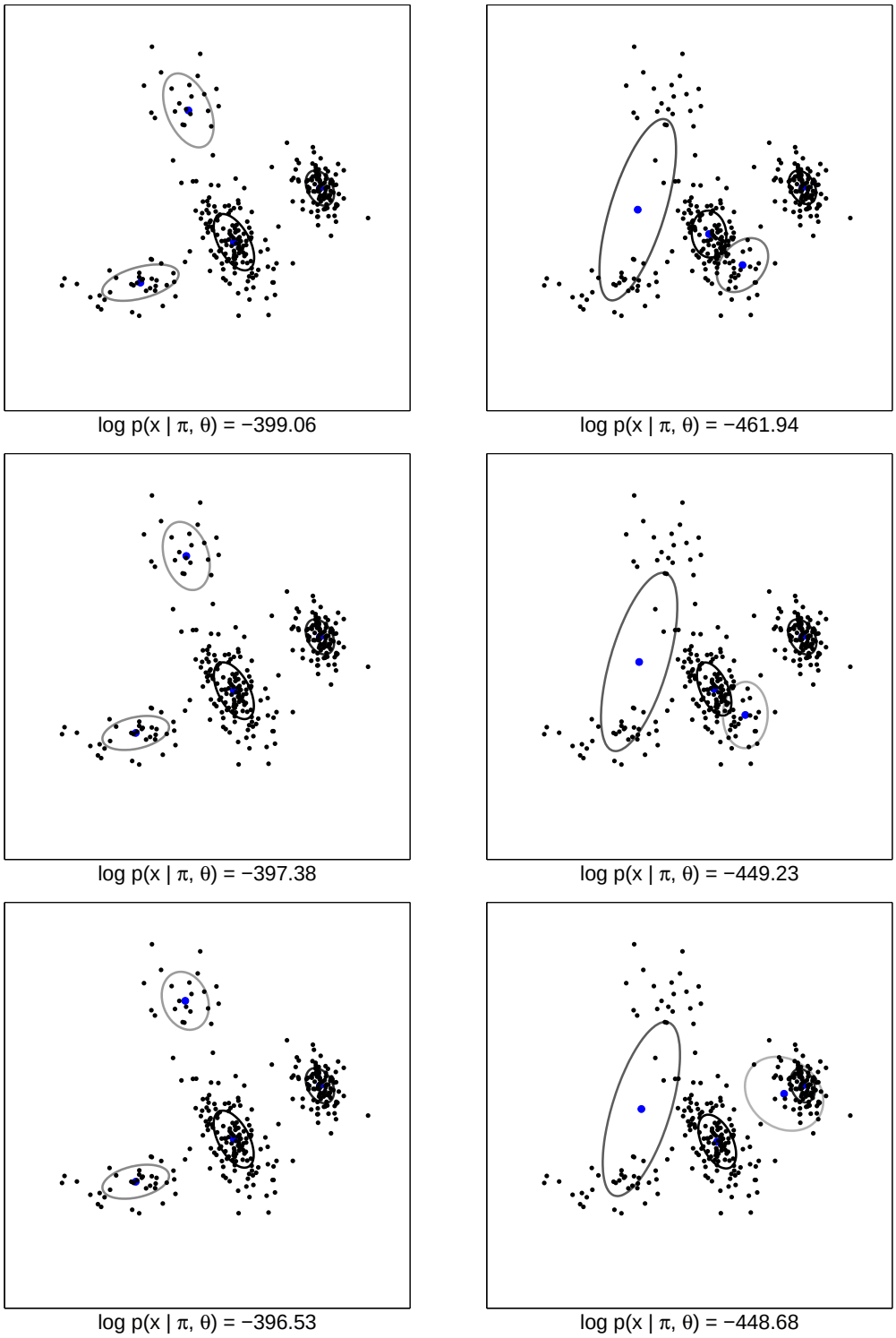


Figure 2.19. Learning a mixture of $K = 4$ Gaussians using the Rao–Blackwellized Gibbs sampler of Alg. 2.2. Columns show the current parameters after T=2 (top), T=10 (middle), and T=50 (bottom) iterations from two random initializations. Each plot is labeled by the current data log–likelihood.

Given previous cluster assignments $z^{(t-1)}$, sequentially sample new assignments as follows:

1. Sample a random permutation $\tau(\cdot)$ of the integers $\{1, \ldots, N\}$.
2. Set $z = z^{(t-1)}$. For each $i \in \{\tau(1), \ldots, \tau(N)\}$, sequentially resample z_i as follows:
   (a) For each of the K clusters, determine the predictive likelihood
   $$f_k(x_i) = p(x_i \mid \{x_j \mid z_j = k, j \neq i\}, \lambda)$$
   This likelihood can be computed from cached sufficient statistics via Prop. 2.1.4.
   (b) Sample a new cluster assignment z_i from the following multinomial distribution:
   $$z_i \sim \frac{1}{Z_i} \sum_{k=1}^{K} (N_k^{-i} + \alpha/K) f_k(x_i) \delta(z_i, k) \qquad Z_i = \sum_{k=1}^{K} (N_k^{-i} + \alpha/K) f_k(x_i)$$
   N_k^{-i} is the number of other observations assigned to cluster k (see eq. (2.162)).
   (c) Update cached sufficient statistics to reflect the assignment of x_i to cluster z_i.
3. Set $z^{(t)} = z$. Optionally, mixture parameters may be sampled via steps 2–3 of Alg. 2.1.

Algorithm 2.2. Rao–Blackwellized Gibbs sampler for a K component exponential family mixture model, as defined in Fig. 2.9. Each iteration sequentially resamples the cluster assignments for all N observations $x = \{x_i\}_{i=1}^N$ in a different random order. Mixture parameters are integrated out of the sampling recursion using cached sufficient statistics of the parameters assigned to each cluster.

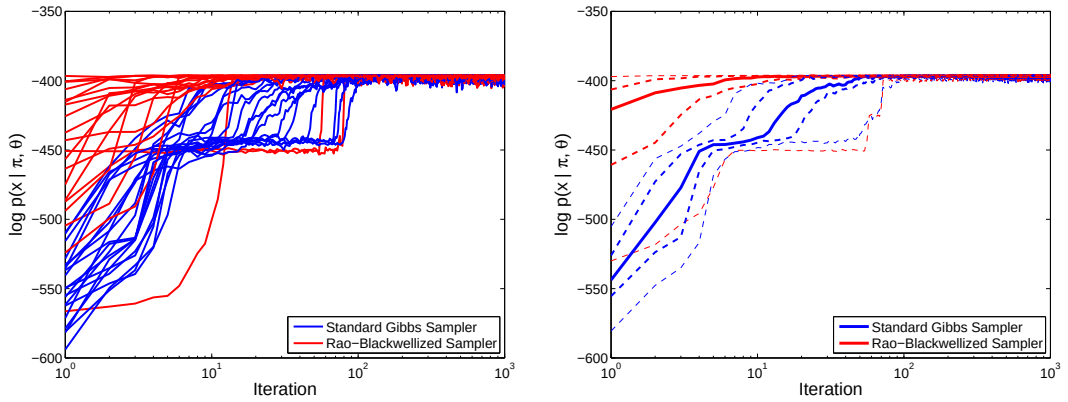


Figure 2.20. Comparison of standard (Alg. 2.1, dark blue) and Rao–Blackwellized (Alg. 2.2, light red) Gibbs samplers for a mixture of $K = 4$ two–dimensional Gaussians. We compare data log–likelihoods at each of 1000 iterations for the single $N = 300$ point dataset of Figs. 2.18 and 2.19. *Left:* Log–likelihood sequences for 20 different random initializations of each algorithm. *Right:* From 100 different random initializations, we show the median (solid), 0.25 and 0.75 quantiles (thick dashed), and 0.05 and 0.95 quantiles (thin dashed) of the resulting log–likelihood sequences. The Rao–Blackwellized sampler has superior typical performance, but occasionally remains trapped in local optima for many iterations.

optima for many iterations (see right columns of Figs. 2.18 and 2.19). These results suggest that while Rao–Blackwellization can usefully accelerate mixing, convergence diagnostics are still important.

2.5 Dirichlet Processes

It is often difficult to find simple parametric models which adequately describe complex, realistic datasets. *Nonparametric* statistical methods avoid assuming restricted functional forms, and thus allow the complexity and accuracy of the inferred model to grow as more data is observed. Strictly speaking, nonparametric models are rarely free of parameters, since they must have a concrete, computationally tractable representation. In Bayesian statistics, nonparametric methods typically learn distributions on function spaces, and thus effectively involve infinitely many parameters [21, 109, 113, 160, 216, 238]. Complexity is controlled via appropriate prior distributions, so that small datasets produce simple predictions, while additional observations induce richer posteriors.

To motivate nonparametric statistical methods, consider De Finetti's representation (see Thm. 2.2.2) of N infinitely exchangeable random variables:

$$p(x_1, x_2, \ldots, x_N) = \int_\Theta p(\theta) \prod_{i=1}^{N} p(x_i \mid \theta)\, d\theta \tag{2.164}$$

In general, this decomposition is only guaranteed when Θ is an infinite–dimensional space of probability measures. Many Bayesian nonparametric methods thus involve families of computationally tractable distributions on probability measures [84]. In particular, the *Dirichlet process* [28, 83, 254] provides a distribution on distributions with many attractive properties, and is widely used in practice [60, 76, 105, 160, 289].

The following sections establish several representations of the Dirichlet process, which characterize its behavior and lead to computationally tractable learning and inference algorithms. We then show that Dirichlet processes provide an elegant alternative to parametric model selection, and discuss extensions to structured, hierarchical models. For other introductions to Dirichlet processes, see [84, 109, 113, 160, 216, 289, 313].

2.5.1 Stochastic Processes on Probability Measures

Because nonparametric methods use stochastic processes to model infinite–dimensional spaces, they are often *implicitly* characterized by the distributions they induce on certain finite statistics. For example, *Gaussian processes* provide a distribution over real–valued functions which is widely used for non–linear regression and classification [1, 109, 229, 253]. By definition, a function $f : \mathcal{X} \to \mathbb{R}$ is distributed according to a Gaussian process if and only if $p(f(x_1), \ldots, f(x_N))$, the density of that function's values at any N points $x_i \in \mathcal{X}$, is jointly Gaussian. This allows Gaussian processes to be tractably parameterized by a mean function and a *covariance kernel* specifying the correlations within any finite point set.

While Gaussian processes define distributions on random functions, a *Dirichlet process* defines a distribution on random probability measures, or equivalently non–negative functions which integrate to one. Let Θ denote a measurable space, as in the parameter space underlying De Finetti's mixture representation (eq. (2.164)). A

Dirichlet process is then parameterized by a *base measure* H on Θ, and a positive scalar *concentration parameter* α. Analogously to the Gaussian case, Dirichlet processes are characterized by the distributions they induce on finite measurable *partitions* (see Fig. 2.21) of the parameter space.

Theorem 2.5.1. *Let H be a probability distribution on a measurable space Θ, and α a positive scalar. Consider a finite partition $(T_1, \ldots, T_K)$ of Θ:*

$$\bigcup_{k=1}^{K} T_k = \Theta \qquad T_k \cap T_\ell = \emptyset \qquad k \neq \ell \tag{2.165}$$

A random probability distribution G on Θ is drawn from a Dirichlet process if its measure on every finite partition follows a Dirichlet distribution:

$$(G(T_1), \ldots, G(T_K)) \sim \mathrm{Dir}(\alpha H(T_1), \ldots, \alpha H(T_K)) \tag{2.166}$$

For any base measure H and concentration parameter α, there exists a unique stochastic process satisfying these conditions, which we denote by $\mathrm{DP}(\alpha, H)$.

Proof. For a characterization as in eq. (2.166) to be valid, probabilities must appropriately add when a partition's cells are combined. The aggregation property of the finite Dirichlet distribution (see eq. (2.43)) is one way to guarantee this. Ferguson originally established the existence of the Dirichlet process via Kolmogorov's consistency conditions [83]. Later, Sethuraman provided a simpler, constructive definition [254] which we describe in Sec. 2.5.2. □

Fig. 2.21 illustrates the consistency requirements relating different partitions of the parameter space Θ. Combining eqs. (2.40) and (2.166), for any region $T \subset \Theta$ the expected measure of a random sample from a Dirichlet process equals

$$\mathbb{E}[G(T)] = H(T) \qquad G \sim \mathrm{DP}(\alpha, H) \tag{2.167}$$

The base measure H thus specifies the mean of $\mathrm{DP}(\alpha, H)$. As we show in Sec. 2.5.3, the concentration parameter α is similar to the precision of a finite Dirichlet distribution, and determines the average deviation of samples from the base measure.

Posterior Measures and Conjugacy

Let $G \sim \mathrm{DP}(\alpha, H)$ be sampled from a Dirichlet process, and $\bar{\theta} \sim G$ be a sample from that distribution. Consider the finite Dirichlet distribution induced by a fixed partition, as in eq. (2.166). Via the conjugacy of the Dirichlet distribution (see eq. (2.45)), the posterior distribution is also Dirichlet:

$$p\big((G(T_1), \ldots, G(T_K)) \mid \bar{\theta} \in T_k\big) = \mathrm{Dir}(\alpha H(T_1), \ldots, \alpha H(T_k) + 1, \ldots, \alpha H(T_K)) \tag{2.168}$$

Note that the observation $\bar{\theta}$ only affects the Dirichlet parameter of the unique, arbitrarily small cell T_k containing it [160]. Formalizing this analysis, it can be shown that the posterior distribution has a Dirac point mass $\delta_{\bar{\theta}}$ centered on each observation.

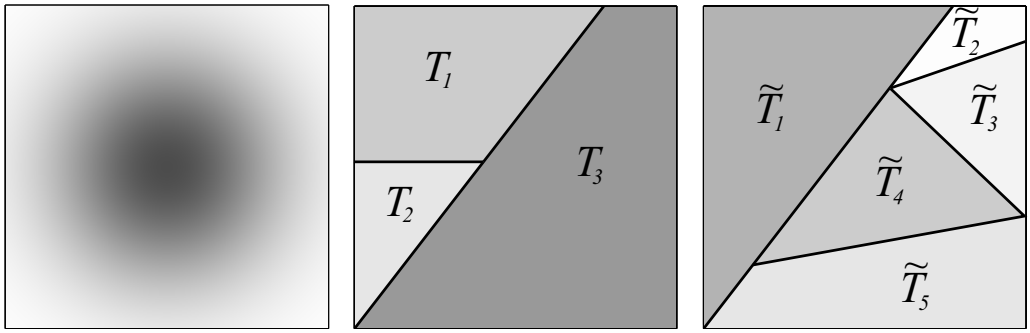


Figure 2.21. Dirichlet processes induce Dirichlet distributions on every finite, measurable partition. *Left:* An example base measure H on a bounded, two–dimensional space Θ (darker regions have higher probability). *Center:* A partition with $K = 3$ cells. The weight that a random measure $G \sim \mathrm{DP}(\alpha, H)$ assigns to these cells follows a Dirichlet distribution (see eq. (2.166)). We shade each cell T_k according to its mean $\mathbb{E}[G(T_k)] = H(T_k)$. *Right:* Another partition with $K = 5$ cells. The consistency of G implies, for example, that $(G(T_1) + G(T_2))$ and $G(\widetilde{T}_1)$ follow identical beta distributions.

Proposition 2.5.1. *Let $G \sim \mathrm{DP}(\alpha, H)$ be a random measure distributed according to a Dirichlet process. Given N independent observations $\bar{\theta}_i \sim G$, the posterior measure also follows a Dirichlet process:*

$$p\big(G \mid \bar{\theta}_1, \ldots, \bar{\theta}_N, \alpha, H\big) = \mathrm{DP}\left(\alpha + N, \frac{1}{\alpha + N}\left(\alpha H + \sum_{i=1}^{N} \delta_{\bar{\theta}_i}\right)\right) \tag{2.169}$$

Proof. As shown by Ferguson [83], this result follows directly from the conjugate form of finite Dirichlet posterior distributions (see eq. (2.45)). See Sethuraman [254] for an alternative proof. □

There are interesting similarities between eq. (2.169) and the general form of conjugate priors for exponential families (see Prop. 2.1.4). The Dirichlet process effectively defines a conjugate prior for distributions on arbitrary measurable spaces. In some contexts, the concentration parameter α can then be seen as expressing confidence in the base measure H via the size of a pseudo–dataset (see [113] for further discussion).

Neutral and Tailfree Processes

The conjugacy of Prop. 2.5.1, which leads to tractable computational methods discussed later, provides one practical motivation for the Dirichlet process. In this section, we show that Dirichlet processes are also characterized by certain conditional independencies. These properties reveal both strengths and weaknesses of the Dirichlet process, and have motivated several other families of stochastic processes.

Let G be a random probability measure on a parameter space Θ. The distribution

of G is *neutral* [69, 84] with respect to a finite partition $(T_1, \ldots, T_K)$ of Θ if and only if

$$G(T_k) \qquad \text{is independent of} \qquad \left\{ \frac{G(T_\ell)}{1 - G(T_k)} \;\middle|\; \ell \neq k \right\} \tag{2.170}$$

given that $G(T_k) < 1$. Thus, for a neutral process, the probability mass assigned to some cell T_k affects the weight of other cells only through the normalization constraint. The relative probabilities assigned to those cells are independent random variables. As shown by the following theorem, the Dirichlet process is characterized by its neutrality with respect to *every* measurable partition.

Theorem 2.5.2. *Consider a distribution $\mathcal{P}$ on probability measures G for some space Θ. Assume that $\mathcal{P}$ assigns positive probability to more than one measure G, and that with probability one samples $G \sim \mathcal{P}$ assign positive measure to at least three distinct points $\theta \in \Theta$. The following conditions are then equivalent:*

(i) $\mathcal{P} = \mathrm{DP}(\alpha, H)$ is a Dirichlet process for some base measure H on Θ.

(ii) $\mathcal{P}$ is neutral with respect to every finite, measurable partition of Θ.

(iii) For every measurable $T \subset \Theta$, and any N observations $\bar{\theta}_i \sim G$, the posterior distribution $p\big(G(T) \mid \bar{\theta}_1, \ldots, \bar{\theta}_N\big)$ depends only on the number of observations that fall within T (and not their particular locations).

Proof. This result was derived by Doksum and Fabius via related characterizations of the finite Dirichlet distribution. See [69, 84] for a more precise description of degenerate cases, and additional references. □

This theorem shows that Dirichlet processes effectively ignore the topology of the parameter space Θ. Observations provide information only about those cells which directly contain them. In addition, an observation near the boundary of a cell provides the same amount of information as an observation in its center. Thus, while neutrality simplifies the structure of posterior distributions, it also limits the expressiveness of the corresponding prior.

For problems in which $\Theta = \mathbb{R}$ is the real line, a less restrictive form of neutrality has been proposed. A random cumulative distribution $F(t) = \Pr[\theta \leq t]$ is *neutral to the right (NTR)* [69, 84] if, for any K times $t_1 < \cdots < t_K$, the normalized increments

$$\left\{ F(t_1), \frac{F(t_2) - F(t_1)}{1 - F(t_1)}, \ldots, \frac{F(t_K) - F(t_{K-1})}{1 - F(t_{K-1})} \right\} \tag{2.171}$$

are mutually independent. This condition is strictly weaker than that of eq. (2.170), and several NTR generalizations of the Dirichlet process have been suggested [69, 313]. Any NTR stochastic process can be expressed as $F(t) = 1 - \exp\{-Y(t)\}$ for some monotonically increasing, independent increments process $Y(t)$. For the Dirichlet process,

increments of $Y(t)$ are exponentially distributed [84, 150]. In addition, NTR processes are *tailfree*, so that the posterior distribution $p\big(F(t) \mid \bar{\theta}\big)$ is independent of observations at later times $\bar{\theta} > t$. Generalizing the conjugacy of Prop. 2.5.1, the posterior distribution of $F(t)$ given an observation $\bar{\theta} \leq t$ remains neutral to the right [69].

While NTR processes can more flexibly model temporal structure than the Dirichlet process, they are limited to the real line. A recently proposed class of *spatial neutral to the right* processes [152] provides one extension to general parameter spaces. Alternatively, tailfree processes can be generalized to define conditional independencies on arbitrary sequences of nested partitions [69, 84]. Analogously to Thm. 2.5.2, only Dirichlet processes are tailfree with respect to every hierarchical partition. However, a broader class of *Pólya tree* distributions [84, 179, 200] can be defined via particular, possibly inhomogeneous partition trees. While this tree structure can encode detailed prior knowledge [180], its use of a fixed discretization scales poorly to high–dimensional spaces, and can produce spurious discontinuities. *Dirichlet diffusion trees* [223] address these issues by using a branching process to sample hierarchical dependency structures.

■ 2.5.2 Stick–Breaking Processes

The preceding section provides several implicit characterizations of the Dirichlet process, including a desirable conjugacy property. However, these results do not directly provide a mechanism for sampling from Dirichlet processes, or predicting future observations. In this section, we describe an explicit *stick–breaking* construction [254] which shows that Dirichlet measures are *discrete* with probability one. This leads to a simple Pólya urn model for predictive distributions known as the *Chinese restaurant process* [28, 233]. These representations play a central role in computational methods for Dirichlet processes.

Consider Prop. 2.5.1, which provides an expression for the posterior distribution of a Dirichlet distributed random measure $G \sim \mathrm{DP}(\alpha, H)$ given N observations $\bar{\theta}_i \sim G$. From eq. (2.167), the expected measure of any set $T \subset \Theta$ then equals

$$\mathbb{E}\big[G(T) \mid \bar{\theta}_1, \ldots, \bar{\theta}_N, \alpha, H\big] = \frac{1}{\alpha + N} \left(\alpha H(T) + \sum_{i=1}^{N} \delta_{\bar{\theta}_i}(T) \right) \tag{2.172}$$

For any finite concentration parameter α, this implies that

$$\lim_{N \to \infty} \mathbb{E}\big[G(T) \mid \bar{\theta}_1, \ldots, \bar{\theta}_N, \alpha, H\big] = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}(T) \tag{2.173}$$

where $\{\theta_k\}_{k=1}^{\infty}$ are the unique values of the observation sequence $\{\bar{\theta}_i\}_{i=1}^{\infty}$, and π_k is the limiting empirical frequency of θ_k. Assuming the posterior distribution concentrates about its mean, eq. (2.173) suggests that Dirichlet measures are discrete with probability one [160]. The following theorem verifies this hypothesis, and provides an explicit construction for the infinite set of mixture weights.

Theorem 2.5.3. *Let $\pi = \{\pi_k\}_{k=1}^{\infty}$ be an infinite sequence of mixture weights derived from the following stick–breaking process, with parameter $\alpha > 0$:*

$$\beta_k \sim \text{Beta}(1, \alpha) \qquad k = 1, 2, \ldots \tag{2.174}$$

$$\pi_k = \beta_k \prod_{\ell=1}^{k-1}(1 - \beta_\ell) = \beta_k\left(1 - \sum_{\ell=1}^{k-1}\pi_\ell\right) \tag{2.175}$$

Given a base measure H on Θ, consider the following discrete random measure:

$$G(\theta) = \sum_{k=1}^{\infty}\pi_k\delta(\theta, \theta_k) \qquad \theta_k \sim H \tag{2.176}$$

This construction guarantees that $G \sim \text{DP}(\alpha, H)$. Conversely, samples from a Dirichlet process are discrete with probability one, and have a representation as in eq. (2.176).

Proof. The consistency of eq. (2.175) follows from an induction argument. Manipulating this expression, it can be shown that

$$1 - \sum_{k=1}^{K}\pi_k = \prod_{k=1}^{K}(1 - \beta_k) \longrightarrow 0$$

with probability one as $K \to \infty$, so that eq. (2.176) defines a valid probability measure. Ferguson established the almost sure discreteness of G using a normalized gamma process representation [83, 168]. Sethuraman later derived the explicit stick–breaking construction for the mixture weights [254]. The beta distribution of eq. (2.174) arises from the form of marginal distributions of finite Dirichlet densities (see eq. (2.44)). □

The *stick–breaking* interpretation of this construction is illustrated in Fig. 2.22. Mixture weights π partition a unit–length "stick" of probability mass among an infinite set of random parameters. The k^{th} mass π_k is a random proportion β_k of the stick remaining after sampling the first $(k-1)$ mixture weights. As is standard in the statistics literature [150, 233, 289], we use $\pi \sim \text{GEM}(\alpha)$ to indicate a set of mixture weights sampled from this process, named after Griffiths, Engen, and McCloskey.

This representation of the Dirichlet process provides another interpretation of the concentration parameter α. Because the stick–breaking proportions $\beta_k \sim \text{Beta}(1, \alpha)$, standard moment formulas (see eq. (2.40)) show that

$$\mathbb{E}[\beta_k] = \frac{1}{1+\alpha} \tag{2.177}$$

For small α, it follows that the first few mixture components are typically assigned the majority of the probability mass. As $\alpha \to \infty$, samples $G \sim \text{DP}(\alpha, H)$ approach the base measure H by assigning small, roughly uniform weights to a densely sampled set of

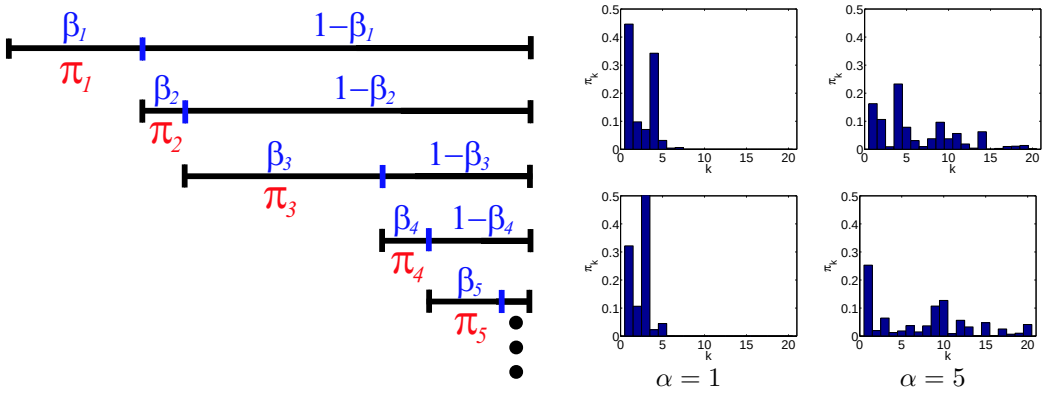


Figure 2.22. Sequential stick–breaking construction of the infinite set of mixture weights $\pi \sim \mathrm{GEM}(\alpha)$ corresponding to a measure $G \sim \mathrm{DP}(\alpha, H)$. *Left:* The first weight $\pi_1 \sim \mathrm{Beta}(1, \alpha)$. Each subsequent weight π_k (red) is some random proportion β_k (blue) of the remaining, unbroken "stick" of probability mass. *Right:* The first $K = 20$ weights generated by four random stick–breaking constructions (two with $\alpha = 1$, two with $\alpha = 5$). Note that the weights π_k do not monotonically decrease.

discrete parameters $\{\theta_k\}_{k=1}^{\infty}$. For a given α and dataset size N, there are strong bounds on the accuracy of particular finite truncations of this stick–breaking process [147], which are often used in approximate computational methods [29, 147, 148, 289].

Several other stick–breaking processes have been proposed which sample the proportions β_k from different distributions [147, 148, 233]. For example, the two–parameter Poisson–Dirichlet, or Pitman–Yor, process [234] can produce heavier–tailed weight distributions which better match power laws arising in natural language processing [117, 287]. As we show next, these stick–breaking processes sometimes lead to predictive distributions with simple Pólya urn representations.

Prediction via Pólya Urns

Because Dirichlet processes produce discrete random measures G, there is a strictly positive probability of multiple observations $\bar{\theta}_i \sim G$ taking identical values. Given N observations $\{\bar{\theta}_i\}_{i=1}^{N}$, suppose that they take $K \leq N$ distinct values $\{\theta_k\}_{k=1}^{K}$. The posterior expectation of any set $T \subset \Theta$ (see eq. (2.172)) can then be written as

$$\mathbb{E}\big[G(T) \mid \bar{\theta}_1, \ldots, \bar{\theta}_N, \alpha, H\big] = \frac{1}{\alpha + N}\left(\alpha H(T) + \sum_{k=1}^{K} N_k \delta_{\theta_k}(T)\right) \tag{2.178}$$

$$N_k \triangleq \sum_{i=1}^{N} \delta(\bar{\theta}_i, \theta_k) \qquad k = 1, \ldots, K \tag{2.179}$$

Note that N_k is defined to be the number of previous observations equaling θ_k, and that K is a random variable [10, 28, 233]. Analyzing this expression, the predictive distribution of the next observation $\bar{\theta}_{N+1} \sim G$ can be explicitly characterized.

Theorem 2.5.4. *Let $G \sim \mathrm{DP}(\alpha, H)$ be distributed according to a Dirichlet process, where the base measure H has corresponding density $h(\theta)$. Consider a set of N observations $\bar{\theta}_i \sim G$ taking K distinct values $\{\theta_k\}_{k=1}^K$. The predictive distribution of the next observation then equals*

$$p\big(\bar{\theta}_{N+1} = \theta \mid \bar{\theta}_1, \dots, \bar{\theta}_N, \alpha, H\big) = \frac{1}{\alpha + N}\left(\alpha h(\theta) + \sum_{k=1}^{K} N_k \delta(\theta, \theta_k)\right) \tag{2.180}$$

where N_k is the number of previous observations of θ_k, as in eq. (2.179).

Proof. Letting T_k be an arbitrarily small set containing θ_k, eq. (2.178) suggests that $\Pr\big[\bar{\theta}_{N+1} = \theta_k\big] \propto N_k$, while the base measure is assigned total posterior probability $\alpha/(\alpha + N)$. For a formal argument, see Blackwell and MacQueen [28]. □

Dirichlet processes thus lead to simple predictive distributions, which can be evaluated by caching the number of previous observations taking each distinct value.

The generative process defined by Thm. 2.5.4 can be interpreted via a generalized *Pólya urn* model [28]. Consider an urn containing one ball for each preceding observation, with a different color for each distinct θ_k. For each ball drawn from the urn, we replace that ball and add one more of the same color. There is also a special "weighted" ball which is drawn with probability proportional to α normal balls, and has some new, previously unseen color $\theta_{\bar{k}} \sim H$. This procedure can be used to sample observations from a Dirichlet process, without explicitly constructing the underlying mixture $G \sim \mathrm{DP}(\alpha, H)$.

Chinese Restaurant Processes

As the Dirichlet process assigns observations $\bar{\theta}_i$ to distinct values θ_k, it implicitly *partitions* the data. Let z_i indicate the subset, or cluster, associated with the i^{th} observation, so that $\bar{\theta}_i = \theta_{z_i}$. The predictive distribution of eq. (2.180) then shows that

$$p(z_{N+1} = z \mid z_1, \dots, z_N, \alpha) = \frac{1}{\alpha + N}\left(\sum_{k=1}^{K} N_k \delta(z, k) + \alpha \delta(z, \bar{k})\right) \tag{2.181}$$

where $\bar{k}$ denotes a new, previously empty cluster. Inspired by the seemingly infinite seating capacity of restaurants in San Francisco's Chinatown, Pitman and Dubins called this distribution over partitions the *Chinese restaurant process* [233]. The restaurant's infinite set of tables are analogous to clusters, and customers to observations (see Fig. 2.23). Customers are social, so that the i^{th} customer sits at table k with probability proportional to the number of already seated diners N_k. Sometimes, however, customers (observations) choose a new table (cluster). Note that there is no *a priori* distinction between the unoccupied tables. Dirichlet processes extend this construction by serving each table a different, independently chosen dish (parameter) θ_k.

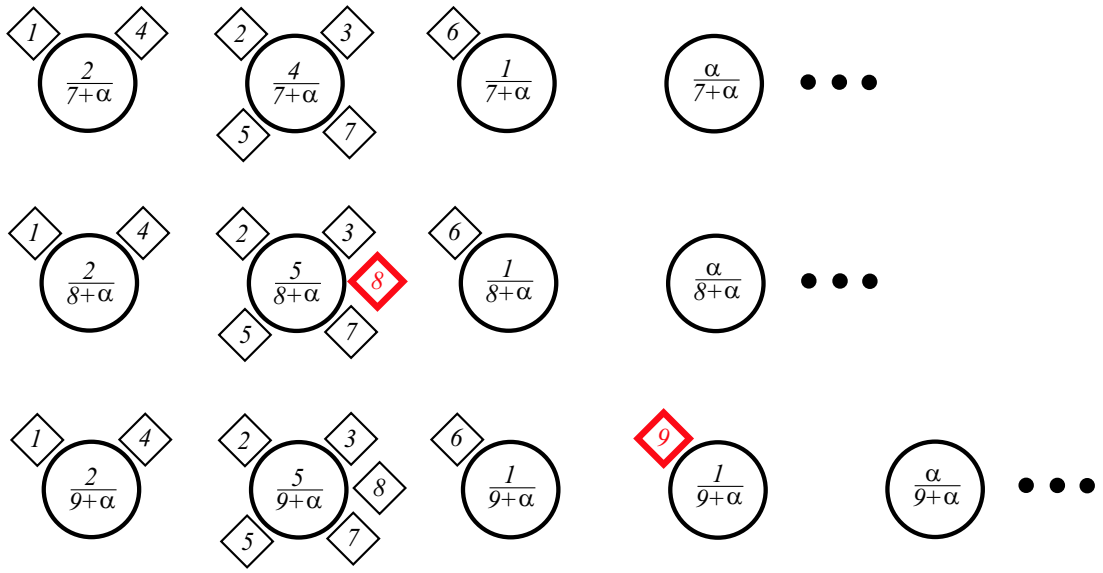


Figure 2.23. Chinese restaurant process interpretation of the partitions induced by the Dirichlet process $\mathrm{DP}(\alpha, H)$. Tables (circles) are analogous to clusters, and customers (diamonds) to a series of observations. *Top row:* A starting configuration, in which seven customers occupy three tables. Each table is labeled with the probability that the next customer sits there. *Middle row:* New customers sit at occupied table k with probability proportional to the number of previously seated diners N_k. In this example, the eighth customer joins the most popular, and hence likely, table. *Bottom row:* Customers may also sit at one of the infinitely many unoccupied tables. The ninth diner does this.

Importantly, the Chinese restaurant process induces an *exchangeable* distribution on partitions, so that the joint distribution is invariant to the order in which observations are assigned to clusters. Exchangeability follows from De Finetti's Theorem [28], given the connection to Dirichlet processes established by Thm. 2.5.4. Alternatively, it can be directly verified via an analysis of eq. (2.181). There are a variety of combinatorial characterizations of the partition structure produced by the Chinese restaurant process [10, 121, 232, 233]. In particular, the number of occupied tables K almost surely approaches $\alpha \log(N)$ as $N \to \infty$. This shows that the Dirichlet process is indeed a nonparametric prior, as it favors models whose complexity grows with the dataset size.

Generalizations of the Chinese restaurant process can be constructed for certain other stick–breaking processes, including the Pitman–Yor process [147, 233]. Importantly, the simple predictive distributions induced by these processes lead to efficient Monte Carlo algorithms for learning and inference [76, 222, 237]. In contrast, other alternatives such as neutral to the right processes may have posterior distributions which lack simple, explicit forms [152].

■ 2.5.3 Dirichlet Process Mixtures

Using nonparametric methods, we now revisit De Finetti's representation (Thm. 2.2.2) of exchangeable random variables $\{x_i\}_{i=1}^N$. To apply this theory when $x_i \in \mathcal{X}$ is continuous, we need a tractable framework for learning infinite–dimensional probability measures. As shown in previous sections, Dirichlet processes lead to posterior distributions with simple, explicit forms. However, because it assigns probability one to discrete measures (Thm. 2.5.3), a Dirichlet process prior expects multiple observations to take *identical* values. Furthermore, Thm. 2.5.2 shows that the posterior measure assigned to x_i would *never* be influenced by observations $x_j \neq x_i$, regardless of their proximity. In many applications, Dirichlet processes are thus too restrictive to directly model continuous observations [216, 232].

To address these issues, we consider a hierarchical model in which observations are sampled from some parameterized family $F(\theta)$. As in finite mixture models (see Fig. 2.9), each observation x_i is based on an independently sampled parameter $\bar{\theta}_i$:

$$\begin{aligned} \bar{\theta}_i &\sim G \\ x_i &\sim F\big(\bar{\theta}_i\big) \end{aligned} \tag{2.182}$$

For greater flexibility and robustness, we place a nonparametric, Dirichlet process prior on the latent parameter distribution $G \sim \mathrm{DP}(\alpha, H)$. The stick–breaking construction of Thm. 2.5.3 then implies that

$$G(\theta) = \sum_{k=1}^{\infty} \pi_k \delta(\theta, \theta_k) \qquad \begin{aligned} \pi &\sim \mathrm{GEM}(\alpha) \\ \theta_k &\sim H(\lambda) \qquad k = 1, 2, \ldots \end{aligned} \tag{2.183}$$

Fig. 2.24 shows a graphical representation of the resulting *Dirichlet process mixture* model [10, 76, 187]. Typically, $F(\theta)$ is some exponential family of densities, and $H(\lambda)$ a corresponding conjugate prior. Note that this construction allows differing observations to be associated with the same underlying cluster. The likelihood $F(\theta)$ effectively imposes a notion of distance on $\mathcal{X}$, and thus allows observations to be extrapolated to neighboring regions. By using a Dirichlet process, however, we avoid constraining these predictions with a global parametric form. Fig. 2.25 illustrates Dirichlet process mixtures in which $\theta_k = (\mu_k, \Lambda_k)$ parameterizes a two–dimensional Gaussian.

The Chinese restaurant process provides another useful representation of Dirichlet process mixtures [76, 237]. Letting z_i denote the unique cluster, or table, associated with x_i, the generative process of eq. (2.182) can be equivalently expressed as

$$\begin{aligned} z_i &\sim \pi \\ x_i &\sim F(\theta_{z_i}) \end{aligned} \tag{2.184}$$

As summarized in Fig. 2.24, marginalizing these indicator variables reveals an *infinite* mixture model with the following form:

$$p(x \mid \pi, \theta_1, \theta_2, \ldots) = \sum_{k=1}^{\infty} \pi_k f(x \mid \theta_k) \tag{2.185}$$

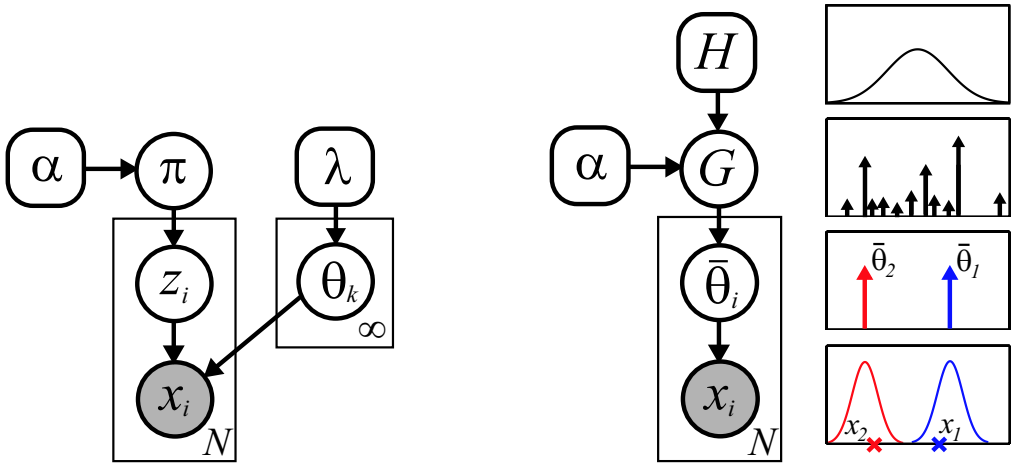


Figure 2.24. Directed graphical representations of an infinite, Dirichlet process mixture model. Mixture weights $\pi \sim \text{GEM}(\alpha)$ follow a stick–breaking process, while cluster parameters are assigned independent priors $\theta_k \sim H(\lambda)$. *Left:* Indicator variable representation, in which $z_i \sim \pi$ is the cluster that generates $x_i \sim F(\theta_{z_i})$. *Right:* Alternative distributional form, in which G is an infinite discrete distribution on Θ. $\bar{\theta}_i \sim G$ are the parameters of the cluster that generates $x_i \sim F(\bar{\theta}_i)$. We illustrate with an infinite Gaussian mixture, where cluster variances are known (bottom) and $H(\lambda)$ is a Gaussian prior on cluster means (top). Sampled cluster means $\bar{\theta}_1, \bar{\theta}_2$, and corresponding Gaussians, are shown for two observations x_1, x_2.

Rather than choose a finite model order K, Dirichlet process mixtures use the stick–breaking prior to control complexity (see Fig. 2.22). As we discuss later, this relaxation leads to algorithms which automatically infer the number of clusters exhibited by a particular dataset. Importantly, the predictive distribution implied by the Chinese restaurant process (eq. (2.181)) has a *clustering* bias, and favors simpler models in which observations (customers) share parameters (dishes). Additional clusters (tables) appear as more observations are generated (see Fig. 2.25).

Learning via Gibbs Sampling

Given N observations $x = \{x_i\}_{i=1}^N$ from a Dirichlet process mixture as in Fig. 2.24, we would like to infer the number of latent clusters underlying those observations, and their parameters θ_k. As with finite mixture models, the exact posterior distribution $p(\pi, \theta \mid x)$ contains terms corresponding to each possible partition z of the observations [10, 187]. While the Chinese restaurant process tractably specifies the prior probability of individual partitions (see eq. (2.181)), explicit enumeration of the exponentially large set of potential partitions is intractable. There is thus an extensive literature on approximate computational methods for Dirichlet process mixtures [29, 76, 121, 147, 148, 151, 222].

In this section, we generalize the Rao–Blackwellized Gibbs sampler of Alg. 2.2 from finite to infinite mixture models. As before, we sample the indicator variables $z = \{z_i\}_{i=1}^N$ assigning observations to latent clusters, marginalizing mixture weights π

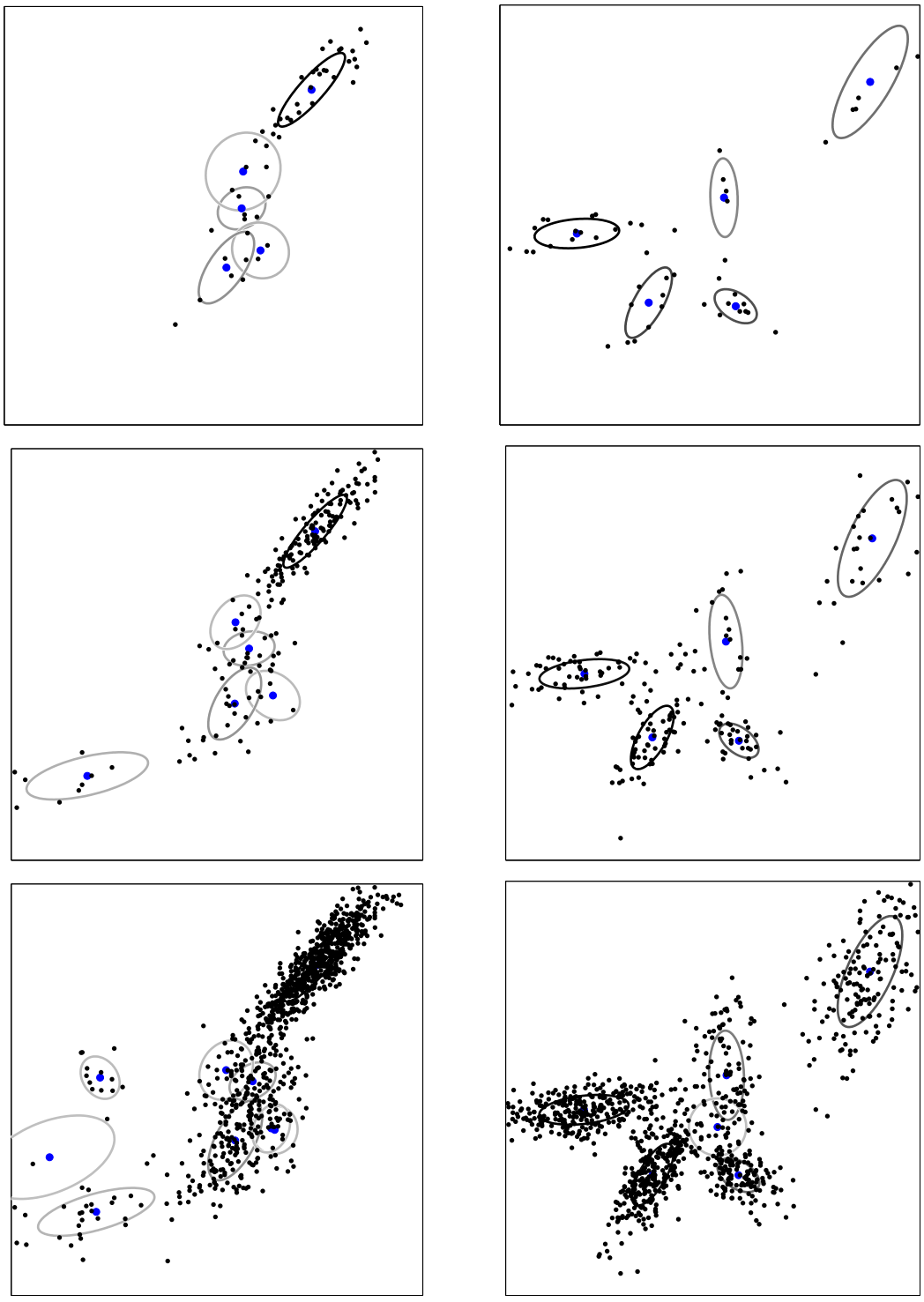


Figure 2.25. Each column shows an observation sequence from a Dirichlet process mixture of 2D Gaussians, with concentration $\alpha = 1$. We show the existing clusters (covariance ellipses, intensity proportional to probability) after $N = 50$ (top), $N = 200$ (middle), and $N = 1000$ (bottom) observations.

and parameters $\{\theta_k\}_{k=1}^{\infty}$. The resulting *collapsed* Gibbs sampler [222] is typically more efficient than alternatives which explicitly sample parameters [76, 237]. For simplicity, we assume that cluster priors $H(\lambda)$ are conjugate to the chosen likelihood $F(\theta)$. Non–conjugate priors can be handled via auxiliary variable methods [222].

Given fixed cluster assignments $z_{\backslash i}$ for other observations, Fig. 2.24 implies that the posterior distribution of z_i factors as follows:

$$p(z_i \mid z_{\backslash i}, x, \alpha, \lambda) \propto p(z_i \mid z_{\backslash i}, \alpha)\, p(x_i \mid z, x_{\backslash i}, \lambda) \tag{2.186}$$

The first term expresses the prior on partitions implied by the Chinese restaurant process. Recall that the Dirichlet process induces an exchangeable distribution on partitions, which is invariant to the order of observations. In evaluating eq. (2.186), we may thus equivalently think of z_i as the *last* in a sequence of N observations. If $z_{\backslash i}$ instantiates K clusters, and assigns N_k^{-i} observations to the k^{th} cluster, eq. (2.181) then implies that

$$p(z_i \mid z_{\backslash i}, \alpha) = \frac{1}{\alpha + N - 1} \left(\sum_{k=1}^{K} N_k^{-i} \delta(z_i, k) + \alpha \delta(z_i, \bar{k}) \right) \tag{2.187}$$

As before, $\bar{k}$ denotes one of the infinitely many unoccupied clusters.

For the K clusters to which $z_{\backslash i}$ assigns observations, the likelihood of eq. (2.186) follows the expression (eq. (2.163)) derived for the finite mixture Gibbs sampler:

$$p(x_i \mid z_i = k, z_{\backslash i}, x_{\backslash i}, \lambda) = p(x_i \mid \{x_j \mid z_j = k, j \neq i\}, \lambda) \tag{2.188}$$

This term is the predictive likelihood of x_i, as determined by Prop. 2.1.4, given the other observations which $z_{\backslash i}$ associates with that cluster. Similarly, new clusters $\bar{k}$ are based upon the predictive likelihood implied by the prior hyperparameters λ:

$$p\big(x_i \mid z_i = \bar{k}, z_{\backslash i}, x_{\backslash i}, \lambda\big) = p(x_i \mid \lambda) = \int_{\Theta} f(x_i \mid \theta)\, h(\theta \mid \lambda)\, d\theta \tag{2.189}$$

Assuming $H(\lambda)$ specifies a proper, conjugate prior, eq. (2.189) has a closed form similar to that of eq. (2.188).

Combining these expressions, we arrive at the Gibbs sampler of Alg. 2.3. As in Alg. 2.2, we cache and recursively update statistics of each cluster's associated observations (see Thm. 2.1.2). Because the infinite set of potential clusters have identical priors, we only explicitly store a randomly sized list of those clusters to which at least one observation is assigned. Standard data structures then allow clusters to be efficiently created when needed (Alg. 2.3, step 2(c)), and deleted if all associated observations are reassigned (Alg. 2.3, step 4). Comparing Algs. 2.2 and 2.3, we see that even though Dirichlet process mixtures have infinitely many parameters, learning is possible via a simple extension of algorithms developed for finite mixture models.

Cluster assignments $z^{(t)}$ produced by the Gibbs sampler of Alg. 2.3 provide estimates $K^{(t)}$ of the *number* of clusters underlying the observations x, as well as their

Given the previous concentration parameter $\alpha^{(t-1)}$, cluster assignments $z^{(t-1)}$, and cached statistics for the K current clusters, sequentially sample new assignments as follows:

1. Sample a random permutation $\tau(\cdot)$ of the integers $\{1, \dots, N\}$.
2. Set $\alpha = \alpha^{(t-1)}$ and $z = z^{(t-1)}$. For each $i \in \{\tau(1), \dots, \tau(N)\}$, resample z_i as follows:
   (a) For each of the K existing clusters, determine the predictive likelihood
   $$f_k(x_i) = p(x_i \mid \{x_j \mid z_j = k, j \neq i\}, \lambda)$$
   This likelihood can be computed from cached sufficient statistics via Prop. 2.1.4. Also determine the likelihood $f_{\bar{k}}(x_i)$ of a potential new cluster $\bar{k}$ via eq. (2.189).
   (b) Sample a new cluster assignment z_i from the following $(K+1)$–dim. multinomial:
   $$z_i \sim \frac{1}{Z_i}\left(\alpha f_{\bar{k}}(x_i)\delta(z_i, \bar{k}) + \sum_{k=1}^{K} N_k^{-i} f_k(x_i)\delta(z_i, k)\right) \qquad Z_i = \alpha f_{\bar{k}}(x_i) + \sum_{k=1}^{K} N_k^{-i} f_k(x_i)$$
   N_k^{-i} is the number of other observations currently assigned to cluster k.
   (c) Update cached sufficient statistics to reflect the assignment of x_i to cluster z_i. If $z_i = \bar{k}$, create a new cluster and increment K.
3. Set $z^{(t)} = z$. Optionally, mixture parameters for the K currently instantiated clusters may be sampled as in step 3 of Alg. 2.1.
4. If any current clusters are empty ($N_k = 0$), remove them and decrement K accordingly.
5. If $\alpha \sim \text{Gamma}(a, b)$, sample $\alpha^{(t)} \sim p(\alpha \mid K, N, a, b)$ via auxiliary variable methods [76].

Algorithm 2.3. Rao–Blackwellized Gibbs sampler for an infinite, Dirichlet process mixture model, as defined in Fig. 2.24. Each iteration sequentially resamples the cluster assignments for all N observations $x = \{x_i\}_{i=1}^N$ in a different random order. Mixture parameters are integrated out of the sampling recursion using cached sufficient statistics. These statistics are stored in a dynamically resized list of those clusters to which observations are currently assigned.

associated parameters. Dirichlet processes thus effectively allow integrated exploration of models with different complexity. Predictions based on these samples average over mixtures of varying size, avoiding the difficulties inherent in selecting a single model. The computational cost of each sampling update is proportional to the number of currently instantiated clusters $K^{(t)}$, and thus varies randomly from iteration to iteration. Asymptotically, $K \to \alpha \log(N)$ as $N \to \infty$ (see [10, 233]), so each iteration of Alg. 2.3 requires approximately $\mathcal{O}(\alpha N \log(N))$ operations to resample all assignments. For practical datasets, however, the number of instantiated clusters depends substantially on the structure and alignment of the given observations.

While predictions derived from Dirichlet process mixtures are typically robust to the concentration parameter α, the number K of clusters with significant posterior probability shows greater sensitivity [76]. In many applications, it is therefore useful to choose a weakly informative prior for α, and sample from its posterior while learning cluster parameters. If $\alpha \sim \text{Gamma}(a, b)$ is assigned a gamma prior [107], its posterior is a simple function of K, and samples are easily drawn via an auxiliary variable method [76]. Incorporating this technique in our Gibbs sampler (Alg. 2.3, step 5), we empirically

find that it converges more reliably, and matches the performance of procedures which tune α via computationally demanding cross–validation.

In Fig. 2.26, we use the Gibbs sampler of Alg. 2.3 to fit a Dirichlet process mixture of Gaussians to $N = 300$ two–dimensional observations. Placing a vague gamma prior $\alpha \sim \text{Gamma}(0.2, 0.1)$ on the concentration parameter, initial iterations frequently create and delete mixture components. However, the sampler quickly stabilizes (see Fig. 2.27), and discovers that with high probability the data was generated by $K = 4$ Gaussians. Fig. 2.27 also compares this Dirichlet process model to a 4–component mixture estimated via the Rao–Blackwellized sampler of Alg. 2.2. Despite having to search over mixtures of varying order, the Dirichlet process sampler typically converges faster. In particular, by creating redundant clusters in early iterations, it avoids local optima which trap the 4–component Gibbs sampler. This behavior is reminiscent of methods which iteratively prune clusters from finite mixtures [87], but arises directly from the Dirichlet process prior rather than complexity–based model selection criteria.

An Infinite Limit of Finite Mixtures

The graphical representation of the Dirichlet process mixture model (see Fig. 2.24) exhibits striking similarities to the finite, K–component mixture model of Fig. 2.9. In this section, we show that the Dirichlet process is indeed the limit as $K \to \infty$ of a particular sequence of finite Dirichlet distributions. This result provides intuition about the assumptions and biases inherent in Dirichlet processes, and leads to alternative computational methods for Dirichlet process mixtures.

As in Sec. 2.2.4, we begin by placing a symmetric Dirichlet prior, with precision α, on the weights π assigned to the K components of a finite mixture model:

$$(\pi_1, \ldots, \pi_K) \sim \text{Dir}\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right) \tag{2.190}$$

Consider the Rao–Blackwellized Gibbs sampler for this finite mixture, as summarized in Alg. 2.2. Given cluster assignments $z_{\setminus i}$ for all observations except x_i, the Dirichlet prior implies the following predictive distribution (see eq. (2.162)):

$$p(z_i = k \mid z_{\setminus i}, \alpha) = \frac{N_k^{-i} + \alpha/K}{\alpha + N - 1} \qquad N_k^{-i} = \sum_{j \neq i} \delta(z_j, k) \tag{2.191}$$

In the limit as $K \to \infty$ with fixed precision α, the predictive probability of clusters k to which observations are assigned ($N_k^{-i} > 0$) approaches

$$\lim_{K \to \infty} p(z_i = k \mid z_{\setminus i}, \alpha) = \frac{N_k^{-i}}{\alpha + N - 1} \tag{2.192}$$

Similarly, the probability of any particular unoccupied cluster approaches zero as K becomes large. However, the total probability assigned to *all* unoccupied clusters is

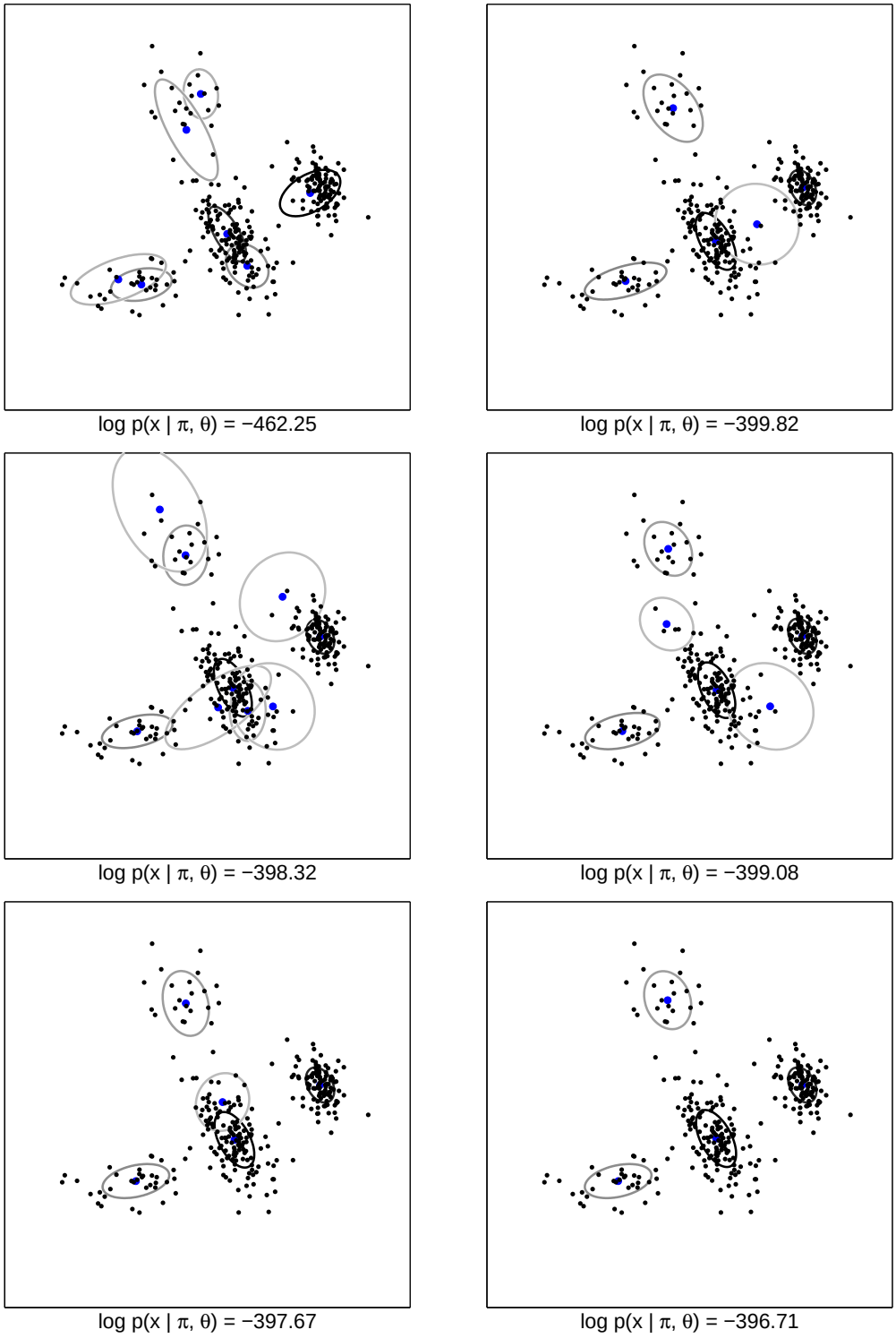


Figure 2.26. Learning a mixture of Gaussians using the Dirichlet process Gibbs sampler of Alg. 2.3. Columns show the parameters of clusters currently assigned to observations, and corresponding data log–likelihoods, after T=2 (top), T=10 (middle), and T=50 (bottom) iterations from two initializations.

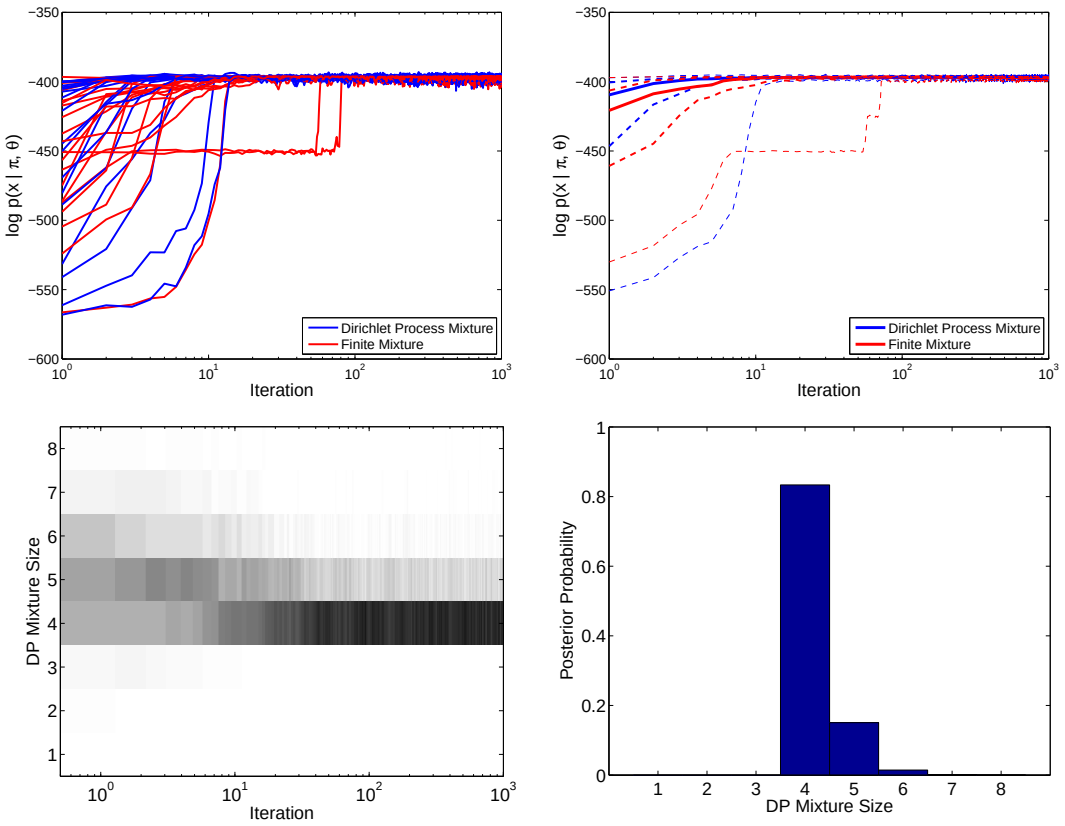


Figure 2.27. Comparison of Rao–Blackwellized Gibbs samplers for a Dirichlet process mixture (Alg. 2.3, dark blue) and a finite, 4–component mixture (Alg. 2.2, light red). We compare data log–likelihoods at each of 1000 iterations for the single $N = 300$ point dataset of Fig. 2.26. *Top left:* Log–likelihood sequences for 20 different random initializations of each algorithm. *Top Right:* From 100 different random initializations, we show the median (solid), 0.25 and 0.75 quantiles (thick dashed), and 0.05 and 0.95 quantiles (thin dashed) of the resulting log–likelihood sequences. *Bottom:* Number of mixture components with at least 2% of the probability mass at each iteration (left, intensity proportional to posterior probability), and averaging across the final 900 iterations (right).

positive, and determined by the complement of existing cluster weights as follows:

$$p(z_i \neq z_j \text{ for all } j \neq i \mid z_{\setminus i}, \alpha) = 1 - \sum_{k | N_k^{-i} > 0} p(z_i = k \mid z_{\setminus i}, \alpha) \tag{2.193}$$

$$\lim_{K \to \infty} p(z_i \neq z_j \text{ for all } j \neq i \mid z_{\setminus i}, \alpha) = 1 - \sum_{k} \frac{N_k^{-i}}{\alpha + N - 1} = \frac{\alpha}{\alpha + N - 1} \tag{2.194}$$

Note that if z_i is not assigned to an occupied cluster, it must be associated with a new cluster $\bar{k}$. Comparing to eq. (2.187), we then see that the limits of eqs. (2.192, 2.194) are equivalent to the predictive distributions implied by the Chinese restaurant process. The Dirichlet process Gibbs sampler of Alg. 2.3 can thus be directly derived as an

infinite limit of Alg. 2.2, without explicitly invoking the theory underlying Dirichlet processes [221, 237].

The relationships suggested by the preceding arguments can be made more precise. In particular, a combinatorial analysis [121, 150] shows that the finite Dirichlet prior of eq. (2.190) induces a joint distribution on partitions z which approaches the Chinese restaurant process as $K \to \infty$. In this limit, predictions based on the finite mixture model also approach those of the corresponding Dirichlet process.

Theorem 2.5.5. *Let H denote a probability measure on Θ, and $f : \Theta \to \mathbb{R}$ a measurable function which is integrable with respect to H. Consider the K–component discrete distribution G^K, as in eq. (2.83), corresponding to a mixture model with weights following the finite Dirichlet prior $\mathrm{Dir}(\alpha)$ of eq. (2.190). As $K \to \infty$, expectations with respect to G^K then converge in distribution to a corresponding Dirichlet process:*

$$\int_\Theta f(\theta)\, dG^K(\theta) \xrightarrow{\mathcal{D}} \int_\Theta f(\theta)\, dG(\theta) \qquad G \sim \mathrm{DP}(\alpha, H) \tag{2.195}$$

Proof. This result was derived via a stick–breaking representation of the Dirichlet process by Ishwaran and Zarepour (see Thm. 2 of [150]). □

Given the correspondence implied by Thm. 2.5.5, the mixture weights $(\pi_1, \ldots, \pi_K)$ of eq. (2.190) should, in some sense, converge to $\pi \sim \mathrm{GEM}(\alpha)$ as $K \to \infty$. As discussed in Sec. 2.1.3, finite Dirichlet distributions with small precisions are biased towards sparse multinomial distributions (see Fig. 2.1). It can be shown that the stick–breaking construction of Thm. 2.5.3 induces a random, *size–biased permutation* [233] in which the largest weights are typically assigned to earlier clusters (for examples, see Fig. 2.22). By rank ordering $\pi \sim \mathrm{GEM}(\alpha)$, we recover the *Poisson–Dirichlet* distribution [233, 234], which is also the limiting distribution of reordered, finite Dirichlet samples [168].

Given the limiting behavior of finite mixture models with Dirichlet priors as in eq. (2.190), they provide a natural mechanism for approximating Dirichlet processes. Indeed, a Gibbs sampler similar to those of Algs. 2.1 and 2.2 has been suggested for approximate learning of Dirichlet process mixtures [148]. In general, however, this finite mixture approximation converges slowly with K, and a large number of potential clusters may be required [148, 150]. More accurate approximations, whose error decreases exponentially with K, are obtained by truncating the stick–breaking representation of Thm. 2.5.3. This approach has been used to develop alternative Gibbs samplers [147, 148], as well as a deterministic, variational approximation [29] which adapts the mean field method described in Sec. 2.3.1.

Model Selection and Consistency

Dirichlet process mixture models provide a popular Bayesian alternative to the kernel density estimators described in Sec. 2.4.2. In such applications, clusters are usually associated with Gaussian kernels [76, 187]. The base measure $H(\lambda)$ may then be used to

encode domain–specific knowledge about the observations' expected location, scale, and variability. For target distributions with sufficiently small tail probabilities, Dirichlet process mixtures of Gaussians provide strongly consistent density estimates [112, 113]. In addition, by allowing posterior covariances to vary across clusters, Dirichlet processes often provide more robust predictions than classic, asymptotically motivated bandwidth selection schemes [263]. Importantly, the Gibbs sampler of Alg. 2.3 also characterizes the posterior uncertainty in the estimated density.

Many other applications of Dirichlet process mixtures involve data generated from some finite, but unknown, number of latent factors [76, 121, 149, 289]. In such cases, the parameters corresponding to different clusters are typically of interest. Several different complexity criteria [87, 203, 314], including Bayesian formulations which optimize predictive likelihoods [46], have been proposed in this context. For applications involving high–dimensional data, however, there may be inherent ambiguities which prevent reliable selection of a single "best" model. Dirichlet process mixtures avoid this issue via an infinite model encompassing finite mixtures of varying order. Mild conditions then guarantee that the Dirichlet process posterior, as characterized by Prop. 2.5.1, asymptotically concentrates on the true set of finite mixture parameters [149].

Other models for finite mixtures place an explicit prior on the number of clusters K, and then separately parameterize mixtures of each order [121, 208, 243]. When mixture weights follow finite Dirichlet distributions, this approach produces the *Dirichlet/multinomial allocation (DMA)* model [121]. In some applications, complex priors $p(K)$ can then be used to encode detailed prior knowledge. However, when less is known about the underlying generative process, these priors involve nuisance parameters which are difficult to specify uninformatively [243, 272]. Indeed, in some applications where the Dirichlet process has favorable asymptotic properties, apparently uninformative finite Dirichlet priors lead to inconsistent parameter estimates [149].

Computational considerations also practically motivate Dirichlet process priors. DMA models are typically learned via Monte Carlo methods which use Metropolis–Hastings moves to step between models of varying order [243, 272]. Such algorithms, including variants of *reversible jump MCMC* [9, 243], require proposal distributions which split, merge, and otherwise transform cluster parameters. Effective proposals must usually be tuned to particular applications, and can be difficult to formulate for hierarchical models of complex, high–dimensional data. While split–merge MCMC methods are readily generalized to Dirichlet process mixtures [55, 121, 151], the simple but effective collapsed Gibbs sampler (Alg. 2.3) has no direct analog for DMA models. For realistic datasets, differences between Dirichlet process and DMA models are often small, with Dirichlet processes exhibiting a slight posterior bias towards mixtures with a few additional, low–weight components [121].

Finally, we note that while Bayesian estimators derived from finite–dimensional models are usually consistent, the asymptotic behavior of nonparametric methods is more subtle [68, 113, 160, 317]. For example, Diaconis and Freedman [68] considered a semiparametric model in which a latent location parameter $\theta \sim \mathcal{N}(0, \Lambda)$, and the un-

known *measurement* distribution underlying independent observations has a Dirichlet process prior $\mathrm{DP}(\alpha, H)$. They demonstrated that a heavy–tailed, Student–t base measure H may then lead to inconsistent estimates of θ. As predicted by more recent theoretical results [113], consistency is regained for log–concave base measures. This and other examples [68, 317] demonstrate the need for careful empirical and, where possible, theoretical validation of nonparametric methods.

■ 2.5.4 Dependent Dirichlet Processes

Many applications involve complex, structured datasets, and cannot be directly posed as standard density estimation problems. In this section, we describe a framework for *dependent Dirichlet processes (DDPs)* [191] which extends nonparametric Bayesian methods to a rich family of hierarchical models.

Consider a continuous or discrete *covariate* space Ω capturing the temporal, spatial, or categorical structure associated with a given dataset. As in many hierarchical models, we associate each $\omega \in \Omega$ with a latent parameter $\theta(\omega)$, whose marginal distribution equals G_ω. Let $\theta = \{\theta(\omega) \mid \omega \in \Omega\}$, $\theta \in \Theta$, denote a global configuration of the parameters. We would like to design a flexible, nonparametric prior for the joint distribution $G(\theta)$. Generalizing the stick–breaking representation of Thm. 2.5.3, a DDP prior takes the following form:

$$G(\theta(\omega)) = \sum_{k=1}^{\infty} \pi_k(\omega)\delta(\theta(\omega), \theta_k(\omega)) \qquad \theta_k \sim H \tag{2.196}$$

In this construction, the base measure H is a stochastic process on Θ. For example, if parameters $\theta_k(\omega)$ are assigned Gaussian marginals H_ω, a Gaussian process provides a natural joint measure [105]. The infinite set of mixture weights then follow a generalized stick–breaking process:

$$\pi_k(\omega) = \beta_k(\omega) \prod_{\ell=1}^{k-1} (1 - \beta_\ell(\omega)) \qquad \beta_k \sim B \tag{2.197}$$

If the stochastic process B is chosen so that its marginals $\beta_k(\omega) \sim \mathrm{Beta}(1, \alpha)$, Thm. 2.5.3 shows that $G_\omega \sim \mathrm{DP}(\alpha, H_\omega)$. However, for appropriately chosen H and B, there will be interesting dependencies in the joint distribution G, implicitly coupling the measures for parameters $\theta(\omega)$ associated with different covariates. See MacEachern [191] for a discussion of conditions ensuring the existence of DDP models.

In the simplest case, the stick–breaking weights of eq. (2.197) are set to the same, constant value $\beta_k(\omega) = \bar{\beta}_k \sim \mathrm{Beta}(1, \alpha)$ for all covariates $\omega \in \Omega$. The resulting DDP models capture dependency by sampling joint parameters θ_k from an appropriately chosen stochastic process [60, 105, 191]. More generally, B may be designed to encourage mixture weights which vary to capture local features of the covariate space [122, 342]. In the following section, we describe a model which uses hierarchically dependent Dirichlet processes to choose weights distinguishing several groups of observations.

Hierarchical Dirichlet Processes

As in Sec. 2.2.4, consider a dataset with J related groups $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_J)$, where $\mathbf{x}_j = (x_{j1}, \ldots, x_{jN_j})$ contains N_j observations. Just as the LDA model [31] shares a finite set of clusters among such groups (see Fig. 2.11), the *hierarchical Dirichlet process (HDP)* [288, 289] provides a nonparametric approach to sharing infinite mixtures.

To construct an HDP, a *global* probability measure $G_0 \sim \text{DP}(\gamma, H)$ is first used to define a set of shared clusters:

$$G_0(\theta) = \sum_{k=1}^{\infty} \beta_k \delta(\theta, \theta_k) \qquad \begin{aligned} \boldsymbol{\beta} &\sim \text{GEM}(\gamma) \\ \theta_k &\sim H(\lambda) \qquad k = 1, 2, \ldots \end{aligned} \tag{2.198}$$

Group–specific mixture distributions $G_j \sim \text{DP}(\alpha, G_0)$ are then independently sampled from a Dirichlet process with discrete base measure G_0, so that

$$G_j(\theta) = \sum_{t=1}^{\infty} \widetilde{\pi}_{jt} \delta(\theta, \widetilde{\theta}_{jt}) \qquad \begin{aligned} \widetilde{\boldsymbol{\pi}}_j &\sim \text{GEM}(\alpha) \\ \widetilde{\theta}_{jt} &\sim G_0 \qquad t = 1, 2, \ldots \end{aligned} \tag{2.199}$$

Each *local* cluster in group j has parameters $\widetilde{\theta}_{jt}$ copied from some global cluster $\theta_{k_{jt}}$, which we indicate by $k_{jt} \sim \boldsymbol{\beta}$. As summarized in the graph of Fig. 2.28, data points in group j are then independently sampled according to this parameter distribution:

$$\begin{aligned} \bar{\theta}_{ji} &\sim G_j \\ x_{ji} &\sim F(\bar{\theta}_{ji}) \end{aligned} \tag{2.200}$$

For computational convenience, we typically define $F(\theta)$ to be an appropriate exponential family, and $H(\lambda)$ a corresponding conjugate prior. As with standard mixtures, eq. (2.200) can be equivalently expressed via a discrete variable t_{ji} indicating the cluster associated with the i^{th} observation:

$$\begin{aligned} t_{ji} &\sim \widetilde{\boldsymbol{\pi}}_j \\ x_{ji} &\sim F(\widetilde{\theta}_{jt_{ji}}) \end{aligned} \tag{2.201}$$

Fig. 2.29 shows an alternative graphical representation of the HDP, based on these explicit assignments of observations to local clusters, and local clusters to global clusters.

Because G_0 is discrete, each group j may create several different copies $\widetilde{\theta}_{jt}$ of the same global cluster θ_k. Aggregating the probabilities assigned to these copies, we can directly express G_j in terms of the distinct global cluster parameters:

$$G_j(\theta) = \sum_{k=1}^{\infty} \pi_{jk} \delta(\theta, \theta_k) \qquad \pi_{jk} = \sum_{t | k_{jt} = k} \widetilde{\pi}_{jt} \tag{2.202}$$

Groups then reuse a common set of global clusters in different proportions. Using Thm. 2.5.1, it can be shown that $\boldsymbol{\pi}_j \sim \text{DP}(\alpha, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ and $\boldsymbol{\pi}_j$ are interpreted as

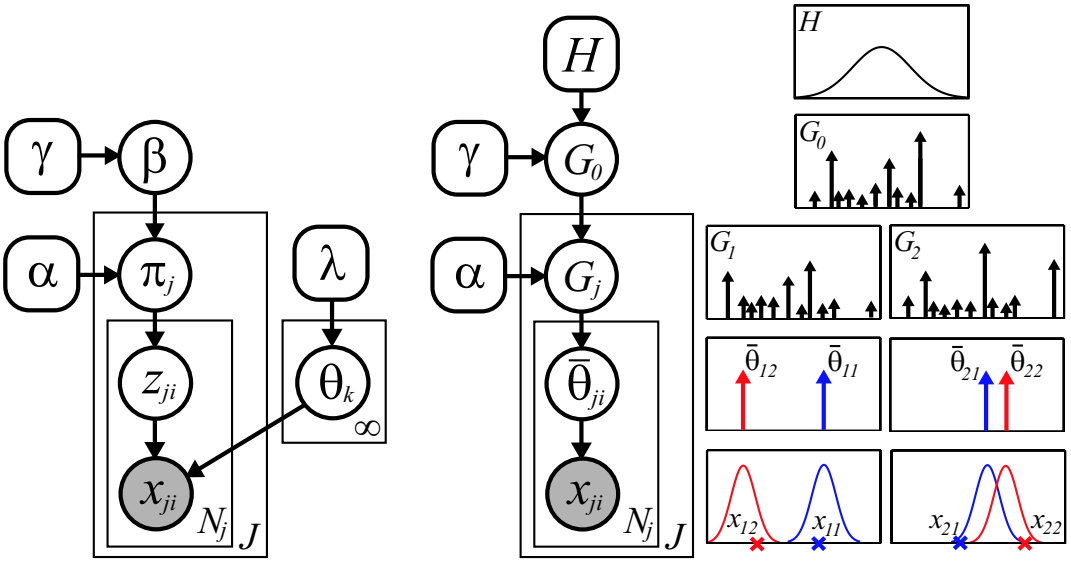


Figure 2.28. Directed graphical representations of a hierarchical Dirichlet process (HDP) mixture model. Global cluster weights $\boldsymbol{\beta} \sim \text{GEM}(\gamma)$ follow a stick–breaking process, while cluster parameters are assigned independent priors $\theta_k \sim H(\lambda)$. *Left:* Explicit stick–breaking representation, in which each group reuses the global clusters with weights $\boldsymbol{\pi}_j \sim \text{DP}(\alpha, \boldsymbol{\beta})$. $z_{ji} \sim \boldsymbol{\pi}_j$ indicates the cluster that generates $x_{ji} \sim F(\theta_{z_{ji}})$. *Right:* Alternative distributional form, in which $G_0 \sim \text{DP}(\gamma, H)$ is an infinite discrete distribution on Θ, and $G_j \sim \text{DP}(\alpha, G_0)$ a reweighted, group–specific distribution. $\bar{\theta}_{ji} \sim G_j$ are then the parameters of the cluster that generates $x_{ji} \sim F(\bar{\theta}_{ji})$. We illustrate with a shared, infinite Gaussian mixture, where cluster variances are known (bottom) and $H(\lambda)$ is a Gaussian prior on cluster means (top). Sampled cluster means $\bar{\theta}_{j1}, \bar{\theta}_{j2}$, and corresponding Gaussians, are shown for two observations x_{j1}, x_{j2} in each of two groups G_1, G_2.

measures on the positive integers [289]. Thus, $\boldsymbol{\beta}$ determines the average weight of local clusters ($\mathbb{E}[\pi_{jk}] = \beta_k$), while α controls the variability of cluster weights across groups. Note that eq. (2.202) suggests the alternative graphical model of Fig. 2.28, in which $z_{ji} \sim \boldsymbol{\pi}_j$ directly indicates the *global* cluster associated with x_{ji}. In contrast, Fig. 2.29 indirectly determines global cluster assignments via local clusters, taking $z_{ji} = k_{jt_{ji}}$.

Comparing these representations to Fig. 2.11, we see that HDPs share clusters as in the LDA model, but remove the need for model order selection. In terms of the DDP framework, the global measure G_0 provides a particular, convenient mechanism for inducing dependencies among the mixture weights in different groups. Note that the discreteness of G_0 plays a critical role in this construction. If, for example, we had instead taken $G_j \sim \text{DP}(\alpha, H)$ with H continuous, the stick–breaking construction of Thm. 2.5.3 shows that groups would learn independent sets of *disjoint* clusters.

Extending the analogy of Fig. 2.23, we may alternatively formulate the HDP representation of Fig. 2.29 in terms of a *Chinese restaurant franchise* [289]. In this interpretation, each group defines a separate restaurant in which customers (observations) x_{ji} sit at tables (clusters) t_{ji}. Each table shares a single dish (parameter) $\widetilde{\theta}_t$, which is

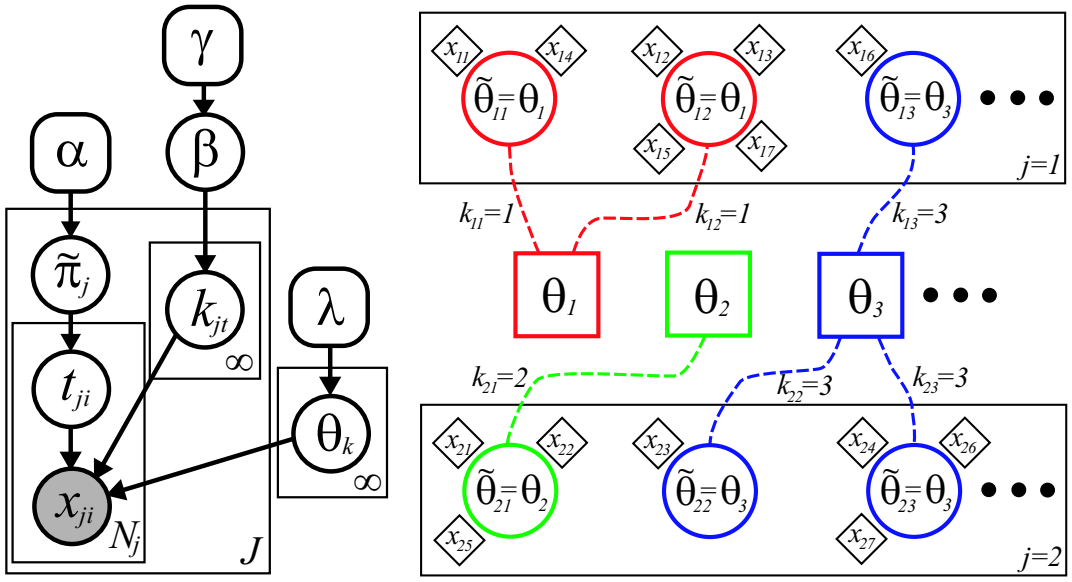


Figure 2.29. Chinese restaurant franchise representation of the HDP model of Fig. 2.28. *Left:* Global cluster parameters are assigned independent priors $\theta_k \sim H(\lambda)$, and reused by groups with frequencies $\boldsymbol{\beta} \sim \text{GEM}(\gamma)$. Each group j has infinitely many local clusters (tables) t, which are associated with a single global cluster $k_{jt} \sim \boldsymbol{\beta}$. Observations (customers) x_{ji} are independently assigned to some table $t_{ji} \sim \tilde{\boldsymbol{\pi}}_j$, and thus indirectly associated with the global cluster (dish) $\theta_{z_{ji}}$, where $z_{ji} = k_{jt_{ji}}$. *Right:* Example in which a franchise menu with dishes θ_k (squares, center) is shared among tables (circles, top and bottom) in two different restaurants (groups). All customers (diamonds) seated at a given table share the same dish (global cluster parameter).

ordered from a menu G_0 shared among restaurants (groups). As before, let k_{jt} indicate the global parameter $\theta_{k_{jt}}$ assigned to table t in group j, and $\mathbf{k}_j$ the parameters for all of that group's tables. We may then integrate over G_0 and G_j (as in eq. (2.181)) to find the conditional distributions of these indicator variables:

$$p(t_{ji} \mid t_{j1}, \ldots, t_{ji-1}, \alpha) \propto \sum_t N_{jt}\delta(t_{ji}, t) + \alpha\delta(t_{ji}, \bar{t}) \tag{2.203}$$

$$p(k_{jt} \mid \mathbf{k}_1, \ldots, \mathbf{k}_{j-1}, k_{j1}, \ldots, k_{jt-1}, \gamma) \propto \sum_k M_k\delta(k_{jt}, k) + \gamma\delta(k_{jt}, \bar{k}) \tag{2.204}$$

Here, M_k is the number of tables previously assigned to θ_k, and N_{jt} the number of customers already seated at the t^{th} table in group j. As before, customers prefer tables t at which many customers are already seated (eq. (2.203)), but sometimes choose a new table $\bar{t}$. Each new table is assigned a dish $k_{j\bar{t}}$ according to eq. (2.204). Popular dishes are more likely to be ordered, but a new dish $\theta_{\bar{k}} \sim H$ may also be selected.

The stick–breaking (Fig. 2.28) and Chinese restaurant franchise (Fig. 2.29) representations provide complementary perspectives on the HDP. In particular, they have each been used to design Monte Carlo methods which infer shared clusters from training

data [289]. In Chap. 5, we describe and extend a Gibbs sampler based on the Chinese restaurant franchise, generalizing the Dirichlet process sampler of Alg. 2.3.

Temporal and Spatial Processes

Models derived from, or related to, the DDP framework have been applied to several application domains. For example, an *analysis of densities* [296, 297] approach has been used to determine multiple related density estimates. This model is similar to the HDP of Fig. 2.28, except that the base measure G_0 is convolved with a Gaussian kernel to construct a continuous, global density estimate. Alternatively, for applications involving observed covariates, the DDP construction of eq. (2.196) has been used to design nonparametric models in which each cluster parameterizes a standard, Gaussian ANOVA model [60]. This method more robustly describes datasets which mix several different global correlation structures.

Related methods have been used to model temporal processes. In particular, *time–sensitive* Dirichlet process mixtures [342] consider applications where each observation has an associated time stamp. A generalization of the Chinese restaurant process then encourages observations at similar times to be associated with the same cluster. Dirichlet processes have also been used to develop an *infinite hidden Markov model* [16], avoiding explicit selection of a finite set of discrete states. The infinite HMM can be seen as a special case of the HDP, in which the global measure G_0 is used to couple the transition distributions associated with different latent states [289].

Gaussian processes provide a standard, widely used framework for modeling spatial data [285, 330]. Generalizing this approach, dependent Dirichlet processes have been used to construct infinite mixtures of Gaussian processes [105]. The marginal distribution at each spatial location is then a Dirichlet process mixture of Gaussians, but the Gaussian parameters associated with nearby sites are correlated. Like DDP models based on ANOVA clusters [60], these mixtures of Gaussian processes implicitly assume absolute spatial locations are statistically meaningful, and require replicated observations at *identical* sites. In Chap. 6, we develop a *transformed Dirichlet process* [282] adapted to datasets with different forms of spatial structure.