Sampling Methods and Approximations

Handout Notes - Generative and Deep Learning (GDL)

Davide Bacciu - University of Pisa

Notation. Random variables are uppercase (e.g., X, S, Z) and observed values are lowercase (e.g., x, s, z). A sample set is denoted by $\mathcal{X} = \{x^{(1)}, \ldots, x^{(L)}\}$, where L is the number of samples. A dataset is denoted by $\mathcal{D}$ with size $N = |\mathcal{D}|$. Expectations under a distribution p are written $\mathbb{E}_p[f(X)]$ and variances as $\mathrm{Var}_p[f(X)]$. When discussing sequential sampling methods, $x^{(\ell)}$ denotes the ℓ-th sampled state of a Markov chain, and $q(x' \mid x)$ denotes a transition kernel. In Bayesian models, θ denotes parameters and Z latent variables.

1 Why sampling appears in probabilistic learning

Sampling is one of the most important practical tools in probabilistic machine learning. At a conceptual level, it answers a simple question:

> *If we know a distribution $p(x)$, how can we generate realizations that behave as if they were drawn from it?*

This matters because many quantities of interest in probabilistic learning are expectations, posterior summaries, or predictive quantities. For example, if we want to compute

$$\mathbb{E}_p[f(X)],$$

then exact computation may require summing or integrating over a very large state space. When this is impossible, we can approximate the expectation by drawing samples from p and averaging the function values.

Sampling is therefore not only a way of generating synthetic data. It is also a way of *doing inference approximately* when exact probabilistic computation is intractable.

2 What sampling is

Let X be a random variable with distribution $p(x)$. A sampling procedure produces a finite set of realizations

$$\mathcal{X} = \{x^{(1)}, \ldots, x^{(L)}\}, \qquad x^{(\ell)} \sim p(x).$$

Each $x^{(\ell)}$ is a realization of the random variable X, and the whole set $\mathcal{X}$ is used to approximate properties of the distribution.

For instance, if X is the outcome of a fair die,

$$p(X = i) = \frac{1}{6}, \qquad i \in \{1, \ldots, 6\},$$

then a valid sampler might return a set such as

$$\mathcal{X} = \{5, 3, 2, 1, 5\}.$$

This is only a finite sample, so it does not look perfectly uniform. Yet as the number of samples grows, the empirical frequencies should approach the true probabilities.

3 Sampling for approximation of expectations

Suppose we want to compute the expectation of a function f under p:

$$\mathbb{E}_p[f(X)].$$

If we have samples $\mathcal{X} = \{x^{(1)}, \ldots, x^{(L)}\}$, we approximate this expectation with the sample average

$$\hat{f}_{\mathcal{X}} = \frac{1}{L} \sum_{\ell=1}^{L} f\left(x^{(\ell)}\right).$$

This is the basic Monte Carlo approximation.

The motivation is that in many models the target distribution is too complex for exact computation. Typical examples include:

- posteriors in hierarchical Bayesian models,
- models with non-conjugate priors,
- energy-based models where normalization is intractable,
- latent-variable models whose posterior cannot be marginalized in closed form.

Sampling allows us to turn integration into averaging. Instead of computing an expectation analytically, we estimate it by repeated random draws.

Worked example — Monte Carlo approximation of an expectation

Let X be a discrete random variable and suppose we want

$$\mathbb{E}_p[f(X)].$$

Given samples $\mathcal{X} = \{x^{(1)}, \ldots, x^{(L)}\}$ from p, define

$$\hat{f}_{\mathcal{X}} = \frac{1}{L} \sum_{\ell=1}^{L} f\left(x^{(\ell)}\right).$$

If the samples are independent and identically distributed from p, then

$$\mathbb{E}[\hat{f}_{\mathcal{X}}] = \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{E}_p[f(X)] = \mathbb{E}_p[f(X)],$$

so the estimator is unbiased. Moreover,

$$\mathrm{Var}(\hat{f}_{\mathcal{X}}) = \frac{1}{L^2} \sum_{\ell=1}^{L} \mathrm{Var}_p[f(X)] = \frac{1}{L} \mathrm{Var}_p[f(X)],$$

provided the variance is finite and the samples are independent. Thus the estimator becomes more stable as L increases.

4 Sampling in Bayesian learning

In Bayesian models, parameters are random variables. Instead of estimating a single point value, we often want to reason about their posterior distribution. For example, with latent variables Z and parameters θ, the posterior may take the form

$$p(\theta, Z \mid \mathcal{D}).$$

If this posterior is intractable, sampling provides a way to approximate it. By drawing samples

$$(\theta^{(1)}, z^{(1)}), \ldots, (\theta^{(L)}, z^{(L)}) \sim p(\theta, Z \mid \mathcal{D}),$$

we can approximate posterior expectations, marginal probabilities, or predictive distributions.

This is especially useful at test time. Given a new observation $x^\star$, Bayesian prediction can be approximated by averaging over posterior samples:

$$p(z^\star \mid x^\star, \mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})}[p(z^\star \mid x^\star, \theta)] \approx \frac{1}{L} \sum_{\ell=1}^{L} p(z^\star \mid x^\star, \theta^{(\ell)}).$$

5 What makes a sampler good?

A sampling algorithm is not judged only by whether it produces numbers. It is judged by the quality of the approximation induced by those numbers.

There are three central ideas:

- **Validity:** samples should eventually reflect the correct target distribution.
- **Unbiasedness of Monte Carlo estimators:** sample averages should have the correct expectation whenever possible.
- **Low variance:** estimates should not fluctuate too much across runs.

5.1 Empirical convergence

For a discrete variable, the empirical frequency of value i is

$$\frac{1}{L} \sum_{\ell=1}^{L} \mathbb{I}[x^{(\ell)} = i],$$

where $\mathbb{I}[\cdot]$ is the indicator function. A fundamental requirement of a good sampler is that

$$\lim_{L \to \infty} \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{I}[x^{(\ell)} = i] = p(X = i) \quad \text{almost surely.}$$

This is the most important asymptotic property: the empirical distribution should converge to the target.

5.2 Unbiasedness

A Monte Carlo estimator $\hat{f}_{\mathcal{X}}$ is unbiased if

$$\mathbb{E}[\hat{f}_{\mathcal{X}}] = \mathbb{E}_p[f(X)].$$

If samples are genuinely distributed according to $p(x)$, then the sample mean is unbiased.

5.3 Variance

Even an unbiased estimator may be unreliable if its variance is large. The variance of $\hat{f}_{\mathcal{X}}$ determines how much the estimate fluctuates from run to run. Low variance means that repeated sampling runs produce similar values, which makes the approximation practically useful.

6 Sampling from a univariate distribution

For a univariate discrete distribution, sampling is straightforward. Suppose

$$p(X = 1) = 0.4, \qquad p(X = 2) = 0.4, \qquad p(X = 3) = 0.2.$$

A simple method is based on a uniform random variable $U \sim \text{Uniform}(0, 1)$. Partition the unit interval according to cumulative probabilities:

$$[0, 0.4) \mapsto 1, \qquad [0.4, 0.8) \mapsto 2, \qquad [0.8, 1] \mapsto 3.$$

Then a draw of U uniquely determines the sampled value.

Worked example — Inverse-CDF style sampling for a discrete univariate distribution

Suppose

$$p(X = 1) = 0.4, \quad p(X = 2) = 0.4, \quad p(X = 3) = 0.2.$$

Define the cumulative distribution:

$$F(1) = 0.4, \qquad F(2) = 0.8, \qquad F(3) = 1.$$

Let $U \sim \text{Uniform}(0, 1)$ and define

$$X = \begin{cases} 1 & \text{if } 0 \le U < 0.4, \\ 2 & \text{if } 0.4 \le U < 0.8, \\ 3 & \text{if } 0.8 \le U \le 1. \end{cases}$$

Then

$$p(X = 1) = p(0 \le U < 0.4) = 0.4,$$
$$p(X = 2) = p(0.4 \le U < 0.8) = 0.4,$$
$$p(X = 3) = p(0.8 \le U \le 1) = 0.2.$$

Hence the procedure exactly samples from the desired distribution.

For univariate distributions, this idea is computationally cheap and exact. The real difficulty arises in the multivariate case.

7 Why multivariate sampling is hard

Now suppose the target distribution is a joint distribution over several variables:

$$p(s_1, \ldots, s_n).$$

A single sample is now a vector

$$x^{(\ell)} = (s_1^{(\ell)}, \ldots, s_n^{(\ell)}).$$

If each variable has C states, then the joint state space has size C^n. This exponential growth is the source of the difficulty.

A naive strategy would be to flatten the full joint into a single categorical variable with C^n values and sample from it as if it were univariate. This is exact in principle, but computationally infeasible except for tiny systems.

A second naive strategy uses the chain rule:

$$p(s_1, \ldots, s_n) = p(s_1)\, p(s_2 \mid s_1) \cdots p(s_n \mid s_1, \ldots, s_{n-1}).$$

One could then sample sequentially:

$$s_1 \sim p(s_1), \qquad s_2 \sim p(s_2 \mid s_1), \qquad \ldots, \qquad s_n \sim p(s_n \mid s_1, \ldots, s_{n-1}).$$

This is again exact in principle. However, computing these conditionals may itself be exponentially expensive if the joint distribution is not already represented in a factorized form.

8 Ancestral sampling in directed graphical models

If the joint distribution is represented as a Bayesian network, the chain-rule factorization is already encoded in the graph. In that case, sequential sampling becomes practical.

Suppose the network factors as

$$p(s_1, \ldots, s_n) = \prod_{i=1}^{n} p(s_i \mid \mathrm{pa}(s_i)),$$

where $\mathrm{pa}(s_i)$ denotes the parents of node s_i. If we sample variables in a topological order of the graph, then each variable can be drawn directly from its local conditional distribution once its parents have been sampled. This is called *ancestral sampling*.

Worked example — Why ancestral sampling is exact

Assume a directed acyclic graph with variables ordered so that parents come before children. Suppose we sample sequentially as

$$s_1^{(\ell)} \sim p(s_1 \mid \mathrm{pa}(s_1)), \quad s_2^{(\ell)} \sim p(s_2 \mid \mathrm{pa}(s_2)), \quad \ldots \quad s_n^{(\ell)} \sim p(s_n \mid \mathrm{pa}(s_n)).$$

Because each parent set is already instantiated when its child is sampled, the probability of generating one full sample $x^{(\ell)} = (s_1^{(\ell)}, \ldots, s_n^{(\ell)})$ is

$$\prod_{i=1}^{n} p(s_i^{(\ell)} \mid \mathrm{pa}(s_i^{(\ell)})) = p(s_1^{(\ell)}, \ldots, s_n^{(\ell)}).$$

Therefore each complete sample is drawn exactly from the joint distribution. If the random draws used at different iterations are independent, then the samples themselves are independent.

This makes ancestral sampling highly attractive whenever it is applicable: it is exact, simple, and usually low-variance because successive samples are independent.

9 Why evidence complicates things

In inference we are rarely interested in the unconditional joint. We usually want a posterior distribution under evidence:

$$p(s_{\setminus e} \mid s_e) = \frac{p(s_{\setminus e}, s_e)}{p(s_e)},$$

where s_e denotes observed variables and $s_{\setminus e}$ denotes the remaining hidden variables.

Ancestral sampling is poorly suited to this setting. Two naive ideas exist:

- modify the graph to account for clamped evidence, which may be as hard as exact inference;
- sample from the original joint and reject all samples inconsistent with the evidence.

The rejection strategy becomes extremely wasteful when the evidence is unlikely. This is why posterior sampling requires different methods.

10 Gibbs sampling: local conditional resampling

Gibbs sampling is one of the simplest and most widely used approximate sampling methods. Instead of drawing a completely new sample independently at each step, it starts from an initial full configuration

$$x^{(1)} = (s_1^{(1)}, \dots, s_n^{(1)})$$

and updates one variable at a time while keeping the others fixed.

At iteration $\ell + 1$, choose a variable s_j and sample

$$s_j^{(\ell+1)} \sim p(s_j \mid s_{\setminus j}^{(\ell)}),$$

where $s_{\setminus j}^{(\ell)}$ denotes the current values of all other variables. All other coordinates remain unchanged:

$$s_i^{(\ell+1)} = s_i^{(\ell)}, \qquad i \neq j.$$

In Bayesian networks, this local conditional depends only on the *Markov blanket* of s_j: its parents, its children, and the other parents of its children. This is what makes Gibbs practical.

Worked example — Deriving the Gibbs conditional from local factors

Let S_j be one variable in a Bayesian network and let $s_{\setminus j}$ denote the current values of all other variables. The full conditional is

$$p(s_j \mid s_{\setminus j}) = \frac{p(s_j, s_{\setminus j})}{\sum_{s'_j} p(s'_j, s_{\setminus j})}.$$

Using the network factorization, every factor not involving s_j cancels between numerator and denominator. The only remaining terms are those in which s_j appears:

$$p(s_j \mid s_{\setminus j}) \propto p(s_j \mid \mathrm{pa}(s_j)) \prod_{k \in \mathrm{ch}(j)} p(s_k \mid \mathrm{pa}(s_k)),$$

where $\mathrm{ch}(j)$ denotes the children of node j. Thus the conditional depends only on the Markov blanket of S_j, not on the full graph.

Evidence is easy to handle in Gibbs sampling: simply never update the observed variables. They remain clamped throughout the chain.

11 Why Gibbs sampling works

At first sight, Gibbs sampling seems strange. The sample at iteration $\ell + 1$ is not drawn directly from $p(x)$ but from a transition kernel

$$q(x^{(\ell+1)} \mid x^{(\ell)}).$$

So why does it eventually produce samples from the desired target?

The answer is that Gibbs sampling defines a Markov chain over the state space. Under suitable conditions (irreducibility and aperiodicity), this Markov chain has a unique stationary distribution. For Gibbs, that stationary distribution is the target distribution $p(x)$.

Thus Gibbs is not exact at finite time in the same sense as ancestral sampling. Instead, it is *asymptotically valid*: after running long enough, the distribution of the chain approaches the target.

Worked example — Stationarity intuition for Gibbs sampling

Let $q(x' \mid x)$ be the Gibbs transition kernel. The distribution of the chain after ℓ steps is some distribution $v^{(\ell)}$ over states. The chain evolves as

$$v^{(\ell+1)}(x') = \sum_x q(x' \mid x)\, v^{(\ell)}(x).$$

A stationary distribution $v^\star$ satisfies

$$v^\star(x') = \sum_x q(x' \mid x)\, v^\star(x).$$

For Gibbs sampling, one can show that $v^\star(x) = p(x)$, so the target distribution is invariant under the Gibbs update. If the chain is ergodic, repeated application of the kernel drives the chain toward this stationary distribution:

$$v^{(\ell)} \to p \qquad \text{as } \ell \to \infty.$$

12 Properties and limitations of Gibbs sampling

Gibbs sampling has important strengths:

- it handles evidence naturally,
- it exploits local conditional structure,
- it is easy to derive in many graphical models.

But it also has weaknesses:

- the first samples may be far from stationarity (*burn-in*),
- successive samples are highly dependent,
- mixing can be slow when variables are strongly correlated.

These dependencies increase estimator variance compared with i.i.d. exact samplers. For this reason, practitioners often:

- discard an initial burn-in phase,
- keep only every K-th sample (*thinning* or subsampling),
- run multiple chains from different initializations.

13 Gibbs sampling as part of MCMC

Gibbs sampling is a member of the broader Markov Chain Monte Carlo (MCMC) family. The general MCMC philosophy is:

> *Construct a Markov chain whose stationary distribution is the target distribution $p(x)$.*

The transition kernel $q(x' \mid x)$ is not unique; many choices are possible. What matters is that the resulting chain be:

- **irreducible**: every state can eventually be reached,
- **aperiodic**: the chain does not get trapped in deterministic cycles.

Under these conditions, the chain has a unique stationary distribution.

This framework includes:

- Gibbs sampling,

- Metropolis–Hastings,
- particle methods for sequential models,
- hybrid / Hamiltonian Monte Carlo,
- and many other specialized samplers.

Different MCMC methods define different transition kernels, and different kernels lead to different convergence and variance properties.

14 Convergence issues in MCMC

Unlike exact sampling, MCMC raises a difficult practical question:

How do we know whether the chain has converged?

In general, convergence cannot be detected perfectly from finite samples. Instead, one relies on heuristics and diagnostics:

- trace plots of sampled quantities,
- running means and variances,
- multiple chains started from different initial states,
- checks for stabilization of summary statistics.

The convergence speed depends strongly on the transition kernel and on initialization. In practice, one often has to wait "long enough" and then assess whether the resulting chain appears stable.

15 Sampling in Bayesian models: a generic workflow

When a Bayesian model has an intractable posterior, a common strategy is:

1. define the target posterior
$$p(\theta, Z \mid \mathcal{D});$$
2. design an MCMC transition kernel q whose stationary distribution is that posterior;
3. initialize the chain at some $(\theta^{(0)}, z^{(0)})$;
4. run the chain, discarding burn-in and possibly thinning the samples;
5. approximate posterior quantities by Monte Carlo averages over the retained sample set.

If $\mathcal{S}$ is the retained set of samples, prediction for a new data point $x^\star$ can be approximated as

$$p(z^\star \mid x^\star, \mathcal{S}) \approx \frac{1}{|\mathcal{S}|} \sum_{\theta^\star \in \mathcal{S}} p(z^\star \mid x^\star, \theta^\star).$$

This is the Bayesian sampling analogue of posterior averaging.

16 Sampling-based approximate inference in LDA

Latent Dirichlet Allocation (LDA) is a canonical example where approximate posterior inference is required. The latent variables include topic assignments and document-level topic proportions, and the posterior is intractable in closed form.

A Gibbs-style strategy updates latent variables by sampling from conditional distributions derived from the current state of the model. At a high level, one alternates between:

- sampling topic assignments for words,
- sampling or integrating document-level topic proportions,
- updating global topic-word parameters.

The resulting procedure is an iterative approximate inference method whose convergence is typically monitored through the joint probability or log-posterior. The exact algebra of these conditionals can be intricate, but the main principle is the same as before: sample each latent block from its conditional distribution given the others.

Worked example — Collapsed-Gibbs intuition in topic models

Suppose a model contains topic assignments Z, document-topic proportions Θ, and topic-word probabilities β. A full Gibbs scheme might sample each block from its conditional posterior:

$$Z \sim p(Z \mid W, \Theta, \beta, \alpha), \qquad \Theta \sim p(\Theta \mid W, Z, \beta, \alpha), \qquad \beta \sim p(\beta \mid W, Z, \Theta, \alpha).$$

If some variables can be integrated out analytically before sampling, the resulting chain may mix better. This is the principle behind *collapsed Gibbs sampling*: fewer variables are sampled, but the sampled conditionals become more informative.