

# Spike and Slab Priors

## 1 Introduction

A spike and slab prior for a random variable  $X$  is a generative model—i.e., a prior—in which  $X$  either attains some fixed value  $v$ , called *the spike*, or is drawn some other prior  $p_{\text{slab}}(x)$ , called *the slab*. In the case that  $v = 0$ ,  $X$  is either zero, or drawn from some other prior; in this case, the spike and slab prior is *sparsity inducing*, offering a principled alternative to e.g. sparsity-inducing regularisers.

The usual way of constructing a spike and slab prior is to introduce a latent variable  $Z \sim \text{Ber}(\theta)$  where  $Z = 0$  means that  $X$  attains the fixed value  $v$  and  $Z = 1$  means that  $X$  is drawn from the slab  $p_{\text{slab}}(x)$ :

$$\begin{aligned} Z &\sim \text{Ber}(\theta), \\ X \mid Z = 0 &\sim \delta(x - v), \\ X \mid Z = 1 &\sim p_{\text{slab}}(x). \end{aligned}$$

Marginalising over  $Z$ , we equivalently have that

$$X \sim \theta p_X(x) + (1 - \theta)\delta(x - v),$$

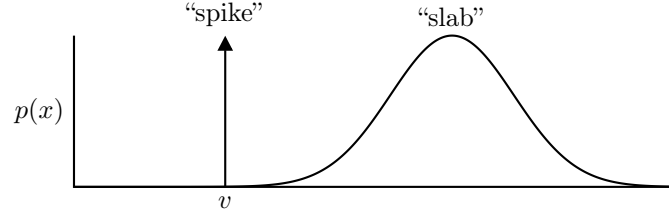
which we recognise as a mixture model with mixture components  $p_X(x)$  and  $\delta(x - v)$ , respectively having weights  $\theta$  and  $1 - \theta$ . [Figure 1](#) illustrates  $p(x)$  in the case of a Gaussian slab.

## 2 Linear Regression with a Spike and Slab Prior

Let  $Y$  be an  $\mathbb{R}$ -valued random variable representing our observations at some point  $x \in \mathbb{R}^n$ , and consider the usual model for linear regression:

$$Y \mid \beta, x \sim \mathcal{N}(\langle \beta, x \rangle, \sigma^2)$$

where  $\langle \beta, x \rangle$  denotes the inner product between  $\beta$  and  $x$ . In the case that  $x$  is high dimensional, we might not have enough data to accurately estimate the coefficients  $\beta$ . One way to mitigate this issue is to build zeros into  $\beta$ , and putting a spike and slab prior on  $\beta$  is a perfectly viable approach to do

Figure 1: The density  $p(x)$  in the case of a Gaussian slab

so:

$$\begin{aligned} Z_i &\sim \text{Ber}(\theta), \\ \beta_i \mid Z_i = 0 &\sim \delta(\beta_i), \\ \beta_i \mid Z_i = 1 &\sim \mathcal{N}(0, \tau^{-1}). \end{aligned}$$

We can equivalently formulate the resulting model in a slightly more compact and convenient form:

$$\begin{aligned} Z_i &\sim \text{Ber}(\theta), \\ \beta_i &\sim \mathcal{N}(0, \tau^{-1}), \\ Y \mid Z, \beta, x &\sim \mathcal{N}(\langle z \circ \beta, x \rangle, \sigma^2) \end{aligned}$$

where  $\circ$  denotes the Hadamard product. Indeed,  $(z \circ \beta)_i = z_i \beta_i = 0$  if  $z_i = 0$  and similarly  $(z \circ \beta)_i = \beta_i$  if  $z_i = 1$ .

Upon observing data  $\mathcal{D} = (x^{(t)}, y^{(t)})_{t=1}^T$ , we wish to compute our posterior belief about  $\beta$  and  $Z$ :

$$p(z, \beta \mid \mathcal{D}) = \frac{1}{p(\mathcal{D})} p(z) p(\beta) \prod_{t=1}^T p(y^{(t)} \mid z, \beta, x^{(t)})$$

where  $p(\mathcal{D})$  denotes the *evidence*:

$$p(\mathcal{D}) = \int p(z) p(\beta) \prod_{t=1}^T p(y^{(t)} \mid z, \beta, x^{(t)}) \, dz \, d\beta.$$

Unfortunately,  $p(\mathcal{D})$  is hard to compute, because it requires summing over the  $2^n$  values that  $Z$  can attain, and it is not clear how to efficiently do so. We must therefore resort to *approximate inference*.

To perform inference in models employing a spike and slab prior, a sampling-based approach, *Gibbs sampling* in particular, is often used. Gibbs sampling states that given some initial  $Z^{(0)}$  and  $\beta^{(0)}$ ,

iterating

$$\begin{aligned}
Z_1^{(i)} &\sim p(z_1 | z_{2:n}^{(i-1)}, \beta^{(i-1)}, \mathcal{D}), \\
Z_2^{(i)} &\sim p(z_2 | z_1^{(i)}, z_{3:n}^{(i-1)}, \beta^{(i-1)}, \mathcal{D}), \\
&\vdots \\
Z_n^{(i)} &\sim p(z_n | z_{1:n-1}^{(i)}, \beta^{(i-1)}, \mathcal{D}), \\
\beta^{(i)} &\sim p(\beta | z^{(i)}, \mathcal{D})
\end{aligned}$$

will eventually yield samples from the joint posterior:

$$(Z^{(i)}, \beta^{(i)}) \sim p(z, \beta | \mathcal{D}) \quad \text{for large enough } i.$$

Fortunately, these conditionals are easy to compute<sup>1</sup>:<sup>[✓]</sup>

$$\begin{aligned}
\log p(z_i | z_{-i}, \beta, \mathcal{D}) &\simeq \log p(z_i) + \log p(\mathcal{D} | z, \beta) \\
&\simeq \log p(z_i) + \frac{1}{\sigma^2} \langle z \circ \beta, \hat{\mu} \rangle - \frac{1}{2\sigma^2} \langle z \circ \beta, \hat{\Sigma}(z \circ \beta) \rangle, \\
\hat{\Sigma} &= \sum_{t=1}^T x^{(t)} x^{(t)\top}, \\
\hat{\mu} &= \sum_{t=1}^T x^{(t)} y^{(t)},
\end{aligned}$$

and

$$\begin{aligned}
p(\beta | z, \mathcal{D}) &\propto p(\beta) p(\mathcal{D} | z, \beta) \\
&\propto \mathcal{N}(\beta; (\tau\sigma^2 I + \tilde{\Sigma})^{-1} \tilde{\mu}, \sigma^2 (\tau\sigma^2 I + \tilde{\Sigma})^{-1}) \\
\tilde{\Sigma} &= \sum_{t=1}^T (z \circ x^{(t)}) (z \circ x^{(t)})^\top, \\
\tilde{\mu} &= \sum_{t=1}^T z \circ x^{(t)} y^{(t)}.
\end{aligned}$$

One can now sample and happily compute expectations under the posterior distribution.

**Remark 1.** Although the generative model specifies each weight  $\beta_i$  to be either zero or nonzero, the posterior over  $Z_i$  will not conclude either case: the posterior over  $Z_i$  instead assigns probabilities to both possibilities of  $\beta_i$  being zero or nonzero. Therefore, the posterior distribution does not yield a “sparse solution”, but rather a weighting of all possible sparse solutions. And this makes perfect sense: only in the limit of infinite data can the model conclude a weight to be zero.

---

<sup>1</sup> In the case that the conditionals cannot be computed analytically, one could use another MCMC method, often Metropolis–Hastings, to sample from the conditionals, yielding a composite procedure often referred to as *Metropolis–Hastings within Gibbs*.

### 3 Diagnosing MCMC

#### 3.1 Big O Notation and Convergence of Simple Monte Carlo Estimates

To begin with, let us quickly recap how the number of samples relates to the accuracy of a Monte Carlo estimate.

**Definition 1** (Small O: Convergence in Probability). If  $(X_n)$  is a sequence of random variables and  $(a_n)$  a sequence of constants, then  $X_n = o_P(a_n)$  means that for every  $\varepsilon > 0$ ,

$$\lim_{n \rightarrow \infty} P(|X_n/a_n| < \varepsilon) = 1.$$

If  $X_n = o_P(a_n)$ , then that  $X_n$  will eventually become arbitrarily close to  $a_n$  in probability; in other words, asymptotically  $X_n$  behaves like  $a_n$ .

**Definition 2** (Big O: Stochastic Boundedness). If  $(X_n)$  is a sequence of random variables and  $(a_n)$  a sequence of constants, then  $X_n = O_P(a_n)$  means that for every  $\varepsilon > 0$  there exist  $\delta_\varepsilon > 0$  and  $N_\varepsilon > 0$  such that

$$P(|X_n/a_n| \leq \delta_\varepsilon) \geq 1 - \varepsilon \quad \text{for all } n \geq N_\varepsilon.$$

If  $X_n = O_P(a_n)$ , then that means that for every  $\varepsilon > 0$ , we can identify a region around  $a_n$  that eventually will contain  $X_n$  with probability at least  $1 - \varepsilon$ ; in other words, asymptotically  $X_n$  is finitely far from  $a_n$ .

Now, consider the simple Monte Carlo estimator

$$X_n = \frac{1}{n} \sum_{i=1}^n X_i,$$

where all  $X_i$  are drawn i.i.d. Let  $\mathbb{E}X_i = \mu$  and regard  $\mathbb{V}X_i = \sigma^2$  as a constant. We then find that

**Proposition 1.** If  $(X_i)$  are drawn i.i.d., then  $\frac{1}{n} \sum_{i=1}^n X_i - \mu = O_p(1/\sqrt{n})$ .

In other words, to gain a digit more accurate results, one needs 100 times more samples.

#### 3.2 Convergence of MCMC Estimates

Unfortunately, the analysis in the foregoing section does not apply to averages computed with samples from a Markov chain: in that case, the samples are not i.i.d., but *correlated* instead.

**Definition 3** (Effective Sample Size (ESS)). For  $n$  samples with autocorrelation  $\rho_t$ , the *effective sample size* is  $n_{\text{ESS}} = n/\tau$  where

$$\tau = 1 + 2 \sum_{i=1}^{\infty} \rho_i.$$

For the purpose of computing averages, the effective sample size  $n_{\text{ESS}}$  is the number independent samples “contained” in correlated samples from a Markov chain. Using this number of “effective samples”  $n_{\text{ESS}}$ , one can apply the analysis from the foregoing section.

**Proposition 2.** If  $(X_i)$  are drawn from a Markov chain, then

$$\frac{1}{n} \sum_{i=1}^n X_i - \mu = O_p(1/\sqrt{n_{\text{ESS}}}).$$

In other words, to gain a digit more accurate results, one needs 100 times more effective samples.

### 3.3 Geweke Test

Testing MCMC code is often difficult [GD14]: algorithms are stochastic, algorithms may perform badly for reasons other than incorrect implementations, and good performance is often a matter of judgement.

We discuss one way of testing correctness of an MCMC algorithm, a test called the *Geweke test* [Gewe04]. Sampling from a joint distribution  $p(x, z)$  can be done in two different ways:

- (1) Sample  $(X, Z)$  from the generative model: first sample  $Z \sim p(z)$  and then  $X | Z \sim p(x | z)$ .
- (2) Start with a sample  $X \sim p(x)$  from the generative model, and produce a sample from  $Z | X \sim p(z | x)$  using the MCMC algorithm. Finally, resample  $X | Z \sim p(x | x)$ .

The Geweke test tests that (1) and (2) produce samples from the same distribution. To do this, one can follow [GD14] and simply employ a P–P plot; Figure 2 illustrates a negative result, an indeterminate result, and a positive result.

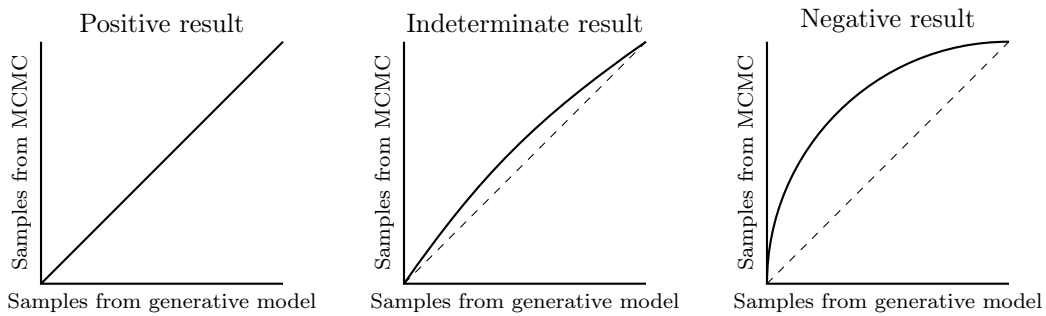


Figure 2: Various outcomes of the P–P in a Geweke test

## References

- [GD14] R. B. Grosse and D. K. Duvenaud. “Testing MCMC Code”. In: *arXiv preprint arXiv:1412.5218* (Dec. 2014). eprint: <https://arxiv.org/abs/1412.5218> (cit. on p. 5).
- [Gew04] John Geweke. “Getting It Right: Joint Distribution Tests of Posterior Simulators”. In: *Journal of the American Statistical Association* 99.467 (2004), pp. 799–804 (cit. on p. 5).