A Level-set Hit-and-run Sampler for Quasi-Concave Distributions

Shane T. Jensen and Dean P. Foster
Department of Statistics, The Wharton School, University of Pennsylvania

Abstract

We develop a new sampling strategy that uses the hit-and-run algorithm within level sets of a target density. Our method can be applied to any quasi-concave density, which covers a broad class of models. Standard sampling methods often perform poorly on densities that are high-dimensional or multi-modal. Our level set sampler performs well in high-dimensional settings, which we illustrate on a spike-and-slab mixture model. We also extend our method to exponentially-tilted quasi-concave densities, which arise in Bayesian models consisting of a log-concave likelihood and quasi-concave prior density. We illustrate our exponentially-tilted level-set sampler on a Cauchy-normal model where our sampler is better able to handle a high-dimensional and multi-modal posterior distribution compared to Gibbs sampling and Hamiltonian Monte Carlo.

1 Introduction

Complex Bayesian models are often estimated by sampling random variables from complicated distributions. This strategy is especially prevalent when Markov Chain Monte Carlo (MCMC) simulation is used to estimate the posterior distribution of a set of unknown parameters. The most common MCMC technique is the Gibbs sampler (Geman and Geman, 1984), where small subsets of parameters are sampled from their conditional posterior distribution given the current values of all other parameters.

With high dimensional parameter spaces, component-wise strategies such as the Gibbs sampler

can encounter problems such as high autocorrelation and the inability to move between local modes. Hamiltonian Monte Carlo (Duane *et al.*, 1987; Neal, 2011) is an alternative MCMC strategy that uses Hamiltonian dynamics to improve convergence to high-dimensional target distributions. Hoffman and Gelman (2011) introduce the No-U-Turn sampler that extends Hamiltonian Monte Carlo. The No-U-Turn sampler has been implemented for a general set of models in the Stan software package (Stan Development Team, 2013).

In high dimensions, one would ideally employ a strategy that sampled the high-dimensional parameter vector directly, instead of a small set of components at a time. In this paper, we develop a sampling algorithm that provides direct samples from a high-dimensional *quasi-concave* density. As we discuss in Section 2.1, quasi-concave densities are a broad class that covers many real data models. We also provide an extension of our algorithm to provide direct samples from a high-dimensional (and potentially multi-modal) *exponentially-tilted quasi-concave* density.

Our procedure is based on the fact that any horizontal slice through a quasi-concave density f will give a convex level set above that slice. By slicing the quasi-concave density f at a sequence of different heights, we divide the density f into a sequence of convex level sets. We then use the hit-and-run algorithm to sample high-dimensional points x within each of these convex level sets, while simultaneously estimating the volume of each convex level set.

As reviewed in Vempala (2005), recent work suggests that the hit-and-run algorithm is an efficient way to sample from high-dimension convex set as long as the start is "warm" (which we discuss in Section 2.1). The hit-and-run algorithm is not as commonplace as other sampling methods (e.g. Metropolis-Hastings), though Chen and Schmeiser (1996) discuss using hit-and-run Monte Carlo to evaluate multi-dimensional integrals.

In Section 2.1, we review several recent results for hit-and-run sampling in convex sets. In Section 2.2,

we outline our procedure for ensuring a warm start within each convex slice, thereby giving us an efficient way of sampling from the entire quasi-concave density. In Section 3, we present an empirical comparison that suggests our "level-set hit-and-run sampling" methodology is much more efficient than Gibbs sampling or Hamiltonian Monte Carlo for high dimensional quasi-concave posterior densities.

We also extend our method to sample efficiently from *exponentially-tilted quasi-concave* densities in Section 2.4. Exponentially-tilted quasi-concave densities are very common in Bayesian models: the combination of a quasi-concave prior density and a log-concave likelihood leads to an exponentially-tilted quasi-concave posterior density. In Section 4, we illustrate the efficiency of our method on a exponentially-tilted quasi-concave posterior density from a model consisting of a Normal data likelihood and a Cauchy prior density. It should be noted that this posterior density can be multi-modal, depending on the relative locations of the prior mode and observed data. Our results in Section 4 suggest that our sampling strategy can accurately estimate posterior distributions which are both high-dimensional and multi-modal.

The popular Bayesian software program WinBUGS (Lunn *et al.*, 2000) uses adaptive rejection sampling Gilks (1992) to obtain samples from log-concave posterior distributions. Our approach can obtain samples from the more general class of quasi-concave (Section 2.1) and exponentially-tilted quasi-concave (Section 2.3) posterior distributions. Our approach has a similar spirit to slice sampling (Neal, 2003) but differs substantially in that we use hit-and-run within horizontal slices of the target density, rather than sampling uniformly along a horizontal line at a randomly-sampled density height. Our hit-and-run strategy allows us more easily obtain samples in high dimensions, whereas multivariate versions of slice sampling are more difficult to implement unless a component-wise strategy is employed.

2 Level-Set Sampling Methodology

2.1 Quasi-Concave Densities and Level Sets

Let $f()$ be a density in a high dimensional space. A density function f is quasi-concave if:

$$C_a = \{\boldsymbol{x} \,|\, f(\boldsymbol{x}) > a\}$$

is convex for all values of a. In other words, the level set C_a of a quasi-concave density f is convex for any value a. Let $\mathcal{Q}$ denote the set of all quasi-concave densities and $\mathcal{D}$ denote the set of all concave densities. All

concave densities are quasi-concave, $\mathcal{D} \subset \mathcal{Q}$, but the converse is not true. Quasi-concave densities are a very broad class that contains posterior densities for many common Bayesian models, including the normal, gamma, student's t, and uniform densities.

Our proposed method is based on computing the volume of a convex level set C by obtaining random samples $\boldsymbol{x}$ from C. Let us assume we are already at a point $\boldsymbol{x}^0$ in C. A simple algorithm for obtaining a random sample $\boldsymbol{x}$ would be a *ball walk*: pick a uniform point $\boldsymbol{x}$ in a ball of size δ around $\boldsymbol{x}^0$, and move to $\boldsymbol{x}$ only if $\boldsymbol{x}$ is still in C.

An alternative algorithm is *hit-and-run*, where a random direction $\boldsymbol{d}$ is picked from current point $\boldsymbol{x}^0$. This direction will intersect the boundary of the convex set C at some point $\boldsymbol{x}^1$. A new point $\boldsymbol{x}$ is sampled uniformly along the line segment defined by direction $\boldsymbol{d}$ and end points $\boldsymbol{x}^0$ and $\boldsymbol{x}^1$, and is thus guaranteed to remain in C.

Lovasz (1999) showed that the hit-and-run algorithm mixes rapidly from a *warm start* in a convex body. The warm start criterion is designed to ensure that our starting point is not stuck in some isolated corner of the convex body.

Vempala (2005) suggests that a random point from convex set $C' \in C$ provides a warm start for convex set C as long as $\text{volume}(C')/\text{volume}(C) \geq 0.5$. Vempala (2005) also presents several results that suggest the hit-and-run algorithm mixes more rapidly than the ball walk algorithm.

2.2 LSHR1: Level-Set Hit-and-run Sampler for Quasi-concave Densities

Our level-set hit-and-run algorithm iteratively samples from increasingly larger level sets of the quasi-concave density, while ensuring that each level set provides a warm start for the next level set. The samples from each level set are then weighted appropriately to provide a full sample from the quasi-concave probability density. We briefly describe our LSHR1 algorithm below, with details and pseudo-code given in our supplementary materials.

Our sampling method must be initialized at the maximum value $\boldsymbol{x}_{\max}$ of the quasi-concave density $f(\cdot)$, which could either be known or found by an optimization algorithm. Our algorithm begins by taking a small level set $C_1 = \{\boldsymbol{x} : f(\boldsymbol{x}) \geq t_1\}$ centered around that maximum value. Starting from $\boldsymbol{x}_{\max}$, we run a hit-and-run sampler within this initial level set for m iterations. Each iteration of the hit-and-run sampler picks a random direction $\boldsymbol{d}$ from current point $\boldsymbol{x}$, which defines a line segment with endpoints at

the edge of the convex level set C_1. The endpoints are found by bisection starting from points outside C_1 along $\boldsymbol{d}$. Once the endpoints of the line segment have been found, a new point $\boldsymbol{x}'$ is sampled uniformly along this line segment, which ensures that $\boldsymbol{x}'$ remains in C_1. We use $\{\boldsymbol{x}\}_1$ to denote the set of all points sampled from C_1 using this hit-and-run method.

After sampling m times from convex level set C_1, we need to pick a new threshold $t_2 < t_1$ that defines a second level set of the quasi-concave probability density $f(\cdot)$. For our sampler to stay *warm* (Section 2.1), we need t_2 to define a convex shape C_2 with volume V_2 less than twice the volume V_1, i.e. $R_{1:2} = V_1/V_2 \geq 0.5$.

We propose a new threshold t_{prop} and then run m iterations of the hit-and-run sampler within the new convex level set C_{prop} defined by $f(\boldsymbol{x}) > t_{\text{prop}}$. We then estimate $\hat{R}_{\text{prop}}$ as the proportion of these sampled points from convex level set C_{prop} that were also contained within the previous level set C_1. We only accept the proposed threshold t_{prop} if $0.55 \leq \hat{R}_{\text{prop}} \leq 0.8$. The lower bound fulfills the *warm* criterion and the *ad hoc* upper bound fulfills our desire for efficiency: we do not want the new level set to be too close in volume to the previous level set.

If the proposed threshold is accepted, we re-define $t_{\text{prop}} \equiv t_2$ and then the convex level set $C_{\text{prop}} \equiv C_2$ becomes our current level set. The collection of sampled points $\{\boldsymbol{x}\}_2$ from C_2 are retained, as well as the estimated ratio of level set volumes $\hat{R}_{1:2}$. Our supplementary materials outlines an adaptive method for proposing a new threshold if the proposed threshold is not accepted.

In this same fashion, we continue to add level sets C_k defined by threshold t_k based on comparison to the previous level sets C_{k-1}. The level set algorithm terminates when the accepted threshold t_k is lower than some pre-specified lower bound K on the probability density $f(\cdot)$.

The result of our LSHR1 procedure is n level sets, represented by a vector of thresholds $(t_1, \ldots t_n)$, a vector of estimated ratios of volumes $(\hat{R}_{1:2}, \ldots, \hat{R}_{n-1:n})$, and the level set collections: a set of m sampled points from each level set $(\{\boldsymbol{x}\}_1, \ldots, \{\boldsymbol{x}\}_n)$.

We obtain L essentially independent[1] samples from the quasi-convex density $f(\cdot)$ by sub-sampling points from the level set collections $(\{\boldsymbol{x}\}_1, \ldots, \{\boldsymbol{x}\}_n)$ with each level set i represented proportional to its probability p_i. The probabilities for each level set is the

product of its height and volume,

$$p_i = (t_{i-1} - t_i) \times V_i$$

/ We have our volume ratios $\hat{R}_{i:i+1}$ as estimates of V_i/V_{i+1}, which allows us to estimate p_i by first calculating

$$\hat{q}_i = (t_{i-1} - t_i) \times \prod_{j=i}^{n} \hat{R}_{j:j+1} \qquad i = 1, \ldots, n \quad (1)$$

where t_0 is the maximum of $f(\cdot)$ and $\hat{R}_{n:n+1} = 1$, and then $\hat{p}_i = \hat{q}_i / \sum \hat{q}_j$.

Further details and pseudo-code for our LSHR1 algorithm are given in our supplementary materials. We demonstrate our LSHR1 algorithm on a spike-and-slab density in Section 3.

2.3 Exponentially-Tilted Quasi-concave Densities

A density $g()$ is a log-concave density function if $\log g()$ is a concave density function. Let $\mathcal{L}$ denote the set of all log-concave density functions. All log-concave density functions are also concave density functions ($\mathcal{L} \subset \mathcal{D}$), and thus all log-concave density functions are also quasi-concave density functions ($\mathcal{L} \subset \mathcal{Q}$).

Bagnoli and Bergstrom (2005) gives an excellent review of log-concave densities. The normal density is log-concave whereas the student's t and Cauchy density are not. The gamma and beta densities are log-concave only under certain parameter settings, e.g. both the uniform and exponential densities are log-concave. The beta(a, b) density with other parameter settings (e.g. $a < 1, b < 1$) can be neither log-concave nor quasi-concave.

Now, let T denote the set of *exponentially-tilted* quasi-concave density functions. A density function h is an exponentially-tilted quasi-concave density function if there exists a quasi-concave density $f \in \mathcal{Q}$ such that $f(\boldsymbol{x})/h(\boldsymbol{x}) = \exp(\boldsymbol{\beta}'\boldsymbol{x})$. Exponentially-tilted quasi-concave densities are a generalization of quasi-concave densities, so $\mathcal{Q} \subset \mathcal{T}$.

These three classes of functions (log-concave, quasi-concave, and exponentially-tilted quasi-concave) are linked by the following important relationship: if $X \sim f$ where $f \in \mathcal{Q}$ and $Y|X \sim g$ where $g \in \mathcal{L}$, then $X|Y \sim h$ where $h \in \mathcal{T}$.

The consequences of this relationship is apparent for the Bayesian modeling. The combination of a quasi-concave prior density for parameters $f(\boldsymbol{\theta})$ and a log-concave likelihood for data $g(\boldsymbol{y}|\boldsymbol{\theta})$ will produce an exponentially-tilted quasi-concave posterior density $h(\boldsymbol{\theta}|\boldsymbol{y})$.

---

[1] Although there is technically some dependence between successive sampled points from the hit-and-run sampler, our scheme of sub-sampling randomly from level set collections $(\{\boldsymbol{x}\}_1, \ldots, \{\boldsymbol{x}\}_n)$ essentially removes this dependence

Examples of log-concave likelihoods for data $g(\boldsymbol{y}|\boldsymbol{\theta})$ include the normal density, exponential density and uniform density. Quasi-concave priors for parameters $f(\boldsymbol{\theta})$ are an even broader class of densities, including the normal density, the t density, the Cauchy density and the gamma density. In Section 4, we examine an exponentially-tilted quasi-concave posterior density resulting from the combination of a multivariate normal density with a Cauchy prior density.

We now extend our level-set hit-and-run sampling methodology to exponentially-tilted quasi-concave densities, which makes our procedure applicable to a large class of Bayesian models consisting of quasi-concave priors and log-concave likelihoods.

2.4 LSHR2: Level-Set Hit-and-Run Sampler for Exponentially-Tilted Quasi-concave Densities

In Section 2.2, we presented our level set hit-and-run algorithm for sampling $\boldsymbol{x}$ from quasiconcave density $f(\boldsymbol{x})$. We now extend our algorithm to sample from an exponentially-tilted quasi-concave density $h(\boldsymbol{x})$.

As mentioned in Section 2.3, exponentially-tilted quasi-concave densities commonly arise as posterior distributions in Bayesian models. In keeping with the usual notation for Bayesian models, we replace our previous variables $\boldsymbol{x}$ with parameters $\boldsymbol{\theta}$. These parameters $\boldsymbol{\theta}$ have an exponentially-tilted quasi-concave posterior density $h(\boldsymbol{\theta}|\boldsymbol{y})$ arising from a log-concave likelihood $g(\boldsymbol{y}|\boldsymbol{\theta})$ and quasi-concave prior density $f(\boldsymbol{\theta})$.

Our LSHR2 algorithm starts just as before, by taking a small level set centered around the maximum value $\boldsymbol{\theta}_{\max}$ of the quasi-concave prior density $f(\cdot)$. This initial level set is defined, in part, as the convex shape of the probability density $f(\cdot)$ above an initial density threshold t_1.

However, we now augment our sampling space with an extra variable p where $p < \log g(\boldsymbol{y}|\boldsymbol{\theta})$. Letting $\boldsymbol{\theta}^\star = (\boldsymbol{\theta}, p)$, our convex shape is now

$$D_1 = \{\boldsymbol{\theta}^\star : f(\boldsymbol{\theta}) \geq t_1 \,,\, p < \log g(\boldsymbol{y}|\boldsymbol{\theta})\}$$

Within this new convex shape, we run an exponentially-weighted version of the hit-and-run algorithm. Specifically, a random $(d+1)$-dimensional direction $\boldsymbol{d}$ is sampled which, along with the current $\boldsymbol{\theta}^\star$, defines a line segment with endpoints at the boundaries of D_1. Instead of sampling uniformly along this line segment, we sample a new point $(\boldsymbol{\theta}^\star)'$ from points on this line segment weighted by $\exp(p)$. The remainder of the LSHR2 algorithm proceeds in the same fashion as our LSHR1 algorithm: we construct a schedule of decreasing thresholds $t_1, t_2, \ldots$

and corresponding level sets $D_1, D_2, \ldots$ such that each level set D_k is a warm start for the next level set D_{k+1}.

Within each of these steps, the exponentially-weighted hit-and-run algorithm is used to sample values $\boldsymbol{\theta}^\star$ within the current level set. As before, D_k is a warm start for the next level set D_{k+1} if the ratio of volumes[2] $\hat{R}_{k:k+1} = V_k/V_{k+1}$ that lies within the range $0.55 \leq \hat{R}_{k:k+1} \leq 0.8$ The algorithm terminates when our decreasing thresholds t_k achieve a pre-specified lower bound K.

Our procedure results in n level sets $(D_1, \ldots, D_n)$, represented by a vector of thresholds $(t_1, \ldots t_n)$, a vector of estimated ratios of volumes $(\hat{R}_{1:2}, \ldots, \hat{R}_{n-1:n})$, and the level set collections: a set of m sampled points from each level set:

$$(\{\boldsymbol{\theta}^\star\}_1, \ldots, \{\boldsymbol{\theta}^\star\}_n)$$

Finally, we obtain L samples $(\boldsymbol{\theta}_1^\star, \ldots, \boldsymbol{\theta}_L^\star)$ by sub-sampling points from the level set collections $(\{\boldsymbol{\theta}^\star\}_1, \ldots, \{\boldsymbol{\theta}^\star\}_n)$, with each level set i represented proportional to its probability p_i. The probability of each level set is still calculated as in (1).

By simply ignoring the sampled dimension p, we are left with samples $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L)$ from the exponentially-tilted quasi-convex posterior density $h(\boldsymbol{\theta}|\boldsymbol{y})$. Details of our LSHR2 algorithm are given in the supplementary materials. We demonstrate our LSHR2 algorithm on a Cauchy-normal model in Section 4.

3 Example 1: Spike-and-slab Model

We illustrate our LSHR1 sampler on a multivariate density which consists of a 50-50 mixture of two normal distributions, both centered at zero but with different variances. Specifically, the first component has variance $\boldsymbol{\Sigma}_0$ with off-diagonal elements equal to zero and diagonal elements equal to 0.05, whereas the second component has variance $\boldsymbol{\Sigma}_0$ with off-diagonal elements equal to zero and diagonal elements equal to 3. Figure 1 gives this spike-and-slab density in a single dimension.

The spike-and-slab density is commonly used as a mixture model for important versus spurious predictors in variable selection models (George and McCulloch, 1997).

This density is quasi-concave and thus amenable to our proposed level-set hit-and-run sampling methodology. Sampling from this density in a single di-

---

[2]We continue to use the term "volume" even though these convex shapes are a combination of d-dimensional θ and the extra log-density p dimension.

Spike and Slab Density in One Dimension

Figure 1: Spike-and-slab density: gray lines indicate density of the two components, black line is the mixture density
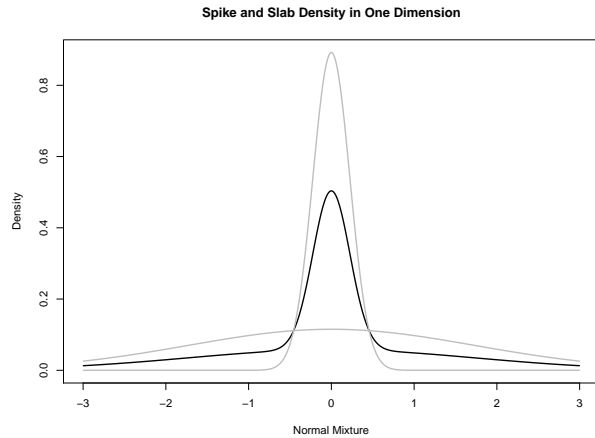
mension can be easily accomplished by Gibbs sampling using data augmentation. However, we will see that Gibbs sampling or Hamiltonian Monte Carlo will not perform adequately in higher dimensions whereas our LSHR1 sampling strategy continues to show good performance in high dimensions.

3.1 Previous Methods for Spike and Slab Density

Before exploring the results from our level-set hit-and-run procedure, we first introduce two current methods for obtaining samples from the spike-and-slab density: Gibbs sampling (Geman and Geman, 1984) and the Stan software (Stan Development Team, 2013) based on Hamiltonian Monte Carlo (Neal, 2011; Hoffman and Gelman, 2011).

The usual Gibbs sampling approach to obtaining a sample x from a mixture density is to augment the variable space with an indicator variable I where $I = 1$ indicates the current mixture component from which x is drawn. The algorithm iterates between

1. Sample x from mixture component dictated by I, i.e.
$$x \sim \begin{cases} N(\mathbf{0}, \boldsymbol{\Sigma}_1) & \text{if} \quad I = 1 \\ N(\mathbf{0}, \boldsymbol{\Sigma}_0) & \text{if} \quad I = 0 \end{cases}$$

2. Sample I with probability based on x:
$P(I = 1) = \frac{\phi(x, \mathbf{0}, \boldsymbol{\Sigma}_1)}{\phi(x, \mathbf{0}, \boldsymbol{\Sigma}_1) + \phi(x, \mathbf{0}, \boldsymbol{\Sigma}_1)}$

where $\phi(x, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the density of a multivariate normal with mean μ and variance $\boldsymbol{\Sigma}$ evaluated at x.

This Gibbs sampling algorithm mixes well when x has a small number of dimensions ($d < 10$), but in higher dimensions, it is difficult for the algorithm to

move between the two components. We see this behavior in Figure 2, where we evaluate results from running the Gibbs sampler for 100000 iterations on the spike-and-slab density with different dimensions ($d = 2, 5, 10, 15$). Specifically, we plot the proportion of the Gibbs samples taken from the first component (the spike) for x of different dimensions.

The mixing of the sampler for lower dimensions ($d = 2$ and $d = 5$) is reasonable, but we can see that for higher dimensions the sampler is extremely sticky and does not mix well at all. In the case of $d = 10$, the sampler only makes a couple moves between the spike component and the slab component during the entire run. In the case of $d = 15$, the sampler does not move into the second component at all during the entire 100000 sample run.

Figure 2: Empirical mixing proportion after each iteration is calculated as the proportion of samples up to that iteration that are from the first component (the spike). Gray lines indicate the true mixing proportion of 0.5
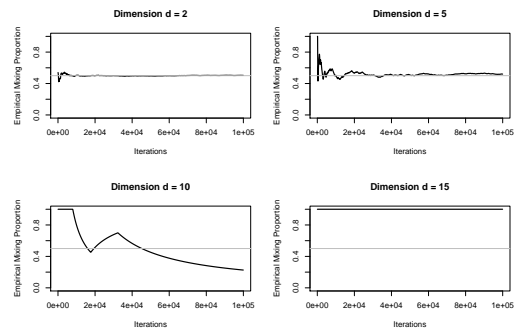
In high dimensions, we can estimate how often the Gibbs sampler will switch from one domain to the other. When the sampler is in one of the components, the variable x will mostly have a norm of $||\mathbf{x}||_2^2 \approx d\sigma^2$ where $\sigma = \sigma_0 = .05$ for component zero and $\sigma = \sigma_1 = 3.0$ for component one. For a variable currently classified into the zero component, the probability of a switch is

$$P\left(I = 1 \,|\, ||\mathbf{x}||_2^2 \approx d\sigma_0^2\right) \approx \left(\frac{\sigma_0\sqrt{e}}{\sigma_1}\right)^d$$

where the approximation holds if $\sigma_0 \ll \sigma_1$. Details of this calculation are given in the supplementary materials. This result suggests that the expected number of iterations until a switch is about $(\sigma_1/\sigma_0\sqrt{e})^d = 36.4^d$. This approximate switching time will be more accurate for large d.

The Stan software (Stan Development Team, 2013) employs Hamiltonian Monte Carlo (Neal, 2011; Hoffman and Gelman, 2011) to sample directly from the

spike-and-slab density rather than using a data augmentation strategy. Since sampling of the component indicator I is avoided, the Stan software has the potential for improved mixing compared to the Gibbs sampler in this situation. We will see in Section 3.2 that this is indeed the case: the Stan software outperforms the Gibbs sampler for this spike-and-slab model. However, both the Stan software and the Gibbs Sampler are outperformed by our level-set hit-and-run (LSHR1) methodology.

3.2 Level-set Sampler for Spike and Slab

Our LSHR1 sampling methodology was implemented on the same spike-and-slab distribution. Our sampling method does not depend on a data augmentation scheme that moves between the two components. Rather, we start from the mode and move outwards in convex level sets defined by thresholds on the density, while running a hit-and-run algorithm within each level set. In this example, the number of thresholds needed scaled linearly with the dimension d, suggesting that our method scales to higher dimensions.

We compare our LSHR1 sampling strategy to both the Gibbs sampling and Stan software alternatives by examining the empirical density of the sampled values in any particular dimension. In Figure 3, we plot the true quantiles of the spike-and-slab density against the quantiles of the first dimension of sampled values from the Gibbs sampler, the Stan software, and our LSHR1 sampler.

We see that for low dimensions ($d = 2$), samples from all three methods are an excellent match to the correct spike-and-slab distribution. However, for higher dimensions ($d = 20$), the Gibbs sampler provides a grossly inaccurate picture of the true distribution, due to the fact that the Markov chain never escapes the spike component of the distribution. The samples from the Stan software are a better match than the Gibbs sampler but still do not match the true distribution particularly well. The samples from our level-set hit-and-run sampler provides a superior match to the true distribution in the $d = 20$ case. We have checked even higher dimensions and the level-set hit-and-run sampler still provides a good match to the true distribution.

4 Example 2: Cauchy-normal Model

We illustrate our LSHR2 algorithm on a Bayesian model consisting of a multivariate normal likelihood

Figure 3: Quantile-quantile plots of samples versus true spike-and-slab distribution. Left is low dimension ($d = 2$), right is high dimension ($d = 20$). Top row is Gibbs sampler, middle row is Stan software, bottom row is our LSHR1 sampler
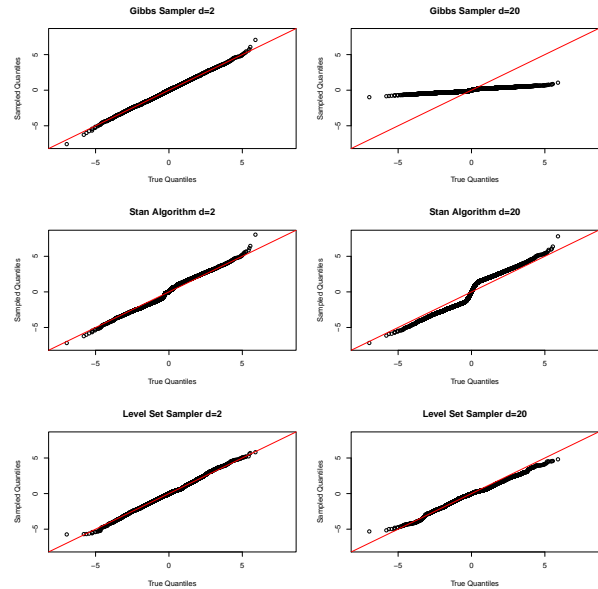
and a Cauchy prior density,

$$\begin{aligned} \boldsymbol{y}|\boldsymbol{\theta} &\sim \text{Normal}(\boldsymbol{\theta}, \sigma^2 \mathbf{I}) \\ \boldsymbol{\theta} &\sim \text{Cauchy}(\mathbf{0}, \mathbf{I}) \end{aligned} \tag{2}$$

where $\boldsymbol{y}$ and $\boldsymbol{\theta}$ are d dimensions and σ^2 is fixed and known. As mentioned in Section 2.3, this combination of a log-concave density $g(\boldsymbol{y}|\boldsymbol{\theta})$ and a quasi-concave density for $f(\boldsymbol{\theta})$ gives an *exponentially-tilted* quasi-concave posterior density $h(\boldsymbol{\theta}|\boldsymbol{y})$.

For some combinations of σ^2 and observed $\boldsymbol{y}$ values, the posterior density $h(\boldsymbol{\theta}|\boldsymbol{y})$ is multi-modal. Figure 4 gives examples of this multi-modality in one and two dimensions. The one-dimensional $h(\boldsymbol{\theta}|\boldsymbol{y})$ in Figure 4a has $y = 10$ and $\sigma^2 = 10.84$, whereas the two-dimensional $h(\boldsymbol{\theta}|\boldsymbol{y})$ in Figure 4 has $\boldsymbol{y} = (10, 10)$ and $\sigma^2 = 12.57$.

In higher dimensions ($d \geq 3$), it is difficult to evaluate (or sample from) the true posterior density $h(\boldsymbol{\theta}|\boldsymbol{y})$ which we need in order to have a gold standard for comparison between the Gibbs sampler and LSHR2 algoirithm. Fortunately, for this simple model, there is a rotation procedure that we detail in our supplementary materials that allows us to accurately estimate the true posterior density $h(\boldsymbol{\theta}|\boldsymbol{y})$.

4.1 Gibbs Sampling for Cauchy Normal Model

The Cauchy-normal model (2) can be estimated via Gibbs sampling by augmenting the parameter space

Figure 4: Posterior density $h(\boldsymbol{\theta}|\boldsymbol{y} = 10)$ for Cauchy-normal model in (a) one and (b) two dimensions.
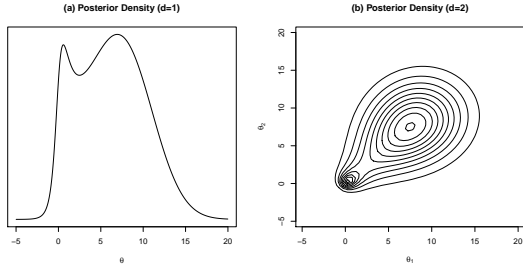
with an extra scale parameter τ^2,

$$
\begin{aligned}
\boldsymbol{y}|\boldsymbol{\theta} &\sim \text{Normal}(\boldsymbol{\theta}, \sigma^2\mathbf{I}) \\
\boldsymbol{\theta}|\tau^2 &\sim \text{Normal}(\mathbf{0}, \tau^2\mathbf{I}) \\
(\tau^2)^{-1} &\sim \text{Gamma}(1/2, 1/2)
\end{aligned}
\tag{3}
$$

Marginalizing over τ^2 gives us the same Cauchy$(\mathbf{0}, \mathbf{I})$ prior for $\boldsymbol{\theta}$. The Gibbs sampler iterates between sampling from the conditional distribution of $\boldsymbol{\theta}|\tau^2, \boldsymbol{y}$:

$$
\boldsymbol{\theta}|\tau^2, \boldsymbol{y} \sim \text{Normal}\left(\frac{\frac{1}{\sigma^2}\boldsymbol{y}}{\frac{1}{\sigma^2} + \frac{1}{\tau^2}}, \frac{1}{\frac{1}{\sigma^2} + \frac{1}{\tau^2}}\mathbf{I} \right) \tag{4}
$$

and the conditional distribution of $\tau^2|\boldsymbol{\theta}, \boldsymbol{y}$:

$$
(\tau^2)^{-1}|\boldsymbol{\theta}, \boldsymbol{y} \sim \text{Gamma}\left(\frac{d+1}{2}, \frac{d+\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta}}{2} \right) \tag{5}
$$

For several different dimensions d, we ran this Gibbs sampler for one million iterations, with the first 500000 iterations discarded at burn-in. In Figure 5, we compare the posterior samples for the first dimension θ_1 from our Gibbs sampler to the true posterior distribution for $d = 1, 2, 10$ and 20.

We see that the Gibbs sampler performs poorly at exploring the full posterior space in dimensions greater than one. Even in two dimensions, the Gibbs sampler struggles to fully explore both modes of the posterior distribution and this problem is exacerbated in dimensions higher than two.

### 4.2 Stan software for Cauchy Normal Model

The Stan software (Stan Development Team, 2013) can also be used to obtain posterior samples of $\boldsymbol{\theta}$ from the Cauchy Normal model (2) using Hamiltonian Monte Carlo (Neal, 2011; Hoffman and Gelman, 2011). For the same set of dimensions ($d = 1, 2, 10$ and 20), we ran the Stan algorithm for one million iterations with the first 500000 iterations discarded at burn-in.

Figure 5: Posterior samples of the first dimension θ_1 of $\boldsymbol{\theta}$ from Gibbs sampler for Cauchy-normal model in different dimensions d. Red curve in each plot represents the true posterior density.
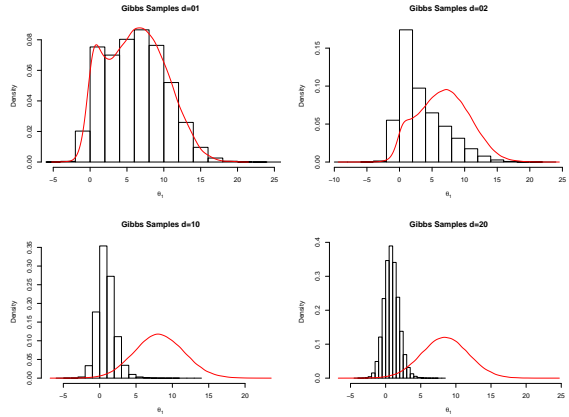
In Figure 6, we compare the posterior samples for the first dimension θ_1 from the Stan software to the true posterior distribution for $d = 1, 2, 10$ and 20.

Figure 6: Posterior samples of the first dimension θ_1 of $\boldsymbol{\theta}$ from the Stan software for Cauchy-normal model in different dimensions d. Red curve in each plot represents the true posterior density.
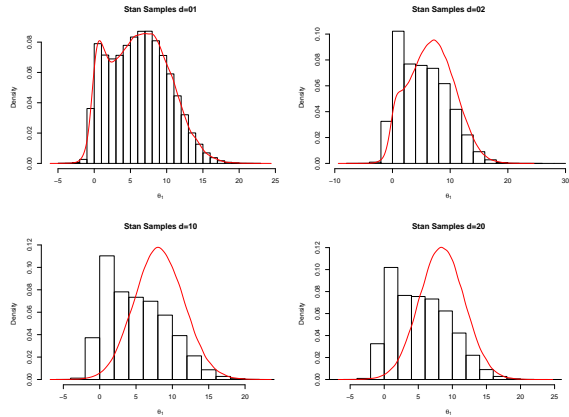
We see that the Stan software performs better than the Gibbs sampler in higher dimensions, but still performs quite poorly at estimating the true posterior distribution in dimensions greater than one. Both the Gibbs sampler and the Stan software struggle to explore both posterior modes, especially in higher dimensions.

### 4.3 LSHR2 sampling for Cauchy-Normal Model

Our LSHR2 level set sampling methodology was implemented on the same Cauchy-normal model. We start from the prior mode and move outwards in con-

vex level sets, while running an exponentially-tilted hit-and-run algorithm within each level set in order to get samples from the posterior distribution. As outlined in Section 2.4, each convex level set is defined by a threshold on the density that is slowly decreased in order to assure that each level set has a warm start. Just as with the spike-and-slab model, the number of needed thresholds scaled linearly with the dimension which suggests our algorithm would scale to even higher dimensions.

In Figure 7, we compare the posterior samples for the first dimension θ_1 from our LSHR2 sampler to the true posterior distribution for $d = 1, 2, 10$ and 20. Our LSHR2 level set sampler (with exponential

Figure 7: Posterior samples of the first dimension θ_1 of $\boldsymbol{\theta}$ from LSHR2 sampler for Cauchy-normal model in different dimensions d. Red curve in each plot represents the true posterior density.
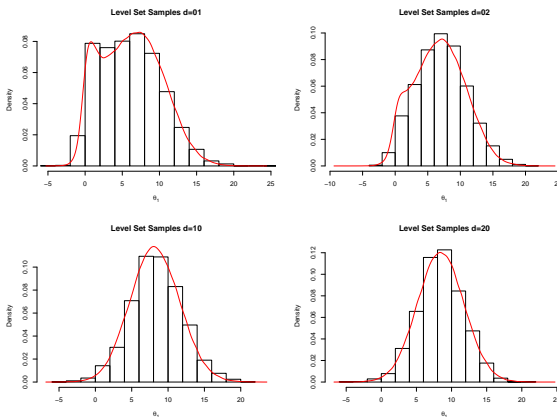
tilting) samples closely match the true posterior density in both low and high-dimensional cases. Comparing Figures 5 and Figures 7 clearly suggests that our level set hit-and-run methodology gives more accurate samples from the Cauchy-normal model in dimensions higher than one, compared with either the Gibbs sampler (Section 4.1) or the Stan software (Section 4.2).

5   Discussion

In this paper, we have developed a general sampling strategy based on dividing a density into a series of level sets, and running the hit-and-run algorithm within each level set. Our basic level-set hit-and-run sampler (LSHR1) can be applied to the broad class of quasi-concave densities, which includes many distributions used in applied statistical modeling. We illustrate our LSHR1 sampler on spike-and-slab density (Section 3), where our procedure performs much better in high dimensions than the

standard Gibbs sampler or the Stan algorithm (Stan Development Team, 2013) that implements Hamiltonian Monte Carlo (Neal, 2011; Hoffman and Gelman, 2011).

We also extend our sampling methodology to an exponentially-tilted level-set hit-and-run sampler (LSHR2). We illustrate our LSHR2 sampler on the posterior density from a Cauchy normal model (Section 4) where our method is much more effective than Gibbs sampling or Hamiltonian Monte Carlo. Although both of our examples are rather simple for the sake of illustration, we believe that our level-set hit-and-run sampling procedure has broad applicability to a wide variety of models. For example, exponentially-tilted quasi-concave densities arise frequently in Bayesian models as the posterior distribution formed from the combination of a log-concave likelihood and quasi-concave prior density. These model classes are more general than the log-concave densities that can be implemented in the popular Bayesian software WinBUGS (Lunn *et al.*, 2000).

A weakness of our approach that we must start from a modal value of the quasi-concave density. An optimization algorithm would be needed in situations where a mode is not known *a priori*. We do not consider this constraint to be particularly limiting since hill-climbing is typically an easier task (especially for quasi-concave densities) compared to the difficulty of sampling in high dimensions. We also saw in Section 4 that only a prior mode is required as a starting point for our exponentially-tilted level-set hit-and-run sampler. Starting from a known prior mode, we were able to explore multiple unknown posterior modes. Our procedure is able to sample effectively from this multi-modal posterior density even in high dimensions, where the Gibbs sampler performs poorly. This example demonstrates that our methodology can handle a multi-modal posterior density, which represents a substantial advance over alternative sampling technology such as the Gibbs sampler or Hamiltonian Monte Carlo (as implemented in the Stan software).

References

Bagnoli, M. and Bergstrom, T. (2005). Log-concave probability and its applications. *Economic Theory* **26**, 445–469.

Chen, M.-H. and Schmeiser, B. W. (1996). General hit-and-run monte carlo sampling for evaluating multidimensional integrals. *Operations Research Letters* **19**, 161–169.

Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics Letters B* **195**, 216–222.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **6**, 721–741.

George, E. I. and McCulloch, R. E. (1997). Approaches for bayesian variable selection. *Statistica Sinica* **7**, 339–373.

Gilks, W. R. (1992). Derivative-free adaptive rejection sampling for Gibbs sampling. *Bayesian Statistics* **4**.

Hoffman, M. D. and Gelman, A. (2011). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *arXiv:1111.4246v1* .

Lovasz, L. (1999). Hit-and-run mixes fast. *Mathematical Programming* **86**, 443–461.

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS – a Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing* **10**, 325–337.

Neal, R. M. (2003). Slice sampling. *Annals of Statistics* **31**, 705–767.

Neal, R. M. (2011). MCMC for using hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X. Meng, eds., *Handbook of Markov Chain Monte Carlo*, 113–162. Chapman & Hall, Boca Raton, FL.

Stan Development Team (2013). Stan: A c++ library for probability and sampling, version 1.3. *http://mc-stan.org/* .

Vempala, S. (2005). Geometric random walks: A survey. *Combinatorial and Computational Geometry* **52**, 573–612.