

Concave Convex Adaptive Rejection Sampling

Dilan Görür and Yee Whye Teh

Gatsby Computational Neuroscience Unit,
University College London
`{dilan,ywteh}@gatsby.ucl.ac.uk`

Abstract. We describe a method for generating independent samples from arbitrary density functions using adaptive rejection sampling without the log-concavity requirement. The method makes use of the fact that a function can be expressed as a sum of concave and convex functions. Using a concave convex decomposition, we bound the log-density using piecewise linear functions for and use the upper bound as the sampling distribution. We use the same function decomposition approach to approximate integrals which requires only a slight change in the sampling algorithm.

1 Introduction

Probabilistic graphical models have become popular tools for addressing many machine learning and statistical inference problems in recent years. This has been especially accelerated by general-purpose inference toolkits like BUGS [1], VIBES [2], and infer.NET [3], which allow users of graphical models to specify the models and obtain posterior inferences given evidence without worrying about the underlying inference algorithm. These toolkits rely upon approximate inference techniques that make use of the many conditional independencies in graphical models for efficient computation.

By far the most popular of these toolkits, especially in the Bayesian statistics community, is BUGS. BUGS is based on Gibbs sampling, a Markov chain Monte Carlo (MCMC) sampler where one variable is sampled at a time conditioned on its Markov blanket. When the conditional distributions are of standard form, samples are easily obtained using standard algorithms [4]. When these conditional distributions are not of standard form (e.g. if non-conjugate priors are used), a number of MCMC techniques are available, including slice sampling [5], adaptive rejection sampling (ARS) [6], and adaptive rejection Metropolis sampling (ARMS) [7].

Adaptive rejection sampling, described in Section 2, is a rejection sampling technique that produces true samples from a given distribution. In contrast, slice sampling and ARMS are MCMC techniques that produce true samples only in the limit of a large number of MCMC iterations (the specific number of iterations required being unknown in most cases of interest). Though this is not a serious drawback in the Gibbs sampling context of BUGS, there are other occasions

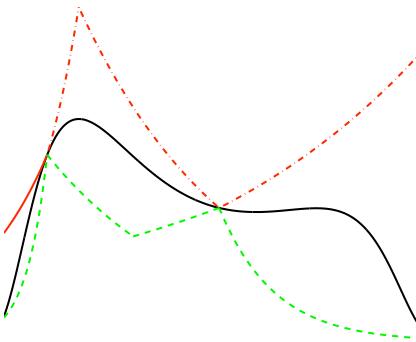


Fig. 1. The piecewise exponential upper and lower bounds on a multi-modal function, constructed by using CCARS. The two points where the upper and lower bounds touch the function are the abscissae.

when true samples are desirable, e.g. in the sequential Monte Carlo inference for coalescent clustering [8].

The major drawback of ARS is that it can only be applied to so-called log-concave distributions—distributions where the logarithm of the density function is concave. This is because ARS constructs and uses proposal distributions whose log densities are piecewise linear upper bounds on the log density of the given distribution of interest. [9] and [10] generalized ARS to T -concave distributions—where the density transformed by a monotonically increasing function T is concave.

In this paper we propose a different generalization of ARS to distributions whose log densities can be expressed as a sum of concave and convex functions. These form a very large class of distributions—as we shall see, almost all densities of interest have decompositions into log concave and log convex functions, and include multimodal densities as well. The only requirements we need are that the densities are differentiable with derivatives of bounded variation and tails that decay at least exponentially. We call our generalization concave-convex adaptive rejection sampling (CCARS).

The basic idea of CCARS, described in Section 3, is to upper bound both the concave and convex components using piecewise linear functions. These upper bounds are used to construct a piecewise exponential proposal distribution for rejection sampling. The method for upper bounding the concave and convex components can be applied to obtain lower bounds as well. Whenever the function is evaluated at a sample, the information is used to refine and tighten the bounds at that point. This ensures higher acceptance probabilities in future proposals. In Section 4 we exploit both bounds to approximate the true density function in an adaptive and efficient manner.

In Section 5 we present experimental results on generating samples from several different probability distributions. In Section 6 we discuss using CCARS

to efficiently construct accurate proposal distributions for sequential Monte Carlo inference in coalescent clustering [8]. We conclude with some discussions on the merits and drawbacks of CCARS in Section 7.

2 Rejection and Adaptive Rejection Sampling

In this section we review rejection sampling and adaptive rejection sampling for completeness' sake. Our description of adaptive rejection sampling will also set the stage for the contributions of this paper in the coming sections.

Rejection Sampling

Suppose we wish to obtain a sample from a distribution with density $p(x)$ on the real line. Rejection sampling is a standard Monte Carlo technique for sampling from $p(x)$. It assumes that we have a proposal distribution with density $q(x)$ from which it is easy to obtain samples, and for which there exists a constant $c > 1$ such that $p(x) < cq(x)$ for all x . Rejection sampling proceeds as follows: obtain a sample $x \sim q(x)$; compute acceptance probability $\alpha = p(x)/cq(x)$; accept x with probability α , otherwise reject and repeat the procedure until some sample is accepted.

The intuition behind rejection sampling is straightforward. Obtaining a sample from $p(x)$ is equivalent to obtaining a sample from a uniform distribution under the curve $p(x)$. We obtain this sample by obtaining a uniform sample from under the curve $cq(x)$, and only accept the sample if it by chance also falls under $p(x)$. We repeat this procedure until we have obtained a sample under $p(x)$. This intuition also shows that the average acceptance probability is $1/c$ thus the expected number of samples required from $q(x)$ is c .

Adaptive Rejection Sampling

When a sample is rejected in rejection sampling the computations performed to obtain the sample are discarded and thus wasted. Adaptive rejection sampling (ARS) [6] addresses this wastage by using the rejected samples to improve the proposal distribution so that future proposals have higher acceptance probability.

ARS assumes that the density $p(x)$ is log concave, that is, $f(x) = \log p(x)$ is a concave function. Since $f(x)$ is concave, it is upper bounded by its tangent lines: $f(x) \leq t_{x_0}(x)$ for all x_0 and x , where $t_{x_0}(x) = f(x_0) + f'(x_0)(x - x_0)$ is the tangent at abscissa x_0 . ARS uses proposal distributions whose log densities are constructed as the minimum of a finite set of tangents:

$$f(x) \leq g_n(x) = \min_{i=1 \dots n} t_{x_i}(x) \quad (1)$$

$$q_n(x) \propto \exp(g_n(x)) \quad (2)$$

where x_1, \dots, x_n are the abscissae of the tangent lines. Since $g_n(x)$ is piecewise linear, $q_n(x)$ is a piecewise exponential distribution that can be efficiently sampled from. Say $x_{n+1} \sim q_n(x)$. If the proposal x_{n+1} is rejected, this implies that

x_{n+1} is likely to be located in a part of the real line where the proposal distribution $q_n(x)$ differs significantly from $p(x)$. Instead of discarding x_{n+1} , we add it to the set of abscissae so that $q_{n+1}(x)$ will more closely match $p(x)$ around x_{n+1} .

In order for $q_n(x)$ to be normalizable it is important that $g_n(x) \rightarrow -\infty$ when $x \rightarrow \infty$ and when $x \rightarrow -\infty$. This can be guaranteed if the initial set of abscissae includes a point x_1 for which $f'(x) > 0$ for all $x < x_1$, and a point x_2 for which $f'(x) < 0$ for all $x > x_2$. These two points can usually be easily found and ensure that the tails of $p(x)$ are bounded by the tails of $q_n(x)$ which are in turn exponentially decaying.

[6] proposed two improvements to the above scheme. Firstly, there is an alternative upper bound that is looser but does not require evaluations of the derivatives $f'(x)$. Secondly, a lower bound on $f(x)$ can be constructed based on the secant lines subtended by consecutive abscissae. This is useful in accepting proposed samples without the need to evaluate $f(x)$ each time. Both improvements are useful when $f(x)$ and $f'(x)$ are expensive to evaluate. In the next section we make use of such secant lines for a different purpose: to upper bound the log convex components in a concave-convex decomposition of the log density.

3 Concave Convex Adaptive Rejection Sampling

In this section we propose a generalization to ARS where the log density $f(x) = f_{\cap}(x) + f_{\cup}(x)$ can be decomposed into concave $f_{\cap}(x)$ and convex $f_{\cup}(x)$ functions. As we will see, such decompositions are natural in many situations and many densities of interest can be decomposed in this way¹. The approach we take is to upper bound $f_{\cap}(x)$ and $f_{\cup}(x)$ separately using piecewise linear upper bounds, so that the sum of the upper bounds is itself piecewise linear and an upper bound of $f(x)$. For simplicity we start with the case where the support of the density is a finite closed interval $[a, b]$, and discuss changes needed for the open interval case in Section 3.1. In the following we shall describe our upper bounds in more detail; see Figure 2 for a pictorial depiction of the algorithm.

As in ARS, the upper bound on the concave $f_{\cap}(x)$ is formed by a series of tangent lines at a set of n abscissae, say ordered $a = x_0 < x_1 \dots < x_n = b$. At each abscissa x_i we form the tangent line

$$t_{x_i}(x) = f_{\cap}(x_i) + f'_{\cap}(x_i)(x - x_i),$$

and the upper bound on $f_{\cap}(x)$ is:

$$f_{\cap}(x) \leq g_{\cap}(x) = \min_{i=0 \dots n} t_{x_i}(x) \quad (3)$$

Consecutive tangent lines $t_{x_i}, t_{x_{i+1}}$ intersect at a point $y_i \in (x_i, x_{i+1})$:

$$y_i = \frac{f_{\cap}(x_{i+1}) - f'_{\cap}(x_{i+1})x_{i+1} - f_{\cap}(x_i) + f'_{\cap}(x_i)x_i}{f_{\cap}(x_i) - f_{\cap}(x_{i+1})}$$

¹ Note that such decompositions are not unique; see Section 5.1.

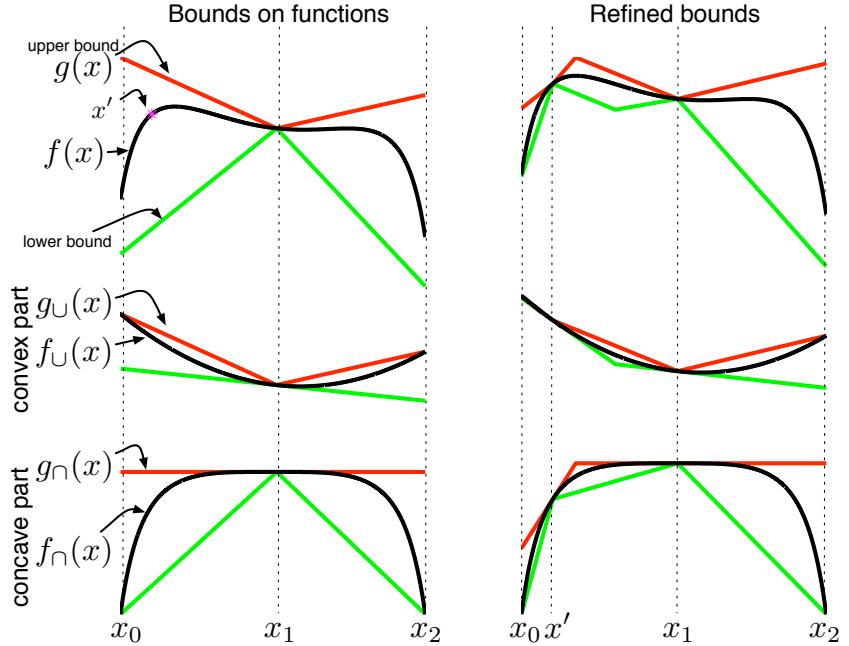


Fig. 2. Concave-convex adaptive rejection sampling. First column: upper and lower bounds on functions $f(x)$, $f_{\cup}(x)$ and $f_{\cap}(x)$. Second column: refined bounds after proposed point x' is rejected.

and $g_{\cap}(x)$ is piecewise linear with the y_i 's forming the change points.

On the other hand, the upper bound on the convex $f_{\cup}(x)$ is formed by a series of n secant lines subtended at the same set of points $x_0 \dots x_n$. For each consecutive pair $x_i < x_{i+1}$ the secant line

$$s_{x_i x_{i+1}}(x) = \frac{f_{\cup}(x_{i+1}) - f_{\cup}(x_i)}{x_{i+1} - x_i}(x - x_i) + f_{\cup}(x_i)$$

is an upper bound on $f_{\cup}(x)$ on the interval $[x_i, x_{i+1}]$, and the upper bound on $f_{\cup}(x)$ is:

$$f_{\cup}(x) \leq g_{\cup}(x) = \max_{i=0 \dots n-1} s_{x_i x_{i+1}}(x) \quad (4)$$

Finally the upper bound on $f(x)$ is just the sum of both upper bounds:

$$f(x) \leq g(x) = g_{\cap}(x) + g_{\cup}(x) \quad (5)$$

Note that $g(x)$ is a piecewise linear function as well, with $2n$ segments. The proposal distribution is then a piecewise exponential distribution with $2n$ segments:

$$q(x) \propto \exp(g(x)) \quad (6)$$

Algorithm 1 Concave-Convex Adaptive Rejection Sampling

```

inputs: functions  $f_{\cap}$ ,  $f_{\cup}$ , domain  $(a, b)$ , numsamples
initialize: abscissae
if  $a = -\infty$  then {bound the left tail}
    search for a point  $x_0$  on the left tail of  $f_{\cap} + f_{\cup}$ , add  $x_0$  as left abscissa.
else
    add  $a$  as the left abscissa.
end if
if  $b = \infty$  then {bound the right tail}
    search for a point  $x_1$  on the right tail of  $f_{\cap} + f_{\cup}$ , add  $x_1$  as right abscissa.
else
    add  $b$  as the right abscissa.
end if
initialize: bounds  $g_{\cap}$  and  $g_{\cup}$ , numaccept = 0.
while numaccept < numsamples do {generate samples}
    sample  $x' \sim q(x) \propto \exp(g_{\cap}(x) + g_{\cup}(x))$ .
    sample  $u \sim \text{Uniform}[0, 1]$ .
    if  $u < \exp(g_{\cap}(x') + g_{\cup}(x') - f_{\cap}(x') - f_{\cup}(x'))$  then
        accept the sample  $x'$ .
        numaccept := numaccept + 1.
    else
        reject sample  $x'$ .
        include  $x'$  in the set of abscissae.
        update the bounds.
    end if
end while

```

Pseudocode for the overall concave-convex ARS (CCARS) algorithm is given in Algorithm 1. At each iteration a sample $x' \sim q(x)$ is drawn from the proposal distribution and accepted with probability $\exp(g(x) - f(x))$. If rejected, x' is added to the list of points to refine the proposal distribution, and the algorithm is repeated. The data structure maintained by CCARS consists of the $n + 1$ abscissae, the n intersections of consecutive tangent lines, and the values of g_{\cap} , g_{\cup} and g evaluated at these $2n + 1$ points.

3.1 Unbounded Domains

Let $p(x) = \exp(f(x))$ be a well-behaved density function over an open domain (a, b) where a and b can be finite or infinite. In this section we consider the behaviour of $f(x)$ near its boundaries and how this may affect our CCARS algorithm.

Consider the behaviour of $f(x)$ as $x \rightarrow a$ (the behaviour near b is symmetrically argued). If $f(x) \rightarrow f(a)$ for a finite $f(a)$, then $f(x)$ is continuous at a and we can construct piecewise linear upper bounds for $f(x)$ such that the corresponding proposal density is normalizable. If $f(x) \rightarrow \infty$ as $x \rightarrow a$ (and in particular a must be finite for $p(x)$ to be properly normalizable), then no piecewise linear upper bound for $f(x)$ exists. On the other hand, if $f(x) \rightarrow -\infty$

then piecewise linear upper bounds for $f(x)$ can be constructed, but such upper bounds consisting of a finite number of segments with normalizable proposal densities exist only if $f(x)$ is log concave near a .

Thus for CCARS to work for $f(x)$ on domain (a, b) we require one of the following situations for its behaviours near a and near b (in the following we consider only case of a ; the b case is similar): either $f(x)$ is continuous at a finite a , or $f(x) \rightarrow -\infty$ as $x \rightarrow a$ and $f(x)$ is log concave on an open interval (a, c) . In case a is finite and $f(x)$ is continuous at a , we simply initialize CCARS with a as an abscissa. Otherwise, we say $f(x)$ has a *log concave tail* at a , and use c as an initial abscissa. Further, the piecewise linear upper bounds of vanilla adaptive rejection sampling can be applied on (a, c) , while CCARS can be applied to the right of c .

3.2 Lower Bounds

Just as in [6] we can construct a lower bound for $f(x)$ so that it need not be evaluated every time a proposed point is to be considered for acceptance. This lower bound can be constructed by reversing the operations on the concave and convex functions: we lower bound $f_{\cap}(x)$ using its secant lines, and lower bound f_{\cup} using its tangent lines. This reversal is perfectly symmetrical and the same code can be reused.

3.3 Concave-Convex Decomposition

The concave-convex adaptive rejection sampling algorithm is most naturally applied when the log density $f(x) = \log p(x)$ can be naturally decomposed into a sum of concave and convex parts, as seen in our examples in Section 5. However it is interesting to observe that many densities of interest can be decomposed in this fashion.

Specifically, suppose that $f(x)$ is differentiable with derivative $f'(x)$ of bounded variation on $[a, b]$. The Jordan decomposition for functions of bounded variations [11] shows that $f'(x) = h_{\cap}(x) + h_{\cup}(x)$ where h_{\cup} is monotonically increasing and h_{\cap} is monotonically decreasing. Integrating, we get $f(x) = f(a) + \int_a^x h_{\cap}(x) + h_{\cup}(x)dx = f(a) + g_{\cap}(x) + g_{\cup}(x)$ where $g_{\cap}(x) = \int_a^x h_{\cap}(x)dx$ is concave, and $g_{\cup}(x) = \int_a^x h_{\cup}(x)dx$ is convex.

Another important issue of such concave-convex decompositions is that they are not unique—adding a convex function to $g_{\cup}(x)$ and subtracting the same function from $g_{\cap}(x)$ preserves convexity and concavity respectively, but can alter the effectiveness of CCARS, as seen in Section 5. We suggest using the “minimal” concave-convex decomposition—one where both are as close to linear as possible.

4 Approximation of Integrals

The sampling method described in the previous section uses piecewise exponential functions for bounding the density function. The upper bound is used as

Algorithm 2 Concave Convex Integral Approximation

```

inputs:  $f_{\cap}$ ,  $f_{\cup}$ , domain  $(a, b)$ , threshold
initialize: abscissae as in Algorithm 1
initialize: upper and lower bounds  $g_{\cap}$ ,  $g_{\cup}$ ,  $l_{\cap}$  and  $l_{\cup}$ 
initialize: calculate the areas under the bounds in each segment,  $\{A_i^g, A_i^l\}$ 
while  $(\sum_i A_i^l)/(\sum_i A_i^g) <$  threshold do {refine bounds}
     $i = \text{argmax}_{i=1,\dots,n} A_i^g - A_i^l$ 
    if  $i$  = a log concave tail segment then
        sample  $x' \sim q(x) \propto \exp(g_{\cap}(x) + g_{\cup}(x))$ 
    else
         $x' = \text{argmax}_{x \in \{z_i^u, z_i^l\}} g_i(x) - l_i(x)$ 
    end if
    include  $x'$  in the set of abscissae.
    update the bounds.
end while

```

the sampling function and the lower bound is used to avoid the expensive function evaluation when possible. What seem to be a byproduct of the sampling algorithm can be used for evaluating the area (or normalizing constant) of the density function. Generally speaking, the adaptive bounds can be used for evaluating the definite integral of any positive function satisfying the conditions of Section 3.3 that can be efficiently represented as a concave convex decomposition (modulo tail behaviour issues in unbounded case).

The area under the upper (lower) bounding piecewise exponential function gives an upper (lower) bound on the area under the unnormalized function $\exp\{f(x)\}$. A measure of the approximation error is the ratio of the areas under the upper and lower bounds. This measure is of interest in case of CCARS as it is the probability that we need to evaluate $f(x)$ when considering a sample for acceptance. Some changes to CCARS make it more efficient for integral approximation, which is discussed in detail below. We call the resulting algorithm concave-convex integral approximation (CCIA).

Note that Algorithm 1 described in the previous section is optimized for requiring as few function evaluations as possible for generating samples from the distribution, therefore ideally it would sample points that have high probability of being accepted. The bounds are updated only if a sampled point is not accepted at the squeezing step, that is, when the acceptance test requires evaluating the function. For integral evaluation, this view is reversed. Since the goal is to fit the function as fast as possible, sampling points with high acceptance probability would waste computation time. As the bound should be adjusted where it is not tight, a better strategy would be to sample those points where there is a large mismatch between the upper and the lower bounds. Therefore, instead of sampling from the upper bound, we can sample from the area between the bounds. Since the bounds are piecewise exponential, this means sampling from the difference of two exponentials. In fact, since we are only interested in op-

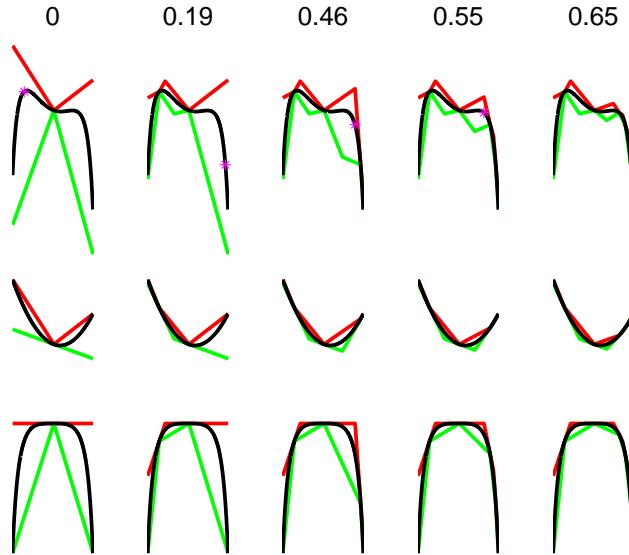


Fig. 3. Evolution of integral approximation bounds on the overall function $f(x)$ (top), the concave part $f_{\cap}(x)$ (middle) and the convex part $f_{\cup}(x)$ (bottom). The segment with the largest area between the bounds is selected deterministically. If one of the end segments is chosen, the new abscissa is sampled from the upper bound, otherwise the point is chosen to be one of the change points. The numbers above the plot show the lower and upper bound area ratio.

timally placing the abscissae rather than generating random samples, sampling can be avoided altogether if we keep the bound structure in mind.

Both upper and the lower bounds touch the function at the same set of abscissae, as seen in Figure 2. Between each pair of consecutive abscissae, two tangents intersect, possibly at different x values for the upper and the lower bound. It is optimal to add to the set of abscissae one of these intersection points for which the bounds are furthest apart.

CCIA, summarized in Algorithm 2, starts similarly to CCARS by initializing the abscissae and the upper and lower bounds $g(x)$, $l(x)$, and calculating the area under both bounds. At each iteration, we find the consecutive pair of abscissae with maximum discrepancy between $g(x)$ and $l(x)$ and add the intersection point with largest discrepancy to the set of abscissae.

The evolution of the bounds over iterations for a bounded function is depicted in Figure 3. The ratio of the upper and lower bounds on the areas are reported above the plots. Initially with one abscissa, the bounds are so loose that the ratio is practically zero. However the bounds get tight reasonably quickly. In

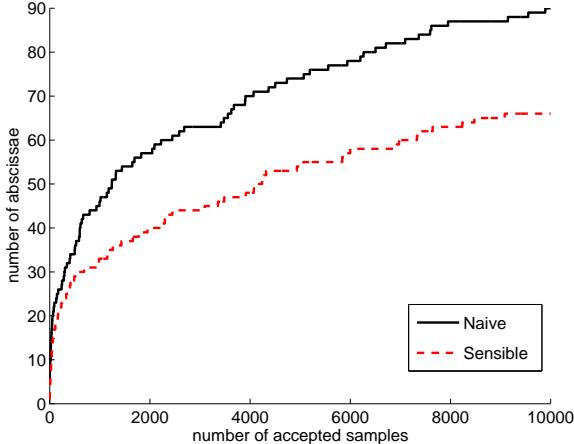


Fig. 4. Demonstration of the difference of a naive versus a sensible function decomposition. The (sensible) dotted curve shows the number of abscissae used as a function of generated samples. The same concave function (a polynomial) was added and subtracted to the concave and convex functions to preserve the original function to produce the (naive) solid curve. The naive decomposition requires much more function evaluations for generating the same number of samples.

the next section, we present experiments on CCIA on densities with unbounded domains.

5 Experiments

As described in the previous sections, adaptively bounding the concave convex function decomposition provides an easy and efficient way to generate independent samples from arbitrary distributions, and to evaluate their normalizing constants. In the following, we present experiments to give an insight about the performance of the algorithms. We start with demonstrating the effect of careless function decomposition on the computational cost. We then apply the algorithms for sampling from some standard but non-log-concave density functions and evaluating their normalization constants.

5.1 Function Decomposition

One important point to keep in mind is that the concave-convex function decomposition is not unique, as discussed in Section 3. Adding and subtracting the same concave function to both $f_{\cap}(x)$ and $f_{\cup}(x)$ preserves the function $f(x)$ and the method is still valid. However, redundancy in the formulation of f_{\cap} and f_{\cup} reduce the efficiency of the method, as demonstrated in Figure 4. Although

the same function is being sampled from, the naive decomposition utilizes many more abscissae.

5.2 Random Number Generation

In this section, we demonstrate the methods on the generalized inverse Gaussian (GIG) distribution and Makeham's distribution, for which there is no standard specialized method of sampling. For both distributions, the log densities are concave for some parameter settings and are naturally expressed as a sum of concave and convex terms otherwise. Since our algorithm reduces to standard ARS for log-concave density functions, we can efficiently sample from these distributions using CCARS for all parameter values.

The *generalized inverse Gaussian* (GIG) distribution is ubiquitously used across many statistical domains, especially in financial data analysis and geostatistics. It is an example of an infinitely divisible distribution and this property allows the construction of nonparametric Bayesian models based on it. The GIG density function is

$$p(x) = \frac{(a/b)^{\lambda/2}}{2K_\lambda(\sqrt{ab})} x^{\lambda-1} \exp\left\{-\frac{1}{2}(ax + bx^{-1})\right\},$$

where $K_\lambda(\cdot)$ is the modified Bessel function of the third kind, $a, b > 0$ and $x > 0$. Sampling from this distribution is not trivial, the most commonly used method being that of [12]. The unnormalized log density is

$$f(x) = (\lambda - 1) \log(x) - \frac{1}{2}(ax + bx^{-1}),$$

which is log-concave for $\lambda > 1$, therefore ARS can be used to sample from it. However, when $\lambda < 1$, the log density is a sum of concave and convex terms. Thus, the function decomposition necessary for CCARS falls out easily;

$$f_{\cap}(x) = -\frac{1}{2}(ax + bx^{-1}), \quad f_{\cup}(x) = (\lambda - 1) \log(x)$$

for $\lambda < 1$.

The second distribution we consider is *Makeham's distribution*, which is used as a representation of the mortality process at adult ages. The density is

$$p(x) = (a + bc^x) \exp\left\{-ax - \frac{b}{\ln(c)}(c^x - 1)\right\}$$

where $b > 0$, $c > 1$, $a > -b$, $x \geq 0$. No specialized method for efficiently sampling from this distribution exists. Similar to the GIG distribution, this function is log-concave for some parameter settings, but not all. Specifically, the density is log-concave for $a < 0$. However for $a > 0$, the log of the first term is convex and the last term is concave which makes it hard for standard algorithms to deal with this distribution. Since it is a sum of a concave and a convex term, the log density is indeed of the form that CCARS can easily deal with:

$$f_{\cap}(x) = -ax - \frac{b}{\ln(c)}(c^x - 1), \quad f_{\cup}(x) = \log(a + bc^x).$$

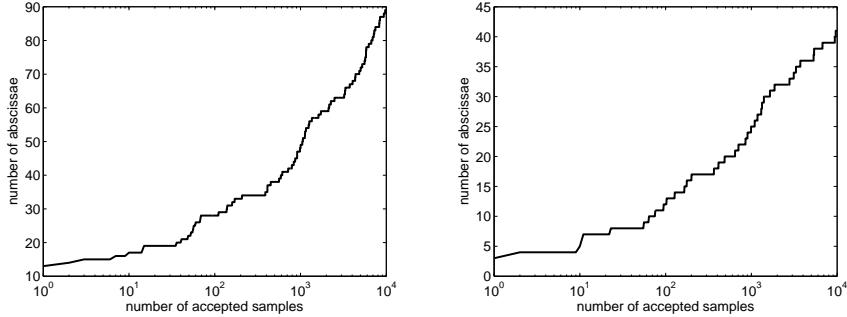


Fig. 5. Generating samples: The change in the number of abscissae while generating several samples for the GIG distribution (left) and Makeham’s distribution (right). The number of abscissae increases slowly.

Generating many samples We assume that evaluating the function $f(x)$ is generally expensive. Therefore, the number of function evaluations gives a measure of the speed of the algorithm. For both CCARS and CCIA, an abscissa is added to the bound every time the functions $f_{\cap}(x)$ and $f_{\cup}(x)$ are evaluated. There is also an overhead of two for checking domain boundaries so that the number of function evaluations will be two plus the number of abscissae. Therefore, to give an intuition about the efficiency of the methods, we report the change of number of abscissae. See Figure 5.

Efficiency for single sample generation As demonstrated by Figure 5, the algorithm efficiently generates multiple independent samples. Generally when the method is used within Gibbs sampling, one only needs a single sample from the conditional distribution at each Gibbs iteration. Therefore it is important to assess the cost of generating a single sample. We did several runs to generate a single sample from the distributions to test the efficiency. The average number of abscissae used in 1000 runs for GIG with $\lambda = -1$ was 7.7, which can also be inferred from Figure 6(a), noting that the bounds get somewhat closer after 7 abscissae. As the convex part gets more dominant with decreasing λ , the number of abscissae necessary to have a good representation of the density increases. See Table 1 for the average number of abscissae for a list of λ values. For all runs, the abscissae were initialized randomly. Note that usually the conditional distributions do not change drastically within a few Gibbs iterations. Therefore the abscissae information of the previous run can be used to have a sensible initialization, decreasing the cost.

Integral estimation The algorithms for refining the bounds when approximating integrals and generating samples is slightly different; the algorithms differ in the manner that they choose a point to add to the bound structure. Figure 6

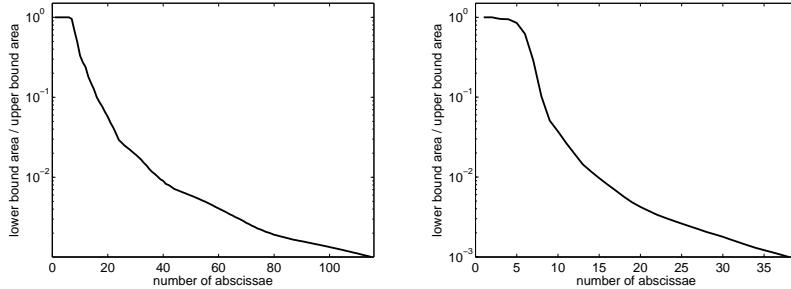


Fig. 6. Approximating integrals: The change in the integral evaluation accuracy as a function of the number of abscissae for the GIG distribution (left) and Makeham’s distribution (right). Integral estimates get more accurate as more abscissae are added.

shows the performance of the algorithm for approximating the area under the unnormalized density (the normalization constants) for GIG and Makeham’s distributions. We see that a reasonable number of abscissae are needed to estimate the normalization constants accurately. Since the concavity of the distributions depends on the parameter settings, we repeated the experiments for several different values and obtained similar curves.

6 Sequential Monte Carlo on Coalescent Clustering

Another application of CCARS, and in fact the application that motivated this line of research, is in the sequential Monte Carlo (SMC) inference algorithm for coalescent clustering [8]. In this section we shall give a brief account of how CCARS can be applied in the multinomial vector coalescent clustering of [8].

Coalescent clustering is a Bayesian model for hierarchical clustering, where a set of n data items (objects) is modelled by a latent binary tree describing the hierarchical relationships among the objects, with objects closer together on the tree being more similar to each other. An example of such a problem is in population genetics, where objects are haplotypic DNA sequences from a

Table 1. Change in the number of abscissae for different parameter values for GIG.

λ	1.5	1.1	1	0.99	0.9	0.5	0	-0.5	-1
AVG	3.1(.6)	3.0(.6)	3.0(.6)	4.1(.8)	4.7(.8)	5.6(1)	6.5(1)	7.1(1.2)	7.7(1.2)
MIN	2	2	2	2	3	3	3	4	4
MAX	6	5	5	7	7	9	10	11	13
MEDIAN	3	3	3	4	5	6	6	7	8

number of individuals, and the latent tree describes the genealogical history of these individuals.

The inferential problem in coalescent clustering is in estimating the latent tree structure given observations of the objects. The SMC inference algorithm proposed by [8] operates as follows: starting with n data items each in its own (trivial) subtree, each iteration of the SMC algorithm proposes a pair of subtrees to merge (coalesce) as well as a time in the past at which they coalesce. The algorithm stops after $n - 1$ iterations when all objects have been coalesced into one tree. Being a SMC algorithm, multiple such runs (particles) are used, and resampling steps are taken to ensure that the particles are representative of the posterior distribution.

At iteration i , the optimal SMC proposal distribution is,

$$p(t, l, r | \theta_{i-1}) \propto e^{-(\frac{n-i+1}{2})(t-t_{i-1})} \prod_d \left\{ 1 - e^{\lambda_d(2t-t_l-t_r)} (1 - \sum_k q_{dk} M_{ldk} M_{rdk}) \right\}$$

where the proposal is for subtrees l and r to be merged at time $t < t_{i-1}$, θ_{i-1} stores the subtrees formed up to iteration $i - 1$, t_{i-1} is the time of the last coalescent event, t_l, t_r are the coalescent times at which l and r are themselves formed, d indexes the entries of the multinomial vector, k indexes the values each entry can take on, λ_d and q_{dk} are parameters of the mutation process, and M_{ldk}, M_{rdk} are messages representing likelihoods of the data under subtrees l and r respectively.

It can be shown that $L_{dlr} = 1 - \sum_k q_{dk} M_{ldk} M_{rdk}$ ranges from -1 to 1 , and the term in curly braces is log convex in t if $L_{dlr} < 0$ and log concave if $L_{dlr} > 0$. Thus the SMC proposal density has a natural log concave-convex decomposition and CCARS can be used to efficiently obtain draws from $p(t, l, r | \theta_{i-1})$. In fact, what is actually done is that CCIA is used to form a tight upper bound on $p(t, l, r | \theta_{i-1})$, which is used as the SMC proposal instead. This is because the area under the upper bound can be computed efficiently, but not the area under $p(t, l, r | \theta_{i-1})$, this area being required to reweigh the particles appropriately.

7 Discussion

We have proposed a generalization of adaptive rejection sampling to the case where the log density can be expressed as a sum of concave and convex functions. The generalization is based on the idea that both the concave and the convex functions can be upper bounded by piecewise linear functions, so that the sum of the piecewise linear functions is a piecewise linear upper bound on the log density itself. We have also described a related algorithm for estimating upper and lower bounds on definite integrals of functions. We experimentally verified that our concave-convex adaptive rejection sampling algorithm works on a number of well-known distributions, and is an indispensable component of a recently proposed SMC inference algorithm for coalescent clustering.

The original adaptive rejection sampling idea of [6] has been generalized in a number of different ways by [9] and [10]. These generalizations are orthogonal to

our approach and are in fact complementary—e.g. we can generalize our work to densities which are sums of concave and convex functions after a monotonic transformation by T .

The idea of concave-convex decompositions have also been explored in the approximate inference context by [13]. There the problem is to find a local minimum of a function, and the idea is to upper bound the concave part using a tangent plane at the current point, resulting in a convex upper bound to the function of interest which can be minimized efficiently, producing the next (provably better) point and iterating until convergence. We believe that concave-convex decompositions of functions are natural in other problems as well and exploiting such structure can lead to efficient solutions for such problems.

We have produced software downloadable at <http://www.gatsby.ucl.ac.uk/~dilan/software>, and intend to release it for general usage. We are currently applying CCARS and CCIA to a new SMC inference algorithm for coalescent clustering with improved run-time and performance.

Acknowledgements

We thank the Gatsby Charitable Foundation for funding.

A Sampling from a Piecewise Exponential Distribution

The proposal distribution $q(x) \propto \exp(g(x))$ is piecewise exponential if $g(x)$ is piecewise linear. In this section we describe how to obtain a sample from $q(x)$.

Suppose the change points of $g(x)$ are $z_0 < z_1 < \dots < z_m$, and $g(x)$ has slope m_i in (z_{i-1}, z_i) . the area A_i under each exponential segment of $\exp(g(x))$ is:

$$A_i = \int_{z_{i-1}}^{z_i} \exp(g(x)) dx = (\exp(g(z_i)) - \exp(g(z_{i-1}))) / m_i \quad (7)$$

We obtain a sample x' from $q(x)$ by first sampling a segment i with probability proportional to A_i , then sampling an $x' \in (z_{i-1}, z_i)$ using the inverse cumulative distribution transform, resulting in: sample $u \sim \text{Uniform}[0, 1]$ and set $x' = \frac{1}{m_i} \log(u e^{m_i z_i} + (1-u) e^{m_i z_{i-1}})$.

References

- [1] Spiegelhalter, D.J., Thomas, A., Best, N., Gilks., W.R.: BUGS: Bayesian inference using Gibbs sampling (1999, 2004)
- [2] Winn, J.: Vibes: Variational inference in Bayesian networks (2004)
- [3] Minka, T., Winn, J., Guiver, J., Kannan, A.: Infer.NET (2008)
- [4] Devroye, L.: Non-uniform Random Variate Generation. Springer, New York (1986)
- [5] Neal, R.M.: Slice sampling. *The Annals of Statistics* **31**(3) (2003) 705–767
- [6] Gilks, W.R., Wild, P.: Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* **41** (1992) 337–348

- [7] Gilks, W.R., Best, N.G., Tan, K.K.C.: Adaptive rejection metropolis sampling within Gibbs sampling. *Applied Statistics* **44**(4) (1995) 455–472
- [8] Teh, Y.W., Daumé III, H., Roy, D.M.: Bayesian agglomerative clustering with coalescents. In: Advances in Neural Information Processing Systems. Volume 20. (2008)
- [9] Hoermann, W.: A rejection technique for sampling from T-concave distributions. *ACM Transactions on Mathematical Software* **21**(2) (1995) 182–193
- [10] Evans, M., Swartz, T.: Random variate generation using concavity properties of transformed densities. *Journal of Computational and Graphical Statistics* **7**(4) (1998) 514–528
- [11] Hazewinkel, M., ed.: Encyclopedia of Mathematics. Kluwer Academic Publ. (1998)
- [12] Dagpunar, J.: An easily implemented generalised inverse Gaussian generator. *Communications in Statistics - Simulation and Computation* **18**(2) (1989) 703–710
- [13] Yuille, A.L., Rangarajan, A.: The concave-convex procedure. *Neural Computation* **15**(4) (2003) 915–936