An Efficient Minibatch Acceptance Test for Metropolis-Hastings

Daniel Seita[1], **Xinlei Pan**[1], **Haoyu Chen**[1], **John Canny**[1,2]
[1] University of California, Berkeley, CA
[2] Google Research, Mountain View, CA
{seita,xinleipan,haoyuchen,canny}@berkeley.edu

Abstract

We present a novel Metropolis-Hastings method for large datasets that uses small expected-size minibatches of data. Previous work on reducing the cost of Metropolis-Hastings tests yield variable data consumed per sample, with only constant factor reductions versus using the full dataset for each sample. Here we present a method that can be tuned to provide arbitrarily small batch sizes, by adjusting either proposal step size or temperature. Our test uses the noise-tolerant Barker acceptance test with a novel additive correction variable. The resulting test has similar cost to a normal SGD update. Our experiments demonstrate several order-of-magnitude speedups over previous work.

1 INTRODUCTION

Markov chain Monte Carlo (MCMC) sampling is a powerful method for computation on intractable distributions. We are interested in large dataset applications, where the goal is to sample a posterior distribution $p(\theta|x_1, \ldots, x_N)$ of parameter θ for large N. The Metropolis-Hastings method (M-H) generates sample candidates from a proposal distribution q which is in general different from the target distribution p, and decides whether to accept or reject based on an acceptance test. The acceptance test is usually a Metropolis test [Metropolis et al., 1953, Hastings, 1970].

Many state-of-the-art machine learning methods, and deep learning in particular, are based on minibatch updates (such as SGD) to a model. Minibatch updates produce many improvements to the model for each pass over the dataset, and have high sample efficiency. In contrast, conventional M-H requires calculations over the full dataset to produce a new sample. Recent results from [Korattikara et al., 2014] and [Bardenet et al., 2014] perform approximate (bounded error) acceptance tests using subsets of the full dataset. The amount of data consumed for each test varies significantly from one minibatch to the next. By contrast, [Maclaurin and Adams, 2014, Bardenet et al., 2016] perform exact tests but require a lower bound on the parameter distribution across its domain. The amount of data reduction depends on the accuracy of this bound, and such bounds are only available for relatively simple distributions.

Here we derive a new test which incorporates the variability in minibatch statistics as *a natural part of the test* and requires less data per iteration than prior work. We use a Barker test function [Barker, 1965], which makes our test naturally error tolerant. The idea of using a noise-tolerant Barker's test function was suggested but not explored empirically in [Bardenet et al., 2016] section 6.3. But the asymptotic test statistic CDF and the Barker function are different, which leads to fixed errors for the approach in [Bardenet et al., 2016]. Here, we show that the difference between the distributions can be corrected with an additive random variable. This leads to a test which is fast, and whose error can be made arbitrarily small.

Our test is applicable when the variance (over data samples) of the log acceptance probability is small enough (less than 1). It's not clear at first why this quantity should be bounded, but we will show that it is "natural" for well-specified models running Metropolis-Hastings sampling with optimal proposals [Roberts and Rosenthal, 2001] on a full dataset. When we reduce the amount of data for the test, the variance goes up. We have to reduce variance in one of several ways. Either:

- Increase the temperature K of the target distribution. Log likelihoods scale as $1/K$, and so the variance of the likelihood ratio will vary as $1/K^2$. Our model is no longer well-specified (we are do-

ing inference at a temperature different from that assumed during data generation), but as we demonstrate in Section 6.2, higher temperature can be advantageous for parameter exploration.
- For continuous distributions, reduce the proposal step size and variance compared to an optimal proposal. The variance of the log acceptance probability scales as the square of proposal step size.
- Utilize Hamiltonian and Langevin Dynamics to judiciously model a system so as to generate distant yet high-quality proposals.

It is worth discussing at this point the typical goals of M-H sampling on large datasets. By the Bernstein-von Mises Theorem, the posterior distribution for a Bayesian inference task has variance that scales inversely with N. Simply sampling from it is one application, but an efficient proposal [Roberts and Rosenthal, 2001] has similar variance to the target and will diffuse to it extremely slowly. For applications to neural networks or models where the posterior is multimodal [Choromanska et al., 2015], samplers will likely get trapped in one of the modes. A common solution is to anneal the sampler, running first at high temperatures to flatten the likelihood landscape. This in turn reduces the variance of the log acceptance probability and allows our test to be applied. These samples can cover the search space densely with small steps rather than taking a few sparse steps towards an optimum. In this mode, M-H tests can be used in similar fashion to Stochastic Gradient Descent. The goal in SGD is to make gradual progress to a posterior mode with each step, taking small steps so that the cumulative displacement has progressively lower variance.

The contributions of this paper are as follows:

- We develop a new, more efficient (in samples per test) minibatch acceptance test with quantifiable error bounds. The test uses a novel additive correction variable to implement a Barker test based on minibatch mean and variance.
- We compare performance of our new test and prior approaches on several datasets. We demonstrate several order-of-magnitude improvements in efficiency (measured as data consumed per test), and show that it does not suffer from long-tailed minibatch sizes.

2 PRELIMINARIES

In the Metropolis-Hastings method [Gilks and Spiegelhalter, 1996, Brooks et al., 2011], a difficult-to-compute probability distribution $p(\theta)$ is sampled using a Markov chain $\theta_1, \ldots, \theta_T$. The sample θ_{t+1} at time $t+1$ is generated using a candidate θ' from a (simpler) proposal distribution $q(\theta'|\theta_t)$, filtered by an acceptance test. The acceptance test is usually a Metropolis test. The Metropolis test has acceptance probability:

$$\alpha(\theta_t, \theta') = \frac{p(\theta')q(\theta_t|\theta')}{p(\theta_t)q(\theta'|\theta_t)} \wedge 1 \tag{1}$$

where $a \wedge b$ denotes $\min(a, b)$. With probability $\alpha(\theta_t, \theta')$, we accept θ' and set $\theta_{t+1} = \theta'$, otherwise set $\theta_{t+1} = \theta_t$. The test is often implemented with an auxiliary random variable $u \sim \mathcal{U}(0, 1)$ with a comparison $u < \alpha(\theta_t, \theta')$; here, $\mathcal{U}(a, b)$ denotes the uniform distribution on the interval $[a, b]$. For simplicity, we drop the subscript t for the current sample θ_t and denote it as θ.

The acceptance test guarantees detailed balance, which means $p(\theta)p(\theta'|\theta) = p(\theta')p(\theta|\theta')$, where $p(\theta'|\theta)$ is the probability of a transition from state θ to θ'. Here, $p(\theta'|\theta) = q(\theta'|\theta)\alpha(\theta, \theta')$. This condition, together with ergodicity, guarantees that the Markov chain has a unique stationary distribution $\pi(\theta) = p(\theta)$. For Bayesian inference, the target distribution is $p(\theta|x_1, \ldots, x_N)$. The acceptance probability is now:

$$\alpha(\theta, \theta') = \frac{p_0(\theta') \prod_{i=1}^{N} p(x_i|\theta')q(\theta|\theta')}{p_0(\theta) \prod_{i=1}^{N} p(x_i|\theta)q(\theta'|\theta)} \wedge 1 \tag{2}$$

where $p_0(\theta)$ is the prior. Computing samples this way requires all N data points, but this is very expensive for large datasets.

To address this challenge, [Korattikara et al., 2014, Bardenet et al., 2014] perform approximate Metropolis-Hasting tests using sequential hypothesis testing. Each iteration, they start with a small minibatch and test whether θ' should be accepted based on approximating $u < \alpha(\theta, \theta')$. If the approximate test cannot decide with sufficient confidence, the minibatch grows and the test repeats. This process continues until a decision. The bounds depend on either an asymptotic Central Limit Theorem [Korattikara et al., 2014] or a concentration bound [Bardenet et al., 2014]. We refer to these algorithms, respectively, as AUSTEREMH and MHSUBLHD. While both show useful reductions in the number of samples required, they suffer from two drawbacks: having to compute all log likelihood tokens each iteration to compute certain statistics,[1] (see Section 5), and resolving small log likelihood ratio differences between the minibatch and full batch versions. We discuss a worst-case scenario in Section 2.2.

[1] Thus, these are not technically minibatch methods.

2.1 NOTATION

Following [Bardenet et al., 2014], we write the test $u < \alpha(\theta, \theta')$ equivalently as $\Lambda(\theta, \theta') > \psi(u, \theta, \theta')$, where[2]

$$\begin{aligned} \Lambda(\theta, \theta') &= \sum_{i=1}^{N} \log \frac{p(x_i|\theta')}{p(x_i|\theta)}, \\ \psi(u, \theta, \theta') &= \log \left(u \frac{q(\theta'|\theta) p_0(\theta)}{q(\theta|\theta') p_0(\theta')} \right). \end{aligned} \tag{3}$$

To simplify notation, we assume that temperature $K = 1$ (saving T to indicate the number of samples to draw). Temperature appears as an exponential on each likelihood, $p(x_i|\theta)^{1/K}$, so the effect would be to act as a $1/K$ factor on $\Lambda(\theta, \theta')$.

To reduce computational effort, an unbiased estimate of $\Lambda(\theta, \theta')$ based on a minibatch $\{x_1^*, \ldots, x_b^*\}$ can be used:

$$\Lambda^*(\theta, \theta') = \frac{N}{b} \sum_{i=1}^{b} \log \frac{p(x_i^*|\theta')}{p(x_i^*|\theta)}. \tag{4}$$

Finally, it will be convenient for our analysis to define $\Lambda_i(\theta, \theta') = N \log(\frac{p(x_i|\theta')}{p(x_i|\theta)})$. Thus, $\Lambda(\theta, \theta')$ is the mean of $\Lambda_i(\theta, \theta')$ over the entire dataset, and $\Lambda^*(\theta, \theta')$ is the mean of the $\Lambda_i(\theta, \theta')$ in its minibatch.

Since minibatches contains randomly selected samples, the values Λ_i are i.i.d. random variables.[3] By the Central Limit Theorem, we expect $\Lambda^*(\theta, \theta')$ to be approximately Gaussian. The acceptance test then becomes a statistical test of the hypothesis that $\Lambda(\theta, \theta') > \psi(u, \theta, \theta')$ by establishing that $\Lambda^*(\theta, \theta')$ is substantially larger than $\psi(u, \theta, \theta')$.

2.2 A WORST-CASE GAUSSIAN EXAMPLE

Let $x_1, \ldots, x_N$ be i.i.d. $\mathcal{N}(\theta, 1)$ with known variance $\sigma^2 = 1$ and (unknown) mean $\theta = 0.5$. We use a uniform prior on θ. The log likelihood ratio is

$$\Lambda^*(\theta, \theta') = N(\theta' - \theta) \left(\frac{1}{b} \sum_{i=1}^{b} x_i^* - \theta - \frac{\theta' - \theta}{2} \right) \tag{5}$$

which is normally distributed over selection of the Normal samples x_i^*. Since the x_i^* have unit variance, their mean has variance $1/b$, and the variance of $\Lambda^*(\theta, \theta')$ is $\sigma^2(\Lambda^*) = (\theta' - \theta)^2 N^2 / b$. In order to pass a hypothesis test that $\Lambda > \psi$, there needs to be a large enough gap (several $\sigma(\Lambda^*)$) between $\Lambda^*(\theta, \theta')$ and $\psi(u, \theta, \theta')$.

[2]Our definitions differ from those in [Bardenet et al., 2014] by a factor of N to simplify our analysis later.

[3]The analysis assumes sampling with replacement although implementations on typical large datasets will approximate this by sampling without replacement.

The posterior is a Gaussian centered on the sample mean μ, and with variance $1/N$ (i.e., $\mathcal{N}(\mu, 1/N)$). In one dimension, an efficient proposal distribution has the same variance as the target distribution [Roberts and Rosenthal, 2001], so we use a proposal based on $\mathcal{N}(\theta, 1/N)$. It is symmetric $q(\theta'|\theta) = q(\theta|\theta')$, and since we assumed a uniform prior, $\psi(u, \theta, \theta') = \log u$. Our worst-case scenario is specified in Lemma 1.

Lemma 1. *For the model in Section 2.2, there exists a fixed (independent of N) constant c such that with probability $\geq c$ over the joint distribution of (θ, θ', u),* AUSTEREMH *and* MHSUBLHD *consume all N samples.*

Proof. See Appendix, Section A.1. □

Similar results can be shown for other distributions and proposals by identifying regions in product space (θ, θ', u) such that the hypothesis test needs to separate nearly-equal values. It follows that the accelerated tests from prior work require at least a constant fraction $\geq c$ in the amount of data consumed per test compared to full-data tests, so their speed-up is $\leq 1/c$. The issue is the use of tail bounds to separate $\Lambda - \psi$ from zero; for certain input/random u combinations, this difference can be arbitrarily close to zero. We avoid this by using the *approximately normal* variation in Λ^* to *replace* the variation due to u.

2.3 MCMC POSTERIOR INFERENCE

There is a separate line of MCMC work drawing principles from statistical physics. One can apply Hamiltonian Monte Carlo (HMC) [Neal, 2010] methods which generate high acceptance *and* distant proposals when run on full batches of data. Recently Langevin Dynamics [Welling and Teh, 2011, Ahn et al., 2012] has been applied to Bayesian estimation on minibatches of data. This simplified dynamics uses local proposals and avoids M-H tests by using small proposal steps whose acceptance approaches 1 in the limit. However, the constraint on proposal step size is severe, and the state space exploration reduces to a random walk. Full minibatch HMC for minibatches was described in [Chen et al., 2014] which allows momentum-augmented proposals with larger step sizes. However, step sizes are still limited by the need to run accurately without M-H tests. By providing an M-H test with similar cost to standard gradient steps, our work opens the door to applying those methods with much more aggressive step sizes without loss of accuracy.

3 A NEW MH ACCEPTANCE TEST

3.1 LOG-LIKELIHOOD RATIOS

For our new M-H test, we denote the exact and approximate log likelihood ratios as Δ and Δ^*, respectively. First, Δ is defined as

$$\Delta(\theta, \theta') = \log \frac{p_0(\theta') \prod_{i=1}^{N} p(x_i|\theta') q(\theta|\theta')}{p_0(\theta) \prod_{i=1}^{N} p(x_i|\theta) q(\theta'|\theta)}, \quad (6)$$

where p_0, p, and q match the corresponding functions within Equation (2). We separate out terms dependent and independent of the data as:

$$\Delta(\theta, \theta') = \underbrace{\sum_{i=1}^{N} \log \frac{p(x_i|\theta')}{p(x_i|\theta)}}_{\Lambda(\theta,\theta')} - \psi(1, \theta, \theta'). \quad (7)$$

A minibatch estimator of Δ, denoted as Δ^*, is

$$\Delta^*(\theta, \theta') = \underbrace{\frac{N}{b} \sum_{i=1}^{b} \log \frac{p(x_i^*|\theta')}{p(x_i^*|\theta)}}_{\Lambda^*(\theta,\theta')} - \psi(1, \theta, \theta'). \quad (8)$$

Note that Δ and Δ^* are evaluated on the full dataset and a minibatch of size b respectively. The term N/b means $\Delta^*(\theta, \theta')$ is an unbiased estimator of $\Delta(\theta, \theta')$.

The key to our test is a smooth acceptance function. We consider functions other than the classical Metropolis test that satisfy the detailed balance condition needed for accurate posterior estimation. A class of suitable functions is specified as follows:

Lemma 2. *If $g(s)$ is any function such that $g(s) = \exp(s)g(-s)$, then the acceptance function $\alpha(\theta, \theta') \triangleq g(\Delta(\theta, \theta'))$ satisfies detailed balance.*

This result is used in [Barker, 1965] to define the Barker acceptance test.

3.2 BARKER (LOGISTIC) ACCEPTANCE FUNCTION

For our new MH test we use the Barker logistic [Barker, 1965] function: $g(s) = (1 + \exp(-s))^{-1}$. Straightforward arithmetic shows that it satisfies the condition in Lemma 2. It is slightly less efficient than the Metropolis test, since its acceptance rate for vanishing likelihood difference is 0.5. However we will see that its overall sample efficiency is much higher than the earlier methods.

Assume we begin with the current sample θ and a candidate sample θ', and that $V \sim \mathcal{U}(0, 1)$ is a uniform random variable. We accept θ' if $g(\Delta(\theta, \theta')) > V$, and reject otherwise. Since $g(s)$ is monotonically increasing, its inverse $g^{-1}(s)$ is well-defined and unique. So an equivalent test is to accept θ' iff

$$\Delta(\theta, \theta') > X = g^{-1}(V) \quad (9)$$

where X is a random variable with the logistic distribution (its CDF is the logistic function). To see this notice that $\frac{dV}{dX} = g'$, that g' is the density corresponding to a logistic CDF, and finally that $\frac{dV}{dX}$ is the density of X. The density of X is symmetric, so we can equivalently test whether

$$\Delta(\theta, \theta') + X > 0 \quad (10)$$

for a logistic random variable X.

3.3 A MINIBATCH ACCEPTANCE TEST

We now describe acceptance testing using the minibatch estimator $\Delta^*(\theta, \theta')$. From Equation (8), $\Delta^*(\theta, \theta')$ can be represented as a constant term plus the mean of b IID terms $\Lambda_i(\theta, \theta')$ of the form $N \log \frac{p(x_i^*|\theta')}{p(x_i^*|\theta)}$. As b increases, $\Delta^*(\theta, \theta')$ therefore has a distribution which approaches a normal distribution by the Central Limit Theorem. We now describe this using an asymptotic argument and defer specific bounds between the CDFs of $\Delta^*(\theta, \theta')$ and a Gaussian to Section 5.

In the limit, since Δ^* is normally distributed about its mean Δ, we can write

$$\Delta^* = \Delta + X_{\text{norm}}, \quad X_{\text{norm}} \sim \bar{\mathcal{N}}(0, \sigma^2(\Delta^*)), \quad (11)$$

where $\bar{\mathcal{N}}(0, \sigma^2(\Delta^*))$ denotes a distribution which is approximately normal with variance $\sigma^2(\Delta^*)$. But to perform the test in Equation (10) we want $\Delta + X$ for a logistic random variable X (call it $X_{\log}$ from now on). In [Bardenet et al., 2016] it was proposed to use Δ^* in a Barker test, and tolerate the fixed error between the logistic and normal distributions.

Our approach is to instead decompose $X_{\log}$ as

$$X_{\log} = X_{\text{norm}} + X_{\text{corr}}, \quad (12)$$

where we assume $X_{\text{norm}} \sim \mathcal{N}(0, \sigma^2)$ and that X_{corr} is a zero-mean "correction" variable with density $C_\sigma(X)$. The two variables are added (i.e., their distributions convolve) to form $X_{\log}$. This decomposition requires an appropriate C_σ, which we derive in Section 4. Using X_{corr} samples from $C_\sigma(X)$, the acceptance test is now

$$\Delta + X_{\log} = (\Delta + X_{\text{norm}}) + X_{\text{corr}} = \Delta^* + X_{\text{corr}} > 0. \quad (13)$$

Therefore, assuming the variance of Δ^* is small enough, if we have an estimate of Δ^* from the current data minibatch, we test acceptance by adding a random variable

X_{corr} and then accept θ' if the result is positive (and reject otherwise).

If $\bar{\mathcal{N}}(0, \sigma^2(\Delta^*))$ is exactly $\mathcal{N}(0, \sigma^2(\Delta^*))$, the above test is exact, and as we show in Section 5, if there is a maximum error ϵ between the CDF of $\bar{\mathcal{N}}(0, \sigma^2(\Delta^*))$ and the CDF of $\mathcal{N}(0, \sigma^2(\Delta^*))$, then our test has an error of at most ϵ relative to the full batch version.

4 THE CORRECTION DISTRIBUTION

Our test in Equation (13) requires knowing the distribution of X_{corr}. In Section 5, we show that the test accuracy depends on the absolute error between the CDFs of $X_{\text{norm}} + X_{\text{corr}}$ and $X_{\log}$. Consequently, we need to minimize this in our construction of X_{corr}. More formally, let $\Phi_{s_X} = \Phi(X/s_X)$ where Φ is the standard normal CDF[4], $S(X)$ be the logistic function, and $C_\sigma(X)$ be the *density* of the correction X_{corr} distribution. Our goal is to solve:

$$C_\sigma^* = \arg\min_{C_\sigma} |\Phi_\sigma * C_\sigma - S| \tag{14}$$

where $*$ denotes convolution. To compute C_σ, we assume the input Y and another variable X lie in the intervals $[-V, V]$ and $[-2V, 2V]$, respectively. We discretize the convolution by discretizing X and Y into $4N+1$ and $2N+1$ values respectively. If $i \in \{-2N, \ldots, 2N\} = \mathcal{I}$ and $j \in \{-N, \ldots, N\} = \mathcal{J}$, then we can write $X_i = i(V/N)$ and $Y_j = j(V/N)$, and the objective can be written as:

$$C_\sigma^* = \arg\min_{C_\sigma} \max_{i \in \mathcal{I}} \left| \sum_{j \in \mathcal{J}} \Phi_\sigma(X_i - Y_j) C_\sigma(Y_j) - S(X_i) \right|.$$

Now define matrix M and vectors u and v such that $M_{ij} = \Phi_\sigma(X_i - Y_j)$, $u_j = C_\sigma(Y_j)$, and $v_i = S(X_i)$, where the indices i and j are appropriately translated to be non-negative for M, u, and v. The problem is now to minimize $\|Mu - v\|_\infty$ with the density non-negative constraint $u > 0$. We approximate this with least squares:

$$u^* = \arg\min_u \ \|Mu - v\|_2^2 + \lambda \|u\|_2^2, \tag{15}$$

with regularization λ. The solution is well-known from the normal equations ($u^* = (M^T M + \lambda I)^{-1} M^T v$) and in practice yields an acceptable L_∞ norm.

With this approach, there is no guarantee that $u^* \geq 0$. However, we have some flexibility in the choice of σ in Equation (14). As we decrease the variance of X_{norm}, the variance of X_{corr} grows by the same amount and is in fact the result of convolution with a Gaussian whose variance is the difference. Thus as σ decreases, $C_\sigma(X)$ grows and approaches the derivative of a logistic function at $\sigma = 0$. It retains some weak negative values for $\sigma > 0$ but removal of those leads to small error. We use $N = 4000$ and $\lambda = 10$ for our experiments, which empirically provided excellent performance. See Table 3 in Appendix B.1 for detailed L_∞ errors for different settings. Algorithm 1 describes our procedure, MHMINIBATCH. A few points:

[4]Hence, Φ_{s_X} is the CDF of a zero-mean Gaussian with standard deviation s_X.

Algorithm 1 Our acceptance test, MHMINIBATCH.

Input: number of samples T, minibatch size m, error bound δ, pre-computed correction $C_1(X)$ distribution, initial sample θ_1.
Output: a chain of T samples $\{\theta_1, \ldots, \theta_T\}$.
for $t = 1$ **to** T **do**
  -Propose a candidate θ' from proposal $q(\theta'|\theta_t)$.
  -Draw a minibatch of m points $\{x_1^*, \ldots, x_m^*\}$.
  -Compute $\Delta^*(\theta_t, \theta')$ and sample variance $s^2_{\Delta^*}$.
  -Estimate moments $\mathbb{E}[|\Lambda_i - \Lambda|]$ and $\mathbb{E}[|\Lambda_i - \Lambda|^3]$ from the sample, and error ϵ from Corollary 1.
  while $s^2_{\Delta^*} \geq 1$ **or** $\epsilon > \delta$ **do**
    -Draw m more samples to augment the minibatch, update Δ^*, $s^2_{\Delta^*}$ and ϵ estimates.
  end while
  -Draw $X_{\text{nc}} \sim \mathcal{N}(0, 1 - s^2_{\Delta^*})$ and $X_{\text{corr}} \sim C_1(X)$.
  if $\Delta^* + X_{\text{nc}} + X_{\text{corr}} > 0$ **then**
    -Accept the candidate, $\theta_{t+1} = \theta'$.
  else
    -Reject and re-use the old sample, $\theta_{t+1} = \theta_t$.
  end if
end for

- It uses an adaptive step size so as to use the smallest possible average minibatch size. Unlike previous work, the size distribution is short-tailed.
- An additional normal variable X_{nc} is added to Δ^* to produce a variable with unit variance. This is not mathematically necessary, but allows us to use a single correction distribution C_1 with $\sigma = 1$ for X_{corr}, saving on memory footprint.
- The sample variance of Δ^* is denoted as $s^2_{\Delta^*}$ and is proportional to $\|\theta' - \theta\|_2^2$.

5 ANALYSIS

We now derive error bounds for our M-H test and the target distribution it generates. In the most similar prior works, [Korattikara et al., 2014], CLT asymptotic arguments are used to show that their approximate acceptance test error (with respect to full-batch M-H) tends to zero as batch size increases, but no quantitative bounds are

given. In addition, to design a test that minimizes data usage given an error bound, they rely on computing a standardized mean μ_{std} each iteration. As the value of μ_{std} depends on all $\log p(x_i|\theta)$ and $\log p(x_i|\theta')$ terms, the test is technically not a minibatch test; [Korattikara et al., 2014] propose a conservative variant which assumes $\mu_{\text{std}} = 0$ and avoids the need to evaluate all likelihood terms. We refer to this as AUSTEREMH(C) and the default, non-conservative variant as AUSTEREMH(NC).

For MHSUBLHD, the test requires direct bounds on the log likelihood ratio, which for general distributions requires knowing $p(x_i|\theta)$ and $p(x_i|\theta')$ for all N samples. In [Bardenet et al., 2014], explicit bounds are given and depend on

$$C_{\theta,\theta'} = \max_{1\le i\le N} |\log p(x_i|\theta') - \log p(x_i|\theta)|. \quad (16)$$

It is possible to analytically determine $C_{\theta,\theta'}$ for logistic regression and other simple models, but it is difficult to do so for more complicated models, so one generally needs to use all $p(x_i|\theta')$ terms[5] to get $C_{\theta,\theta'}$, like the AUSTEREMH(NC) test. In contrast, we use quantitative forms of the CLT which rely on measurable statistics from a *single* minibatch.

In Section 5.1, we present bounds on the absolute and relative error (in terms of the CDFs) of the distribution of Δ^* versus a Gaussian. We then show in Section 5.2 that these bounds are preserved after the addition of other random variables (e.g., X_{nc} and X_{corr}). It then follows that the acceptance test has the same error bound.

5.1 BOUNDING THE ERROR OF Δ^* FROM A GAUSSIAN

We use the following quantitative central-limit result:

Lemma 3. *Let $X_1, \ldots, X_n$ be a set of zero-mean, independent, identically-distributed random variables with sample mean $\bar{X}$ and sample variance s_X^2 where:*

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad s_X = \frac{1}{n}\left(\sum_{i=1}^{n}(X_i - \bar{X})^2\right)^{\frac{1}{2}}. \quad (17)$$

This means $t = \bar{X}/s_X$ has an approximate Student's distribution which approaches a Gaussian. Then

$$\sup_x |\Pr(t < x) - \Phi(x)| \le \frac{6.4\mathbb{E}[|X|^3] + 2\mathbb{E}[|X|]}{\sqrt{n}}. \quad (18)$$

Proof. See Appendix, Section A.2. □

[5]The sample code provided by [Bardenet et al., 2014] computes $C_{\theta,\theta'}$ by traversing the entire data, thus providing no performance advantage over the complete test.

Lemma 3 demonstrates that if we know $\mathbb{E}[|X|]$ and $\mathbb{E}[|X|^3]$, we can bound the error of the normal approximation, which decays as $O(n^{-\frac{1}{2}})$. Making the change of variables $y = xs_X$, Equation (18) becomes

$$\sup_y \left|\Pr(\bar{X} < y) - \Phi\left(\frac{y}{s_X}\right)\right| \le \frac{6.4\mathbb{E}[|X|^3] + 2\mathbb{E}[|X|]}{\sqrt{n}} \quad (19)$$

showing that the distribution of $\bar{X}$ approaches the normal distribution $\mathcal{N}(0, s_X)$ whose variance is s_X, measured from the sample.

To apply this to our test, let $X_i = \Lambda_i(\theta,\theta') - \Lambda(\theta,\theta')$, so that the X_i are zero-mean, i.i.d. variables. If instead of all n samples, we only extract a subset of b samples corresponding to our minibatch, we can connect $\bar{X}$ with our Δ^* term: $\bar{X} = \Delta^*(\theta,\theta') - \Delta(\theta,\theta')$, so that $s_X = s_{\Delta^*}$. We can now substitute into Equation (19) and displace by the mean, giving:

Corollary 1.

$$\sup_y \left|\Pr(\Delta^* < y) - \Phi\left(\frac{y-\Delta}{s_{\Delta^*}}\right)\right| \le \frac{6.4\mathbb{E}[|X|^3] + 2\mathbb{E}[|X|]}{\sqrt{b}} \quad (20)$$

where the upper bound can be expressed as $\epsilon(\theta,\theta',b)$. Corollary 1 shows that the distribution of Δ^* approximates a Normal distribution with mean Δ and variance $s^2_{\Delta^*}$. Furthermore, it bounds the error with *estimable quantities*: both $\mathbb{E}[|X|]$ and $\mathbb{E}[|X|^3]$ can be estimated as means of $|\Lambda_i - \Lambda|$ and $|\Lambda_i - \Lambda|^3$, respectively, on each minibatch. We expect this will often be accurate enough on minibatches with hundreds of points, but otherwise bootstrap CIs can be computed.

5.2 ADDING RANDOM VARIABLES

We next relate the CDFs of distributions and show that bounds are preserved after adding random variables.

Lemma 4. *Let $P(x)$ and $Q(x)$ be two CDFs satisfying $\sup_x |P(x) - Q(x)| \le \epsilon$ with x in some real range. Let $R(y)$ be the* density *of another random variable y. Let P' be the convolution $P * R$ and Q' be the convolution $Q * R$. Then $P'(z)$ (resp. $Q'(z)$) is the CDF of sum $z = x + y$ of independent random variables x with CDF $P(x)$ (resp. $Q(x)$) and y with density $R(y)$. Then*

$$\sup_x |P'(x) - Q'(x)| \le \epsilon. \quad (21)$$

Proof. See Appendix, Section A.3. □

From Lemma 4, we have the following Corollary:

Corollary 2. *If $\sup_y |\Pr(\Delta^* < y) - \Phi(\frac{y-\Delta}{s_{\Delta^*}})| \le \epsilon(\theta,\theta',b)$, then*

$$\sup_y |\Pr(\Delta^* + X_{\text{nc}} + X_{\text{corr}} < y) - S(y-\Delta)| \le \epsilon(\theta,\theta',b)$$

where $S(x)$ is the standard logistic function, and X_{nc} and X_{corr} are generated as per Algorithm 1.

Proof. See Appendix, Section A.4. □

Corollary 2 shows that the bounds from Section 5.1 are preserved after adding random variables, so our test remains accurate. In fact we can do better ($O(n^{-1})$ instead of $O(n^{-1/2})$) by using a more precise limit distribution under an additional assumption. We review this in Appendix A.5.

5.3 BOUNDS ON THE STATIONARY DISTRIBUTION

Bounds on the error of an M-H test imply bounds on the stationary distribution of the Markov chain under appropriate conditions. Such bounds were derived in both [Korattikara et al., 2014] and [Bardenet et al., 2014]. We include the result from [Korattikara et al., 2014] (Theorem 1) here: Let $d_v(P, Q)$ denote the total variation distance between two distributions P and Q. Let $\mathcal{T}_0$ denote the transition kernel of the exact Markov chain, $\mathcal{S}_0$ denote the exact posterior distribution, and $\mathcal{S}_\epsilon$ denote the stationary distribution of the approximate transition kernel.

Lemma 5. *If $\mathcal{T}_0$ satisfies the contraction condition $d_v(P\mathcal{T}_0, \mathcal{S}_0) < \eta d_v(P, \mathcal{S}_0)$ for some constant $\eta \in [0, 1)$ and all probability distributions P, then*

$$d_v(S_0, S_\epsilon) \leq \frac{\epsilon}{1-\eta} \tag{22}$$

where ϵ is the bound on the error in the acceptance test.

6 EXPERIMENTS

In Sections 6.1 and 6.2, we benchmark MHMINIBATCH against the tests from [Bardenet et al., 2014] and [Korattikara et al., 2014], with the latter using a grid-search over minibatch sizes m and per-test thresholds ϵ described in Appendix B.2.1. Throughout our descriptions, we refer to a *trial* as the period when an algorithm collects all its desired samples $\{\theta_1, \ldots, \theta_T\}$, generally with $T = 3000$ or $T = 5000$.

6.1 MIXTURE OF GAUSSIANS

This model is adapted from [Welling and Teh, 2011] by increasing the number of samples to 1 million. The parameters are $\theta = \langle \theta_1, \theta_2 \rangle$, and the generation process is

$$\begin{aligned} \theta &\sim \mathcal{N}(0, \text{diag}(\sigma_1^2, \sigma_2^2)) \\ x_i &\sim 0.5 \cdot \mathcal{N}(\theta_1, \sigma_x^2) + 0.5 \cdot \mathcal{N}(\theta_1 + \theta_2, \sigma_x^2). \end{aligned} \tag{23}$$

We set $\sigma_1^2 = 10, \sigma_2^2 = 1$ and $\sigma_x^2 = 2$. We fix $\theta = \langle 0, 1 \rangle$. The original paper sampled 100 data points and estimated the posterior. We are interested in performance on larger problems and so sampled 1,000,000 points to form the posterior of $p(\theta) \prod_{i=1}^{1,000,000} p(x_i|\theta)^{1/K}$ with the same prior from Equation (23). This produces a much sharper posterior with two very narrow peaks. Our goal is to reproduce the original posterior, so we adjust the temperature to $K = 10,000$. Taking logs, we get the target as shown in the far left of Figure 1.

We benchmark with AUSTEREMH(C) and MHSUBLHD. We initialized MHMINIBATCH and MHSUBLHD with $m = 100$. For MHSUBLHD, we increase sizes geometrically with $\gamma = 1.5$ and use parameters $p = 2, \delta = 0.01$. All methods collect 5000 samples using a random walk proposer with covariance matrix $\text{diag}(0.15, 0.15)$, which means the M-H test is responsible for shaping the sample distribution.

Figure 1 shows scatter plots of the resulting θ samples for the three methods, with darker regions indicating a greater density of points. There are no obvious differences, showing that MHMINIBATCH reaches an acceptable posterior. We further measure the similarity between each set of samples and the actual posterior. Due to space constraints, results are in Appendix B.2.2.

Figure 2 shows that MHMINIBATCH dominates in terms of speed and efficiency. The histograms of the (final) minibatch sizes used each iteration show that our method consumes significantly less data; the distribution is short-tailed and the mean is 210, more than an order of magnitude better compared to the other two methods (averages are 15562 and 16857). Sizes correspond to the running times of the methods, excluding the likelihood computation of all data points for MHSUBLHD, which would drastically increase running time.

6.2 LOGISTIC REGRESSION

We next test logistic regression for the binary classification of 1s versus 7s on the MNIST [LeCun and Cortes, 1998] dataset and (a subset of) infinite MNIST [Loosli et al., 2007]. For the former, extracting all 1s and 7s resulted in 13,000 training samples, and for the latter, we used 87,000 additional (augmented) 1s and 7s to get 100,000 training samples. Both datasets use the same test set, with 2,163 samples. Henceforth, we call them MNIST-13k and MNIST-100k, respectively.

For all methods, we impose a uniform prior on θ and again use a random walk proposer, with covariance matrix $0.05I$ for MNIST-13k and $0.01I$ for MNIST-100k. The default temperature setting is a constant at $K = 100$ for MNIST-13k and MNIST-100k. Performance

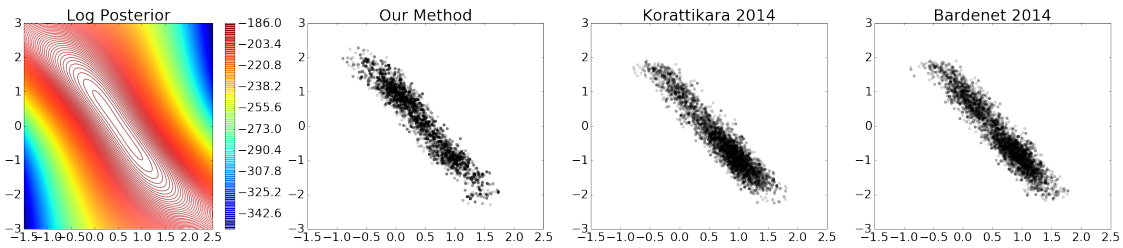

Figure 1: The log posterior contours and scatter plots of sampled θ values using different methods.

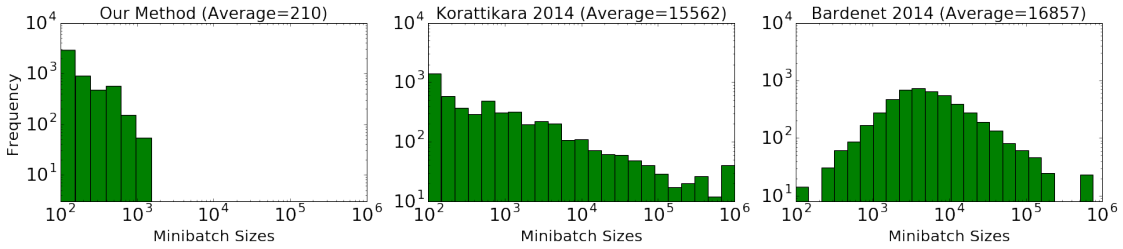

Figure 2: Minibatch sizes used in Section 6.1's experiment. The axes have the same (log-log scale) range.

of all methods implicitly relies on the step size and temperature. Setting temperature too low or step size too high will result in slow convergence for all methods. For MNIST-13k, each method generated 5000 samples for ten independent trials; due to MNIST-100k's higher computational requirements, the methods generated 3000 samples for five independent trials. For additional parameter settings and an investigation on tuning step sizes, see Appendix B.3.

For MHSUBLHD, we tried to use the provided symbolic bound for $C_{\theta,\theta'}$ described in [Bardenet et al., 2014], but it was too high and provided no performance benefit. Instead we use the empirical $C_{\theta,\theta'}$ from the entire dataset.

The first two subplots of Figure 3 display the prediction accuracy on both datasets for all methods as a function of the cumulative training points processed.[6] To generate the curves, for each of the sampled vectors θ_t, $t \in \{1, \ldots, T\}$, we use θ_t as the logistic regression parameter. The results indicate that our test is more efficient, obtaining convergence more than an order of magnitude faster than AUSTEREMH(NC) and several orders of magnitude compared to AUSTEREMH(C) and MHSUBLHD. We also observe the advantage of having higher temperature from the third plot in Figure 3. During the exploration period, the accuracy rapidly increases, and then after 200 samples, we switch the temperature to 1, but this requires the step size to decrease, hence the smaller changes in accuracy.

[6]The curves do not span the same length over the x-axis since the methods consume different amounts of data.

Table 1: Average minibatch sizes ($\pm$ one standard deviation) on logistic regression on MNIST-13k and MNIST-100k. The averages are taken over 10 independent trials (5000 samples each) for MNIST-13k and 5 independent trials (3000 samples each) for MNIST-100k.

Method/Data	**MNIST-13k**	**MNIST-100k**
MHMINIBATCH	129.8 ± 3.4	202.5 ± 13.6
AUSTEREMH(NC)	973.8 ± 49.8	1098.3 ± 44.9
AUSTEREMH(C)	1924.3 ± 52.4	2795.6 ± 364.0
MHSUBLHD	10783.4 ± 78.9	14977.3 ± 582.0

Figure 4 shows log-log histograms of minibatch sizes for the three methods on MNIST-100k. (Figure 5 in Appendix B.3 contains results for MNIST-13k.) The histograms only represent one representative trial; Table 1 contains the average of the average minibatch sizes ($\pm$ one standard deviation) across all trials. MHMINIBATCH, with average minibatch sizes of roughly 129.8 and 202.5 for MNIST-13k and MNIST-100k, respectively, consumes more than 5x fewer data points than the next-best method, AUSTEREMH(NC). We reiterate, however, that both AUSTEREMH(NC) and MHSUBLHD require computing $\log p(x_i|\theta)$ and $\log p(x_i|\theta')$ for all x_i each iteration. Our results here do not count that extra data consumption. Only our method and AUSTEREMH(C) rely solely on the minibatch of data each iteration.

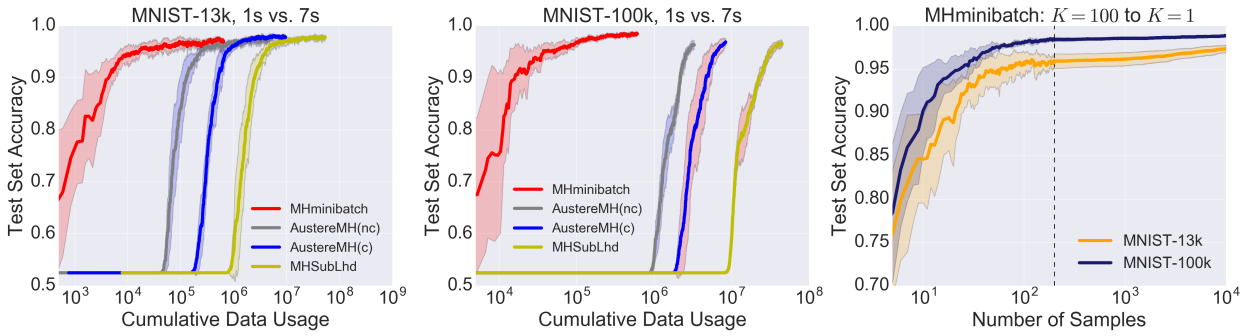

Figure 3: Binary classification accuracy of the MCMC methods on the 1s vs 7s logistic regression task for MNIST-13k (left plot) and MNIST-100k (middle plot) as a function of cumulative data usage. The right plot reports performance of MHMINIBATCH on both datasets when the temperature starts at 100 and drops to 1 after a "burn-in" period of 200 samples (vertical dashed line) of θ. For all three plots, one standard deviation is indicated by the shaded error regions.

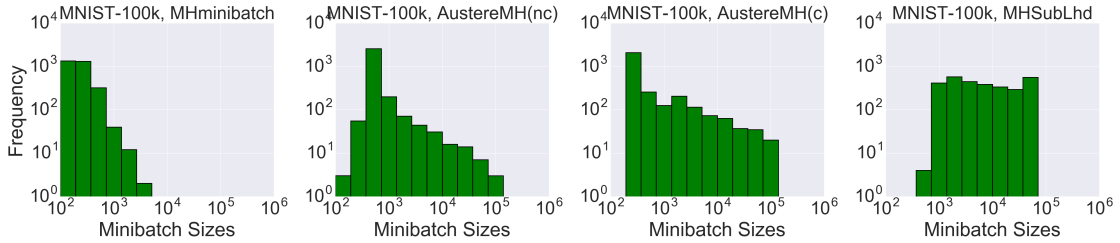

Figure 4: Minibatch sizes for a representative trial of logistic regression on MNIST-100k (analogous to Figure 2). Both axes are on a log scale and have the same ranges across the three histograms. See Section 6.2 for details.

Table 2: **TODO TABLE WITH SMF RESULTS**

RMSE	**MovieLens 10m**	**Netflix**
MHMINIBATCH	???	???
ADAGrad	???	???

6.3 SPARSE MATRIX FACTORIZATION

Daniel: TODO this is our third experiment. Results are in progress...

We benchmark with the ADAGrad [Duchi et al., 2011] implementation of Sparse Matrix Factorization described in [Canny and Zhao, 2013]. (Daniel: is this the correct citation for BIDMach? I assume this is what we will use)

7 CONCLUSIONS

We have derived an M-H test for minibatch MCMC which approximates full data tests. We present theoretical results and experimentally show the benefits of our test on Gaussian mixtures, logistic regression, and sparse matrix factorization. Directions for future work include integrating our algorithm with [Korattikara et al., 2014] by applying both tests each iteration or utilizing the variance reduction techniques suggested in [Chen and Ghahramani, 2016]. More elaborate extensions involve running more experiments on other models such as neural networks and providing a recipe for how to use our algorithm (following the framework of [Ma et al., 2015]).

References

Sungjin Ahn, Anoop Korattikara Balan, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. Towards scaling up markov chain monte carlo: an adaptive subsampling approach. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On markov chain monte carlo methods for tall data. *Journal of Machine Learning Research (JMLR)*, 2016.

A. A. Barker. Monte-carlo calculations of the radial dis-

tribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18:119–133, 1965.

V. Bentkus, F. Gotze, and W.R.vanZwet. An edgeworth expansion for symmetric statistics. *Annals of Statistics*, 25(2), 1997.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

John Canny and Huasha Zhao. Big data analytics with small footprint: Squaring the cloud. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, 2013.

T. Chen, E.B. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

Yutian Chen and Zoubin Ghahramani. Scalable discrete sampling as a multi-armed bandit problem. In *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 2016.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surface of multilayer networks. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics, AISTATS*, 2015.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*, 12: 2121–2159, July 2011. ISSN 1532-4435.

W.R. Gilks and DJ Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57: 97–109, 1970.

Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the metropolis-hastings budget. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Gaëlle Loosli, Stéphane Canu, and Léon Bottou. Training invariant support vector machines using selective sampling. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA., 2007.

Y. Ma, T. Chen, and E.B. Fox. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems 28*, 2015.

Dougal Maclaurin and Ryan P. Adams. Firefly monte carlo: Exact MCMC with subsets of data. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, (UAI)*, 2014.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21, 1953.

Radford M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54: 113–162, 2010.

Y. Novak. On self-normalized sums and students statistic. *Theory of Probability and its Applications*, 49(2): 336–344, 2005.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various metropolishastings algorithms. *Statistical Science*, 16(4):351367, 2001.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

Appendix

This appendix is divided into two main parts. Appendix A provides the proofs that we omitted from the main text due to space constraints. Appendix B provides further details on the correction distribution derivation and on our three main experiments to assist understanding and reproducibility. Our code is open-source on GitHub.[7]

A PROOFS OF LEMMAS AND COROLLARIES

A.1 PROOF OF LEMMA 1

Choose $(\theta' - \theta) \in \pm\frac{1}{\sqrt{N}}[0.5, 1]$ (event 1) and $(\theta - 0.5) \in \pm\frac{1}{\sqrt{N}}[0.5, 1]$ filtered for matching sign (event 2). As discussed in Lemma 1, both $q(\theta'|\theta)$ and $p(\theta|x_1, \ldots, x_N)$ have variance $1/N$. If we denote Φ as the CDF of the standard normal distribution, then the former event occurs with probability $p_0 = 2(\Phi(\sqrt{N}\frac{1}{\sqrt{N}}) - \Phi(\sqrt{N}\frac{0.5}{\sqrt{N}})) = 2(\Phi(1) - \Phi(0.5)) \approx 0.2997$. The latter event, because we restrict signs, occurs with probability $p_1 = \Phi(1) - \Phi(0.5) \approx 0.14988$.

These events together guarantee that $\Lambda^*(\theta, \theta')$ is negative by inspection of Equation (24) below. This implies that we can find a $u \in (0, 1)$ so that $\psi(u, \theta, \theta') = \log u < 0$ equals $\mathbb{E}[\Lambda^*(\theta, \theta')]$. Specifically, choose u_0 to satisfy $\log u_0 = \mathbb{E}[\Lambda^*(\theta, \theta')]$. Using $\mathbb{E}[x_i^*] = 0.5$ and Equation (5), we see that

$$\log u_0 = N(\theta' - \theta)\frac{1}{b} \cdot \mathbb{E}\left[\sum_{i=1}^{b} x_i^* - \theta - \frac{\theta' - \theta}{2}\right] = -N(\theta' - \theta)\left(\theta - 0.5 + \frac{\theta' - \theta}{2}\right). \tag{24}$$

Next, consider the minibatch acceptance test $\Lambda^*(\theta, \theta') \not\approx \psi(u, \theta, \theta')$ used in [Korattikara et al., 2014] and [Bardenet et al., 2014] , where $\not\approx$ means "significantly different from" under the distribution over samples. This is

$$\Lambda^*(\theta, \theta') \not\approx \psi(u_0, \theta, \theta') \iff N(\theta' - \theta) \cdot \frac{1}{b}\sum_{i=1}^{b} x_i^* - \theta - \frac{\theta' - \theta}{2} \not\approx \log u_0 \tag{25}$$

$$\iff \frac{1}{b}\sum_{i=1}^{b} x_i^* - \left(\theta + \frac{\theta' - \theta}{2} + \frac{\log u_0}{N(\theta' - \theta)}\right) \not\approx 0 \tag{26}$$

$$\iff \frac{1}{b}\sum_{i=1}^{b} x_i^* - 0.5 \not\approx 0. \tag{27}$$

Since the x_i^* have mean 0.5, the resulting test with our chosen u_0 will never correctly succeed and must use all N data points. Furthermore, if we sample values of u near enough to u_0, the terms in parenthesis will not be sufficiently different from 0.5 to allow the test to succeed.

The choices above for θ and θ' guarantee that

$$\log u_0 \in -[0.5, 1][0.75, 1.5] = [-1.5, -0.375]. \tag{28}$$

Next, consider the range of u values near u_0:

$$\log u \in \log u_0 + [-0.5, 0.375]. \tag{29}$$

The size of the range in u is at least $\exp([-2, -1.125]) \approx [0.13534, 0.32465]$ and occurs with probability at least $p_2 = 0.18932$. With u in this range, we rewrite the test as:

$$\frac{1}{b}\sum_{i=1}^{b} x_i^* - 0.5 \quad \not\approx \quad \frac{\log u/u_0}{N(\theta' - \theta)} \tag{30}$$

[7] https://github.com/BIDData/BIDMach/blob/master/src/main/scala/BIDMach/updaters/MHTest.scala

so that, as in Equation (27), the LHS has expected value zero. Given our choice of intervals for the variables, we can compute the range for the right hand side (RHS) assuming[8] that $\theta' - \theta > 0$:

$$\min\{\text{RHS}\} = \frac{-0.5}{\sqrt{N} \cdot 0.5} = -\frac{1}{\sqrt{N}} \quad \text{and} \quad \max\{\text{RHS}\} = \frac{0.375}{\sqrt{N} \cdot 0.5} = \frac{0.75}{\sqrt{N}} \tag{31}$$

Thus, the RHS is in $\frac{1}{\sqrt{N}}[-1, 0.75]$. The standard deviation of the LHS given the interval constraints is at least $0.5/\sqrt{b}$. Consequently, the gap between the LHS and RHS in Equation (30) is at most $2\sqrt{b/N}$ standard deviations, limiting the range in which the test will be able to "succeed" without requiring more samples.

The samples θ, θ' and u are drawn independently and so the probability of the conjunction of these events is $c = p_0 p_1 p_2 = 0.0085$.

A.2 PROOF OF LEMMA 3

The following bound is given immediately after Corollary 2 from [Novak, 2005]:

$$-6.4\mathbb{E}[|X|^3] - 2\mathbb{E}[|X|] \leq \sup_x |\Pr(t < x) - \Phi(x)|\sqrt{n} \leq 1.36\mathbb{E}[|X|^3]. \tag{32}$$

This bound applies to $x \geq 0$. Applying the bound to $-x$ when $x < 0$ and combining with $x > 0$, we obtain the weaker but unqualified bound in Equation (18).

A.3 PROOF OF LEMMA 4

We first observe that

$$P'(z) - Q'(z) = \int_{-\infty}^{+\infty} (P(z-x) - Q(z-x))R(x)dx,$$

and since $\sup_x |P(x) - Q(x)| \leq \epsilon$ it follows that $\forall z$:

$$-\epsilon = \int_{-\infty}^{+\infty} -\epsilon R(x)dx \leq \int_{-\infty}^{+\infty} (P(z-x) - Q(z-x))R(x)dx \leq \int_{-\infty}^{+\infty} \epsilon R(x)dx = \epsilon, \tag{33}$$

as desired.

A.4 PROOF OF COROLLARY 2

We apply Lemma 4 twice. First take:

$$P(y) = \Pr(\Delta^* < y) \quad \text{and} \quad Q(y) = \Phi\left(\frac{y - \Delta}{s_{\Delta^*}}\right) \tag{34}$$

and convolve with the distribution of X_n which has density $\phi(X/\sigma_n)$ where $\sigma_n^2 = 1 - s_{\Delta^*}^2$. This yields the next iteration of P and Q:

$$P'(y) = \Pr(\Delta^* + X_{\text{nc}} < y) \quad \text{and} \quad Q'(y) = \Phi\left(y - \Delta\right) \tag{35}$$

Now we convolve with the distribution of X_{corr}:

$$P''(y) = \Pr(\Delta^* + X_{\text{nc}} + X_{\text{corr}} < y) \quad \text{and} \quad Q''(y) = S\left(y - \Delta\right) \tag{36}$$

Both steps preserve the error bound $\epsilon(\theta, \theta', b)$. Finally $S(y - \Delta)$ is a logistic CDF centered at Δ, and so $S(y - \Delta) = \Pr(\Delta + X_{\log} < y)$ for a logistic random $X_{\log}$. We conclude that the probability of acceptance for the actual test $\Pr(\Delta^* + X_{\text{nc}} + X_{\text{corr}} > 0)$ differs from the exact test $\Pr(\Delta + X_{\log} > 0)$ by at most ϵ.

[8] If $\theta' - \theta < 0$, then the range would be $\frac{1}{\sqrt{N}}[-0.75, 1]$ but this does not matter for the purposes of our analysis.

A.5 IMPROVED ERROR BOUNDS BASED ON SKEW ESTIMATION

We show that the CLT error bound can be improved to $O(n^{-1})$ using a more precise limit distribution under an additional assumption. Let μ_i denote the i^{th} moment, and b_i denote the i^{th} absolute moment of X. If Cramer's condition holds:

$$\lim_{t\to\infty} \sup |\mathbb{E}[\exp(itX)]| < 1, \tag{37}$$

then Equation 2.2 in Bentkus et al.'s work on Edgeworth expansions [Bentkus et al., 1997] provides:

Lemma 6. *Let $X_1, \ldots, X_n$ be a set of zero-mean, independent, identically-distributed random variables with sample mean $\hat{X}$ and with t defined as in Lemma 3. If X satisfies Cramer's condition, then*

$$\sup_x \left| \Pr(t < x) - G\left(x, \frac{\mu_3}{b_2^{3/2}}\right) \right| \leq \frac{c(\epsilon, b_2, b_3, b_4, b_{4+\epsilon})}{n}$$

where

$$G_n(x, y) = \Phi(x) + \frac{y(2x^2+1)}{6\sqrt{n}}\Phi'(x). \tag{38}$$

Lemma 6 shows that the average of the X_i has a more precise, skewed CDF limit $G_n(x, y)$ where the skew term has weight proportional to a certain measure of skew derived from the moments: $\mu_3/b_2^{3/2}$. Note that if the X_i are symmetric, the weight of the correction term is zero, and the CDF of the average of the X_i converges to $\Phi(x)$ at a rate of $O(n^{-1})$.

Here the limit $G_n(x, y)$ is a normal CDF plus a correction term that decays as $n^{-1/2}$. Importantly, since $\phi''(x) = x^2\phi(x) - \phi(x)$ where $\phi(x) = \Phi'(x)$, the correction term can be rewritten giving:

$$G_n(x, y) = \Phi(x) + \frac{y}{6\sqrt{n}}(2\phi''(x) + 3\phi(x)) \tag{39}$$

From which we see that $G_n(x, y)$ is a linear combination of $\Phi(x)$, $\phi(x)$ and $\phi''(x)$. In Algorithm 1, we correct for the difference in σ between Δ^* and the variance needed by X_{corr} using X_{nc}. This same method works when we wish to estimate the error in Δ^* vs $G_n(x, y)$. Since all of the component functions of $G_n(x, y)$ are derivatives of a (unit variance) $\Phi(x)$, adding a normal variable with variance σ' increases the variance of all three functions to $1 + \sigma'$. Thus we add X_{nc} as per Algorithm 1 preserving the limit in Equation (39).

The deconvolution approach can be used to construct a correction variable X_{corr} between $G_n(x, y)$ and $S(x)$ the standard logistic function. An additional complexity is that $G_n(x, y)$ has additional parameters y and n. Since these act as a single multiplier $\frac{y}{6\sqrt{n}}$ in Equation (39), its enough to consider a function $g(x, y')$ parametrized by $y' = \frac{y}{6\sqrt{n}}$. This function can be computed and saved offline. As we have shown earlier, errors in the "limit" function propagate directly through as errors in the acceptance test. To achieve a test error of 10^{-6} (close to single floating point precision), we need a y' spacing of 10^{-6}. It should not be necessary to tabulate values all the way to $y' = 1$, since y' is scaled inversely by the square root of minibatch size. Assuming a max y' of 0.1 requires us to tabulate about 100,000. Since our x resolution is 10,000, this leads to a table with about 1 billion values, which can comfortably be stored in memory. However, if $g(x, y)$ is moderately smooth in y, it should be possible to achieve similar accuracy with a much smaller table. We leave further analysis and experiments with $g(x, y)$ as future work.

B ADDITIONAL EXPERIMENT DETAILS

B.1 OBTAINING THE CORRECTION DISTRIBUTION (SECTION 4)

In Section 4, we described our derivation of the correction distribution C_σ for random variable X_{corr}. Table 3 shows our L_∞ error results for the convolution (Equation (14)) based on various hyperparameter choices. We test using $N = 2000$ and $N = 4000$ points for discretization within a range of $X_{\text{corr}} \in [-20, 20]$, covering essentially all the probability mass. We also vary σ from 0.8 to 1.1.

We observe the expected tradeoff. With smaller σ, our C_σ is closer to the ideal distribution (as judged by L_∞ error), but this imposes a stricter upper bound on the sample variance of Δ^* before our test can be applied, which thus results

Table 3: Errors (L_∞) in $X_{\text{norm}} + X_{\text{corr}}$ versus $X_{\log}$, with $N = 4000$ (top row) and $N = 2000$ (bottom row).

$N = 2000$	$\sigma = 0.8$
λ	L_∞ error
100	2.6e-3
10	4.0e-4
1	6.7e-5
0.1	1.4e-5
0.01	**5.0e-6**

$N = 2000$	$\sigma = 0.9$
λ	L_∞ error
100	3.3e-3
10	6.4e-4
1	1.6e-4
0.1	**1.3e-4**
0.01	2.7e-4

$N = 2000$	$\sigma = 1.0$
λ	L_∞ error
100	4.4e-3
10	1.3e-3
1	**1.1e-3**
0.1	2.0e-3
0.01	3.6e-3

$N = 2000$	$\sigma = 1.1$
λ	L_∞ error
100	6.8e-3
10	**4.6e-3**
1	7.5e-3
0.1	1.3e-2
0.01	2.4e-2

$N = 4000$	$\sigma = 0.8$
λ	L_∞ error
100	8.3e-4
10	1.3e-4
1	2.5e-5
0.1	**6.7e-6**
0.01	7.4e-6

$N = 4000$	$\sigma = 0.9$
λ	L_∞ error
100	1.2e-3
10	2.6e-4
1	**1.0e-4**
0.1	2.0e-4
0.01	3.9e-4

$N = 4000$	$\sigma = 1.0$
λ	L_∞ error
100	1.9e-3
10	**8.9e-4**
1	1.6e-3
0.1	2.8e-3
0.01	5.2e-3

$N = 4000$	$\sigma = 1.1$
λ	L_∞ error
100	**4.3e-3**
10	6.0e-3
1	1.0e-2
0.1	1.2e-2
0.01	3.5e-2

Table 4: Gaussian Mixture Model Statistics

Metric	MHMINIBATCH	AUSTEREMH	MHSUBLHD
Equation 40	-1430.0	-1578.9	-1232.7
Chi-Squared	3313.9	3647.7	2444.1

in larger minibatch sizes. Conversely, a more liberal upper bound means we avail ourselves of smaller minibatch sizes, but at the cost of a less stable derivation for C_σ.

We chose $N = 4000, \sigma = 1$, and $\lambda = 10$ to use in our experiments, which empirically exhibits excellent performance. This is reflected in the description of MHMINIBATCH in Algorithm 1, which assumes that we used $\sigma = 1$ but we reiterate that the choice is arbitrary so long as $0 < \sigma < \sqrt{\pi^2/3} \approx 1.814$, the standard deviation of the standard logistic distribution, since there must be some variance left over for X_{corr}.

B.2 GAUSSIAN MIXTURE MODEL EXPERIMENT (SECTION 6.1)

B.2.1 Grid Search

For the Gaussian mixture experiment, we use the conservative method from [Korattikara et al., 2014], which avoids the need for recomputing log likelihoods of each data point each iteration by choosing baseline minibatch sizes m and per-test thresholds ϵ beforehand, and then using those values for the entirety of the trials. We experimented with the following values:

- $\epsilon \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.2\}$
- $m \in \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$

and chose the (m, ϵ) pairing which resulted in the lowest expected data usage given a selected upper bound on the error. Through personal communication with Korattikara et al. [2014], we were able to use their same code to compute expected data usage and errors.

The main difference between AUSTEREMH(C) and AUSTEREMH(NC)[9] is that the latter needs to run a grid search each iteration (i.e. after each time it makes an accept/reject decision for one sample θ_t). We use the same ϵ and m candidates above for AUSTEREMH(NC).

[9] AUSTEREMH(NC) is used in Section 6.2.

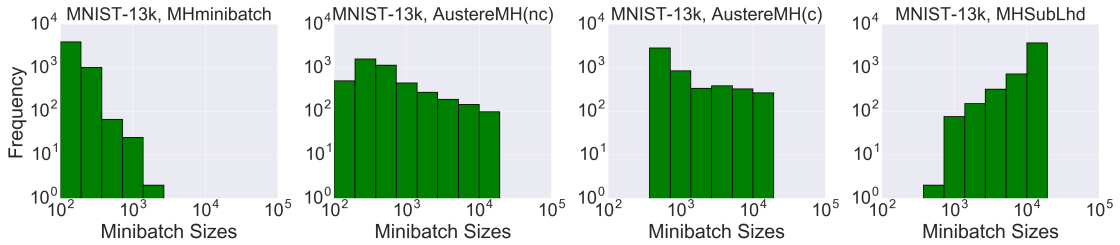

Figure 5: Minibatch sizes for a representative trial of logistic regression on MNIST-13k (analogous to Figure 2). Both axes are on a log scale and have the same ranges across the three histograms. See Section 6.2 for details.

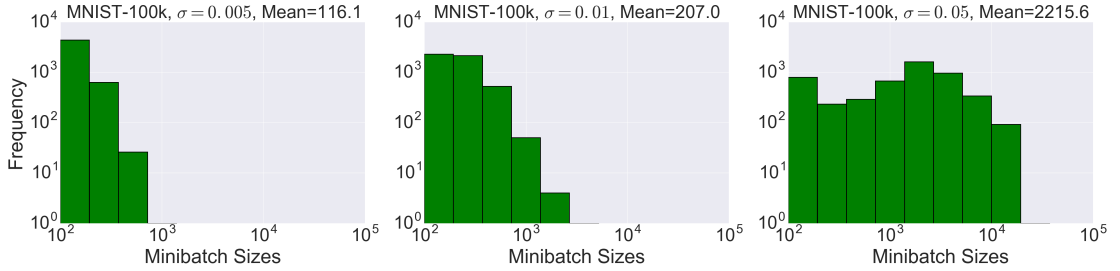

Figure 6: **Daniel: TODO describe**

B.2.2 Gaussian Mixture Model Metrics

Daniel: Xinlei, you should be in charge of checking this section and adding some stuff to Table 4, perhaps averages plus or minus one standard deviation.

We discretize the posterior coordinates into bins with respect to the two components of θ. The probability P_i of a sample falling into bin i is the integral of the true posterior over the bin's area. A single sample should therefore be multinomial with distribution P, and a set of n (ideally independent) samples is $\mathrm{Multinomial}(P, n)$. This distribution is simple and we can use it to measure the quality of the samples rather than use general purpose tests like KL-divergence or likelihood-ratio, which are problematic with zero counts.

For large n, the per-bin distributions are approximated by Poissons with parameter $\lambda_i = P_i n$. Given samples $\{\theta_1, \ldots, \theta_T\}$, let c_j denote the number of individual samples θ_i that fall in bin j out of N_{bins} total. We have

$$\log p(c_1, \ldots, c_{N_{\text{bins}}} | P_1, \ldots, P_{N_{\text{bins}}}) = \sum_{j=1}^{N_{\text{bins}}} c_j \log(nP_j) - nP_j - \log(\Gamma(c_j + 1)). \tag{40}$$

Table 4 shows the likelihoods. To facilitate interpretation we perform significance tests using Chi-Squared distribution (also in Table 4). Scores lie between [Korattikara et al., 2014] and [Bardenet et al., 2014], but the variance of these values is high and ordering changes depending on the range of samples generated. **Daniel: one reviewer had a problem with this, and I agree. The easiest fix here is to simply report standard deviation values. Also, we should emphasize that we are not aiming to get the best score by this metric but simply something that exhibits some form of "correctness."**

B.3 LOGISTIC REGRESSION EXPERIMENT (SECTION 6.2)

Figure 5 shows the histograms for the four methods on one representative trial of MNIST-13k, which shows similar relative performance of the four methods as shown in Figure 4 (which uses MNIST-100k). In particular, MHMINIBATCH exhibits a shorter-tailed distribution and consumes nearly an order of magnitude fewer data points compared to AUSTEREMH(NC), the next-best method; see Table 1 for details.

Next, we investigate the impact of the step size σ for the random walk proposers with covariance matrix σI. Note that

Table 5: Parameters for the logistic regression experiments.

Value	MNIST-13k	MNIST-100k
Temperature K	100	100
Number of samples T	5000	3000
Number of trials	10	5
Step size σ for random walk proposer with covariance σI	0.05	0.01
MHMINIBATCH and MHSUBLHD minibatch size m	100	100
AUSTEREMH(C) chosen Δ^* bound	0.1	0.2
AUSTEREMH(C) minibatch size m from grid search	450	300
AUSTEREMH(C) per-test threshold ϵ from grid search	0.01	0.01
AUSTEREMH(NC) chosen Δ^* bound	0.05	0.1
MHSUBLHD γ	2.0	2.0
MHSUBLHD p	2	2
MHSUBLHD δ	0.01	0.01

I is 784×784 as we did not perform any downsampling or data preprocessing other than rescaling the pixel values to lie in $[0, 1]$.

For this, we use the larger dataset MNIST-100k, and test with $\sigma \in \{0.005, 0.01, 0.05\}$. We keep other parameters consistent with the experiments in Section 6.2, in particular, keeping the initial minibatch size $m = 100$, which is also the amount the minibatch increments by if we need more data. Figure 6 indicates minibatch histograms (again, using the log-log scale) for one trial of MHMINIBATCH using each of the step sizes. We observe that by tuning MHMINIBATCH, we are able to adjust the average number of data points in a minibatch across a wide range of values. Here, the smallest step size results in an average of just 116.1 data points per minibatch, while increasing to $\sigma = 0.05$ (the step size used for MNIST-13k) results in an average of 2215.6. This relative trend is also present for both AUSTEREMH variants and MHSUBLHD.

Table 5 indicates the relevant parameter settings for the logistic regression experiments. Unless otherwise stated, values apply to all methods tested. For values from [Korattikara et al., 2014] or Bardenet et al. [2014], we use their notation to be consistent.

B.4 SPARSE MATRIX FACTORIZATION EXPERIMENT (SECTION 6.3)

Daniel: TODO