Adaptive Metropolis Sampling and Optimization with Product Distributions

David H. Wolpert[1*] and Chiu Fan Lee[2†]

[1]NASA Ames Research Center
MailStop 269-1, Moffett Field, CA 94035-1000

[2]Clarendon Laboratory
Physics Department, Oxford University
Oxford OX1 3PU, U.K.

January 19, 2005

Abstract

The Metropolis-Hastings (MH) algorithm is a way to IID sample a provided target distribution $\pi(x)$. It works by repeatedly sampling a separate proposal distribution $T(x, x')$ to generate a random walk $\{x(t)\}$ which converges to a set of samples of π. Here, we introduce a T-updating phase after the cooling period and before sampling begins. In the updating phase, $\{x(t)\}$ is used to update T at t and our update method corresponds to the information-theoretically optimal mean-field approximation to π. We employ our algorithm to sample the energy distribution for several spin-glasses and we demonstrate the superiority of our algorithm to the conventional MH algorithm.

*dhw@email.arc.nasa.gov
†c.lee1@physics.ox.ac.uk

1 Introduction

1.1 Overview of Metropolis-Hastings

Monte Carlo methods are a powerful tool for evaluating integrals and simulating stochastic systems. The core of any such method is an algorithm for producing a many-point IID sample of a provided **target probability distribution** $\pi(x \in X)$. Often this is done by using the ratio $\pi(x')/\pi(x'')$ to fill in a Markov transition matrix $a(x', x'') \equiv P(x(t+1) = x' \mid x(t) = x'')$. That matrix is then iteratively applied starting from a randomly chosen initial x. For a proper relationship between a and π, the resultant random walk asymptotically gives the desired IID sample of $\pi(x)$.

One popular method for constructing the transition matrix a is the Metropolis-Hastings (MH) algorithm [1, 2, 3, 4]. This a produced by this algorithm is parameterized by a **proposal distribution** $T(x, x')$. Typically T is set before the start of the Markov chain in a π-independent manner and fixed throughout the running of that chain. The rate at which the random walk produced in the associated Markov chain converges to the desired IID sample is crucially dependent on the relation between T and π however.

An important example of this is that if $T(x, x') = \pi(x)$, then the Markov chain produced by MH is a perfect IID sampler of π (see below). Unfortunately, typically one cannot exploit this because one cannot evaluate $\pi(x)$. (Only the ratios of $\pi(x)$'s values for different x can be evaluated.) However since the set $\{x(t)\}$ produced by the MH algorithm is (eventually) an IID sample of π, one can use $\{x(t)\}$ to produce a empirical estimate of π [5]. This suggests that we empirically update T during the random walk to be an increasingly accurate estimate of π.

1.2 Approximating the target distribution

Typically X is high-dimensional, and for the density estimation of π to work well it must be restricted to producing estimates from a relatively low-dimensional space, $\mathcal{Q}$. Intuitively, the idea is to try to find the $q \in \mathcal{Q}$ that is "closest" to π and use that to update T, presuming that this will produce the most quickly converging random walk. We generically call such algorithms Adaptive Metropolis Hastings (AMH). To specify an AMH algorithm one must fix the measure of closeness, the choice of $\mathcal{Q}$, and the precise details of the resultant density estimation algorithm. One must then specify how the estimates of $\pi(x)$ are used to update $T(x, x')$.

The most popular way to measure closeness between probability distributions is with the Kullback-Leibler (KL) distance [6]:

$$D(p||p') \equiv -\sum_x p(x)\ln[p'(x)/p(x)] \quad (1)$$

Recent work in Probability Collectives [12, 7, 8, 11] provides insight into how to do density estimation to minimize KL distance when $\mathcal{Q}$ is low-dimensional. In particular, say $\mathcal{Q}$ is the set of all product distributions over X, $q(x) = \prod_i q_i(x_i)$. (Loosely speaking, this is equivalent to a mean-field approximation.) Then $D(q||\pi)$ is minimized if

$$q_i(x_i) \propto e^{-\beta E(\ln(\pi)|x_i)} \ , \ \forall i \quad (2)$$

as shown in [7, 8]. (Note that $D(q||\pi)$ is just the associated fre energy of q if π is a Boltzmann distribution.)

Unfortunately, the expectation values in Eq. 2 depends on the $q_{j \neq i}$. Usually we cannot solve this coupled set of equations in closed form. As an alternative though, we can use sampling processes to perform an iterative search for the q that minimizes $D(q||\pi)$. More concretely, we can use IID samples of q to form estimates of $E(\ln(\pi) \mid x_i = s)$ for all variables x_i and associated potential values s. Those estimates are all that is needed to perform a step in a Newton's method search for $\text{argmin}_q[D(q||\pi)]$ [7, 8, 9]. Because $\mathcal{Q}$ is a product distribution, this estimation procedure scales well to large spaces X. Moreover, the estimates and the associated updates of q are parallelized by construction, lending the algorithm to particularly fast implementation.

Here, we take a different approach and consider the KL distance from π to q rather than vice-versa. This is arguably a more appropriate kind of distance measure, given the information-theory derivation of KL distance [7, 8]. Moreover, the product distribution minimizing this distance can be written down directly: it has the same marginals as π, i.e., the optimal q obeys $q_i = \pi_i \ \forall i$ [7, 8]. And as the random walk of the conventional MH algorithm converges to the desired IID sample of π, the i'th component of the elements of the random walk, $\{x_i(t)\}$, becomes an IID sample of π_i. So if the number of possible x_i values is not too large, we can use simple histogramming of the elements of the random walk produced by the MH algorithm to form our estimate of each marginal π_i, and therefore of the q minimizing $D(q||\pi)$. In particular, we don't have to run a parallel process of IID sampling of q and updating it accordingly to form such an estimate.

In the next section we review the MH algorithm's details. Next we present the details of our AMH algorithm here. We end with experiments validating our algorithm.

2 Metropolis-Hastings algorithm

For a transition matrix $a(x, y)$ to preserve probability, $\sum_x a(x, y) = 1\ \forall y$ (since $\sum_{x,z} a(x, z)\delta(y, z)$ must equal 1 for all y). Conservation of probability also means that any eigenfunctions that lie on the unit simplex have eigenvalue 1, i.e., they are fixed points of a. All other eigenfunctions connect points within the simplex, i.e., have the sum of their components $= 0$.[1] Those eigenfunctions cannot have eigenvalues > 1, as otherwise repeated application of a to a point in the simplex that isn't an eigenfunction would map it off the simplex. Similarly, if a has only one fixed point, they can't have eigenvalues of 1, and so must have eigenvalues < 1. So if a has just a single eigenfunction and we express a distribution in terms of the eigenfunctions of a, we see that running that distribution through that matrix maps the distribution geometrically closer (say according to a L^2 norm) to a's fixed point.

So if a has only one fixed point, a Markov chain based on a will map any initial point x' (i.e., any initial distribution over x values, $\delta(x, x')$) to that fixed point distribution. In other words, if π is that fixed point distribution, then for large enough n we can write

$$\pi(x(n)) \approx \int dx(1)dx(2)\ldots dx(n-1)\ \delta(x(1), x') \prod_{t=2}^{n} a(x(t), x(t-1)) \quad (3)$$

Conversely, say we use a to construct a random walk, i.e., say we iterate the (Markovian) process of applying $a(x, x(t))$ to the current point $x(t)$ to get a distribution over x, which is then sampled to get the next point $x(t + 1)$. If the first point in the walk is set to x', then the probability that the random walk is the set of points $\{(x(t)\}$ is just $\delta(x(1), x') \prod_{t=2}^{n} a(x(t), x(t-1))$. Marginalizing this over all $x(t < n)$ and comparing to Eq. 3, we see that the probability distribution of the n'th point in the walk is just π.

So say we produce many random walks and look at the set of last points in each one. Those points will form a many-point IID sample of π. Since this is true for any starting point x', we can instead daisy-chain those Markov processes one after the other, i.e., run one particularly long Markov chain to get many samples of π. The MH algorithm exploits this by constructing a transition matrix whose single fixed point is the desired distribution π. It works as follows:

1. Given current state $x(t)$, draw y from the proposal distribution $T(x(t), y)$.

[1] Write $\sum_y a(x, y)v(y) = \alpha v(x)$, and then sum both sides over x. Since $\sum_x a(x, y) = 1$, we get $\sum_y v(y) = \alpha \sum_x v(x)$, which for $\alpha \neq 1$ implies $\sum_x v(x) = 0$.

2. Draw a random number r uniformly in $[0, 1]$ and update

$$x(t+1) = \begin{cases} y, & \text{if } r \le R(x(t), y) \\ x(t), & \text{otherwise} \end{cases} \tag{4}$$

where

$$R(x, y) = \min\left\{1 \ , \ \frac{\pi(y)T(y,x)}{\pi(x)T(x,y)}\right\}. \tag{5}$$

3. Repeat from step 1.

Note that, as claimed previously, for $T(s, t) = \pi(t)$, R always equals 1, and the newly sampled point is always accepted.

Let b^t be the distribution at time t. Then the MH algorithm is equivalent to multiplying b^t by the transition matrix

$$\begin{aligned} a(x(t+1) \neq x(t), x(t)) \;\; &= \\ T(x(t), x(t+1)) \min[1, &\frac{T(x(t+1), x(t))\ \pi(x(t+1))}{T(x(t), x(t+1))\ \pi(x(t))}] \end{aligned} \tag{6}$$

with $a(x(t), x(t))$ given by normalization. Now if

$$\int dx\ a(y, x)p(x) - p(y) = \int dx\ [a(y, x)p(x) - a(x, y)p(y)] = 0, \tag{7}$$

then $p(y)$ will not change under the transition matrix. Accordingly, for b^t to be a fixed point of this transition matrix it suffices to have detailed balance: $\forall x, y$,

$$a(y, x)\ b^t(x) = a(x, y)\ b^t(y) \tag{8}$$

If both T and π are nowhere zero, in light of Eq. 6, this means that b^t must equal π. (These conditions can be weakened, but they suffice for this synopsis of MH.) So for such T and π there is a fixed point of a at π, as desired.[2]

Say we allow T and therefore a to update stochastically from earlier elements of the random walk, and write a_t for the transition matrix at time t. Assume each a_t has only one fixed point, which for all t is the same distribution π. (The difference among the a_t is what their other eigenfunctions are.) For exaple, this is the case if each a_t is generated as in the MH algorithm, just from different associated proposal distributions, T^t.

Since the application of any such a_t to any distribution maps it geometrically closer to π, the application of any sequence of such a_t must iteratively map the distribution closer and closer to π. In other words,

[2] The uniqueness of that fixed point holds so long as a is irreducible and acyclic [?].

the approximation of Eq. 3 still holds if each a is replaced by a a_t, no matter how the sequence of $\{a_t\}$ are determined. (Intuitively, the distance from the distribution at time t to π is a Lyaponov function.)

Unfortunately, this modification of Eq. 3 does not give us the distribution of the n'th point in the random walk when the a^t are set adaptively from the elements in the random walk itself. For that to be the case we would need

$$\begin{aligned}\pi(x(n)) \;\approx\; & \int dx(1)dx(2)\ldots dx(n-1)\; da_2 da_3 \ldots da_n \\ & \prod_{t=2}^{n} P(a_t \mid x(1), \ldots, x(t-1))\; a_t(x(t), x(t-1)) \\ & \qquad\qquad \times\, \delta(x(1), x'). \end{aligned} \tag{9}$$

However because the weight of each integration variable $x(t' < t)$ in the integral is "warped" by the $P(a_t \mid x(1), \ldots, x(t-1))$ terms, integrating over it is no longer equivalent to the application of a matrix $a_{t'}(x(t'+1), x(t'))$ to a vector $b^{t'}(x(t'))$. So the fact that such a matrix multiplication maps $b^{t'}$ closer to π provides no assurances concerning our random walk.

We can circumvent this by having $\{a_t\}$ determined ahead of time, from a previous random walk which had a fixed transition matrix. This is the approach we adopt here.

3 Our AMH algorithm

3.1 General considerations

As mentioned above, our AMH algorithm is based on using the random walk to form increasingly accurate estimates of π and then updating T^t accordingly, i.e., it is a particular choice of $P(T^t \mid x(t))$. There are a number of subtleties one should account for in making this choice.

In practice there is almost always substantial discrepancy between π and q, since $\mathcal{Q}$ is a small subset of the set of all possible π. This means that setting $T(x, y) = q(y)$ typically results in frequent rejections of the sample points. The usual way around this problem in conventional MH (where T is fixed before the Markov process starts) is to restrict $T(x, y)$ so that x and y must be close to one another. Intuitively, this means that once the walk finds an x with high $\pi(x)$, the y's proposed by $T(x, y)$ will also have reasonable high probability (assuming π is not too jagged). We integrate this approach into our AMH algorithm by setting $T(x, y)$ to be $q(y)$ "masked" to force y to be close to x.

Another important issue is that the earlier a point is on the random walk, the worse it serves as a sample of π. To account for this, one shouldn't form $q_i(x_i = s)$ at time n simply as the fraction of the points for which $x_i(t < n) = s$. Instead we form those estimates by geometrically aging the points in forming q. This means that more recent points have more of an effect on our estimate of π. This aging has the additional advantage that it makes the evolution of τ a relatively low-dimensional Markov process, which intuitively should help speed convergence.

In [3, 2, 4] related ideas of how to exploit online-approximations of π that are generated from the random walk were explored. None of that work explicitly considers information-theoretic measures of distance (like KL distance) from the approximation to π. Nor is there any concern to "mask" the estimate of π in that work. The algorithms considered in that work also make no attempt to account for the fact that the early $x(t)$ should be discounted relative to the later ones. In addition, not using product distributions, parallelization would not be as straightforward with these alternatives schemes.

3.2 Details of our algorithm

Our proposed algorithm consists of three successive phases: the first of these is the cooling phase and the third is the data collecting phase. In both of those phases, the conventional Metropolis-Hastings algorithm is used, i.e., there is no updating on the proposal distribution. The second phase is where the proposal distribution is adaptively updated. The details are presented below:

Let N be the number of components of x and q^t the estimate of π at the t'th step of the walk. We consider the following algorithm:

1. Set $T^t(x, y)$ to $q^t(y)$ masked so that y and x differ in only one component:

$$T^t(x, y) \;\propto\; \delta\left(\sum_{i=1}^{N} \delta(x_i - y_i) - N + 1\right) \prod_{k=1}^{N} q_i^t(y_i) \; . \tag{10}$$

2. As in conventional MH, sample $[0, 1]$ uniformly to produce a r and set

$$x(t+1) = \begin{cases} y, & \text{if } r \leq R^t(x(t), y) \\ x(t), & \text{otherwise} \end{cases} \tag{11}$$

where

$$R^t(x, y) = \min\left\{1 \; , \; \frac{\pi(y) T^t(y, x)}{\pi(x) T^t(x, y)}\right\} . \tag{12}$$

3. *Only in phase 2:*
   Periodically update q. If $\mathrm{mod}_N(t+1) = 0$, then update the set $\{q_i^t\}$ by the non-negative multiplier $\alpha < 1$:
   For all i, x'_i, if $x'_i = x_i(t)$
   $$q_i^{t+1}(x'_i) = \alpha(q_i^t(x'_i) - 1) + 1 \tag{13}$$
   otherwise
   $$q_i^{t+1}(x'_i) = \alpha q_i^t(x'_i) \tag{14}$$
   If $\mathrm{mod}_N(t+1) \neq 0$, then $q_i^{t+1}(x'_i) = q_i^t(x'_i)$. To avoid *freezing* the proposal distribution, q_i is not allowed to get too close to the boundary of the probability simplex (i.e., less than $0.2 \times$ the initial uniform distribution).
4. $t \leftarrow t+1$. Repeat from step 1.

We note again that in the the first phase of the algorithm, T is uniform and step 3 is not implemented, in the second phase, all steps above are implemented and in the third phase, step 3 is not implemented. We also note that our updating method depend only on the current state and hence is much more efficient than previously proposed methods [2].

4 Experiments

4.1 Sampling Experiments

Currently there is no consensus on how to quantify "how close" a set $\{x(t)\}$ is to an IID sample of π. One approach is to input the set into a density estimation algorithm [5]. One can then use KL distance from that estimated distribution to π as the desired quantification. This can be problematic in high-dimensional spaces though, where the choice of density estimation algorithm would be crucial. However say we have a contractive mapping $F : x \in X \rightarrow y \in Y$ where Y is a low-dimensional space that captures those aspects of X that are of most interest. We can apply F to the $\{x(t)\}$ to produce its image in Y, $\{y(t)\}$. Next one can apply something as simple and (relatively) unobjectionable as histogramming to do the density estimation translating $\{y(t)\}$ to an associated estimate of the generating distribution over Y. We can then use KL distance between that histogram and $F(\pi)$ as the desired quantification of how good our transition matrix is. This is the approach we took here.

Another point one has to concern with is the existence of Kullback-Leibler (KL) divergence which plagues almost all Monte Carlo generating distribution. This corresponds to the situation where no samples are obtained in region where π is non-zero. This is not a serious problem if the total probability, ϵ, associated to KL divergences is negligible because the discrepany obtained on any expected value calculations will be bound by ϵ. Figure 2 is devoted to gauging this effect.

Our first experiment concerns the Ising spin-glass model:

$$H(x) = \frac{1}{2} \sum_{<i,j>} J_{ij} x_i x_j + \sum_i h_i x_i \tag{15}$$

where $< i, j >$ denotes summation over all neighbours. In this function the J_{ij} and h_i are randomly generated integers in the interval $[-5\ ,\ 5]$ and the x_i can take on values -1 and 1. Our task is to sample the associated Boltzmann distribution:

$$\pi(x) \propto \exp(-H(x)/T) \tag{16}$$

where T corresponds to the temperature in a thermodynamic setting. We have chosen spin-glasses for illustration because it is generally believed that they display salient features of complex disordered systems [10]. (Indeed, searching for spin-glass ground states is a NP-complete problem.)

We have performed experiments on spin-glasses in a 1D ring formation (with 50, 75 and 100 spins shown in Figure 1). In these experiments, we firstly run, with random initial states, 5 long Markov chains (800,000$\times N$ steps where N is the number of spins and data are collected at the last quarter of chain) with the conventional MH algorithm. We then average the energy distributions obtained to form our *target* distribution. Its closeness to the true distribution is suggested by smallest of the KL distances of the original distributions (the bottom three lines in Figure 1).

We then produced 100 samples of energy distributions with the MH and the adaptive MH methods, with chains of 40,000$\times N$ steps each. We note that in the adaptive MH method, $q_i(x_i)$ corresponds to the the probability of spin i being in state x_i. Data are again collected in the last quarter of each chain in both case, and we performed proposal distribution updates as detailed before in the third quarter of the chains in the adaptive M-H case (with the updating parameter $\alpha =$ 0.98).

Figures 1 and 2 show the results of these experiments with the error bars being the errors on the means. We see that AMH (with $\circ$ markers) outperforms conventional MH (with * markers) in sampling, as well as in avoiding KL divergence. Similar experiments on a 2D lattice have also been performed and the adaptive M-H shows similar superior performance over conventional MH.

4.2 Optimization Experiments

We now turn to using our algorithm to the problem of optimization. We consider the same problem as before with 100 spins. In the simulation, we randomly generate 20 different sets of $\{J, h\}$ for the hamiltonian in eq. 15. The temperature is set to go from 1 to 0.05 in 19 equal steps. We produce 50 samples each for the MH and AMH versions of the algorithm and the results are presented in Fig. 3..

5 Conclusion

With the product distribution assumption, we have proposed a new adaptive Metropolis-Hastings which is easy to implement and we have shown its superiority over conventional Metropolis-Hastings with computer experiments. Compared with adaptive Metropolis-Hastings proposals [2, 3, 4], we have demonstrated the usefulness of our proposed algorithm with highly non-trivial examples, i.e., spin-glasses, which highlights the usefulness of our proposed algorithm for sampling complex distribution. With annealing in temperature, our method is also shown to be useful in hard optimization.

Besides, the q produced by AMH has many uses beyond improved sampling. It can be used as an estimate of the marginals of π, i.e., as an estimate of the optimal mean-field approximation to π. Because they are product distributions, the successive q^t can also be used as the control settings in adaptive distributed control [8, 11]. (In this application $\{x(t)\}$ is the sequence of control variable states and π is log of the objective function.) It's being a product distribution also means that the final q can be used to help find the bounded rational equilibria of a non-cooperative game with shared utility functions [7].

Acknowledgements: We thank Bill Macready for stimulating discussion. C.F.L. thanks NASA, University College (Oxford) and NSERC (Canada) for finanical support.

References

[1] M. West, Computing Sciences and Statistics **24**, 325 (1993).

[2] C. Sims, unpublished (1998).

[3] J. Gasemyr, Scandinavian Journal of Statistics, **30**, 159 (2003).

[4] J.N. Corcoran and U. Schnieder, unpublished (2004).

[5] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification* (2nd Ed., Wiley and Sons, New York, 2000).

[6] T. Cover and J. Thomas, *Elements of Information Theory* (Wiley-Interscience, New York, 1991).

[7] D.H. Wolpert, in *Complex Engineering Systems*, A.M.D. Braha and Y. Bar-Yam (eds) (2004).

[8] D.H. Wolpert and S. Bieniawski, in *Proceedings of CDC'04* (2004).

[9] W. Macready and D.H. Wolpert, submitted to ICCS04.

[10] S. Kauffman and S. Levin, J. Theor. Biol. **128**, 11 (1987); E. Weinberger, Phys. Rev. A **44**, 6399 (1991); K.H. Fisher and J.A. Hertz, *Spin glasses* (Cambridge University Press, Cambridge, 1991).

[11] C.F. Lee and D.H. Wolpert, in *Proceedings of the Third International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS 2004)* (IEEE Press, 2004).

[12] D.H. Wolpert, Factoring a Canonical Ensemble, cond-mat/0307630, 2003.

Figure 1: (* = MH, $\circ$ = AMH, + = MH with long chains. The error bars are errors on the means.)

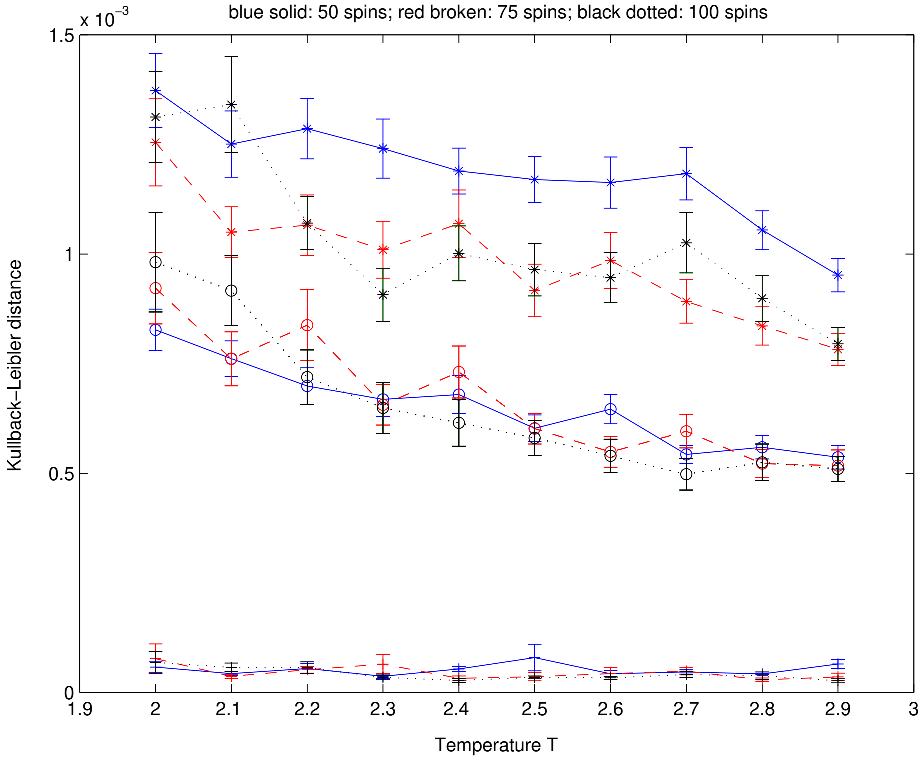

Figure 2: (* = MH, ∘ = AMH. The error bars are errors on the means.)

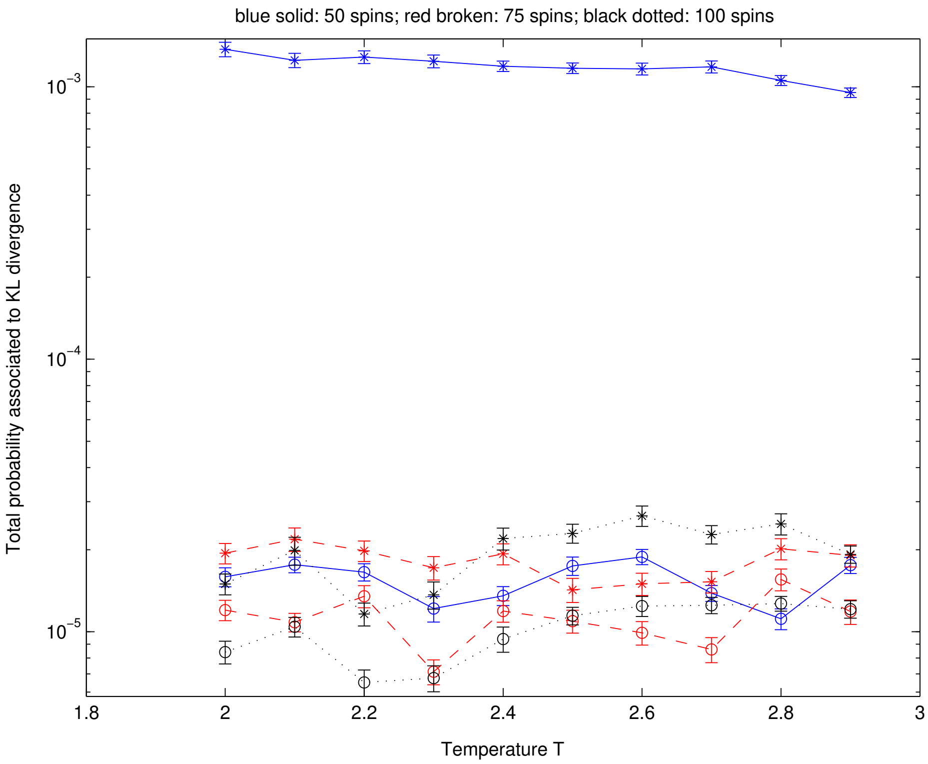

Figure 3: Results of 20 different spin-glasses. We note constants are added to the y-axis so that the minimum energies found by AMH are zero. ($*$ = MH, $\circ$ = AMH. The error bars are errors on the means.)