JMLR: Workshop and Conference Proceedings vol 35:1–18, 2014

Near-Optimal Herding

Nick Harvey NICKHAR@CS.UBC.CA
University of British Columbia, Department of Computer Science

Samira Samadi SAMIRASA@CS.UBC.CA
University of British Columbia, Department of Computer Science

Abstract

Herding is an algorithm of recent interest in the machine learning community, motivated by inference in Markov random fields. It solves the following sampling problem: given a set $\mathcal{X} \subset \mathbb{R}^d$ with mean μ, construct an infinite sequence of points from $\mathcal{X}$ such that, for every $t \geq 1$, the mean of the first t points in that sequence lies within Euclidean distance $O(1/t)$ of μ. The classic Perceptron boundedness theorem implies that such a result actually holds for a wide class of algorithms, although the factors suppressed by the $O(1/t)$ notation are exponential in d. Thus, to establish a non-trivial result for the sampling problem, one must carefully analyze the factors suppressed by the $O(1/t)$ error bound.

This paper studies the best error that can be achieved for the sampling problem. Known analysis of the Herding algorithm give an error bound that depends on geometric properties of $\mathcal{X}$ but, even under favorable conditions, this bound depends linearly on d. We present a new polynomial-time algorithm that solves the sampling problem with error $O\big(\sqrt{d}\log^{2.5}|\mathcal{X}|/t\big)$ assuming that $\mathcal{X}$ is finite. Our algorithm is based on recent algorithmic results in *discrepancy theory*. We also show that any algorithm for the sampling problem must have error $\Omega(\sqrt{d}/t)$. This implies that our algorithm is optimal to within logarithmic factors.

Keywords: Herding, discrepancy theory, deterministic sampling methods

1. Introduction

Herding is a topic that has attracted much interest in the machine learning community over the past few years. It was originally proposed (Welling (2009)) as a method for the “sampling” from a Markov random field that agrees with a given data set. The traditional approach for this scenario is to use maximum likelihood estimation to infer parameters of the model, and then to sample from that model. A drawback of the traditional approach is that the maximum likelihood estimation step is computationally intractable in general. The Herding approach sidesteps this intractability problem — it does not infer the model parameters, but instead deterministically produces a sequence of “samples” whose moments rapidly converge to the moments of the given data set.

The Herding algorithm has several attractive properties. Computationally it is very simple — it is a greedy algorithm whose pseudocode is only two lines. Yet despite its simplicity, it has a dignified lineage. Herding is related to the Perceptron method (Gelfand et al. (2010)), to conditional gradient methods (Bach et al. (2012)), and to quadrature methods (Bach et al. (2012); Huszár and Duvenaud (2012)). Herding can naturally be extended to a kernel method (Chen et al. (2010)).

Finally, an intriguing property is that the samples produced by Herding converge *faster* than independent random samples, in the sense that the error after t Herding samples is inversely proportional to t, whereas the error of t random samples is inversely proportional to $\sqrt{t}$ (Chen et al. (2010)). Due to this intriguing property, recent work has studied using Herding ideas to improve convergence of the Gibbs sampler (Bornn et al. (2013)); this relates to interesting directions in combinatorics (Holroyd and Propp (2010)) and Markov chain Monte Carlo methods (Chen et al. (2011)).

The purpose of this paper is to rigorously analyze the theoretical underpinnings of the Herding algorithm. In particular, since Herding's convergence properties have received much attention, we wish to determine the best convergence achievable by *any* algorithm. One of our main observations is that there is a connection between the sampling question and *discrepancy theory*, an important topic in combinatorics and geometry. We will exploit this connection in several parts of our results.

To begin, we present a fairly simple lower bound on the convergence rate of any algorithm for the sampling question (see Section 4). We then observe that this lower bound is actually optimal if a classic conjecture in discrepancy theory is true. Next, we present a partial result towards this conjecture which is algorithmic and uses recent breakthroughs in discrepancy theory (Bansal (2010); Lovett and Meka (2012)). This yields a new, polynomial-time algorithm to produce a sequence of samples whose convergence rate is within a logarithmic factor of optimal. In contrast, the best known upper bound on the convergence of the Herding algorithm is significantly worse, even under generous assumptions. We conjecture that the actual convergence of the Herding algorithm is strictly worse than the convergence of our new algorithm, and we prove that this is true if convergence is measured in the ℓ_∞-norm.

1.1. Outline

The rest of this paper is organized as follows. In Section 2, we present formal definitions and briefly explain our main results. In Section 2.2, we present a brief overview of discrepancy theory which is necessary to explain our results in detail. In Section 3, we present our main algorithm and analyze its convergence rate. In Section 4, we provide the detail of our lower bound proof for Sampling Problem. In Section 5, we conclude our work and discuss future directions.

2. Formal Definitions and Main Results

The following notation will be used throughout the paper. We write $[n]$ for the set $\{1, 2, ..., n\}$. For any matrix M we let $M_{i,*}$ refer to the i^{th} row of M, and $M_{*,j}$ refer to the j^{th} column of M. All logarithms are in base 2. The ℓ_p-norm is denoted $\|\cdot\|_p$. The i^{th} standard basis vector is denoted e_i.

2.1. Sampling Problem

Let $\mathcal{X} \subset \mathbb{R}^d$ be a finite set with $|\mathcal{X}| = n$. Let $\mu = \sum_{x\in\mathcal{X}} x/n$ and $\delta = \max_{x\in\mathcal{X}} \|x\|_2$. More generally, one could assume that $\mathcal{X}$ is infinite and that it lies in an infinite-dimensional Hilbert space. In this paper we will restrict to the case of finite $\mathcal{X}$ and finite dimensions.

Definition 1 (Sampling Problem) *We wish to find an infinite sequence of points $x_1, x_2, ...$ from the set $\mathcal{X}$. Each x_i is called a pseudosample. The error of the first t pseudosamples is $\|\sum_{i\le t} x_i/t - \mu\|_2$.*

The goal is to ensure that, for all $t \geq 1$, the error of the first t pseudosamples is at most α/t, where α is a quantity that can depend arbitrarily on d, n and $\mathcal{X}$.

A concise statement of the Herding algorithm is as follows:

$$x_{t+1} \in \operatorname*{arg\,max}_{x \in \mathcal{X}} \langle w_t, x \rangle \tag{1a}$$

$$w_{t+1} = w_t + \mu - x_{t+1} \tag{1b}$$

The initial weight vector w_0 is initialized somewhat arbitrarily; the literature has proposed setting $w_0 = 0$ or $w_0 = \mu$ (Chen et al. (2010); Bach et al. (2012)). We can also assume[1] that Herding algorithm choses the smallest index in the set $\{\arg\max_{x \in \mathcal{X}} \langle w_t, x \rangle\}$. The prior analysis of this algorithm can be stated as follows. Let r be the maximum value such that the ℓ_2-ball centered at μ of radius r is contained in the convex hull of $\mathcal{X}$. It has been shown[2] that the error of the Herding algorithm is $O\big((\|w_0\|_2 + \delta^2/r)/t\big)$ (Chen et al. (2010)).

Initially it may seem intriguing that the error is proportional to $1/t$, but this is an instance of a much more general phenomenon. It was observed (Gelfand et al. (2010)) that if (1a) is replaced with any rule that ensures $\langle w_t, x_{t+1} \rangle \geq 0$, then the Perceptron Boundedness Theorem (Block and Levin (1970)) implies that the error is at most α/t for some quantity α. However, the proof of Block and Levin uses compactness and does not give any quantitative bound on α. Giving an effective bound on α was an open question for decades, until recently (Amaldi and Hauser (2005)) proved a concise bound that is exponential in d.

Thus, the interesting aspect of the Sampling Problem is not that the error is proportional to $1/t$, but rather determining a precise value of α such that the error is at most α/t. The purpose of this paper is to address the following question, which seems fundamental.

Question 2 *What is the best value of α (as a function of d, n or $\mathcal{X}$) that can be obtained in the Sampling Problem?*

The analysis of the Herding algorithm gives a bound on α that depends on r, and hence on the geometry of $\mathcal{X}$. An undesirable aspect of this bound is that it does not reflect the *intrinsic* geometry: it is highly sensitive to rescalings of $\mathcal{X}$. Imagine multiplying the first coordinate of all points in $\mathcal{X}$ by a quantity ϵ and keeping other coordinates unchanged — if the coordinates represent features of the data, then this amounts to simply changing the units of the first feature. One can see that, as $\epsilon \to 0$, r decreases by a factor of roughly ϵ, whereas δ remains mostly unchanged. Thus, such scalings can arbitrarily increase the ratio δ/r, and hence the error bound of the Herding algorithm.

For any convex body $\mathcal{K}$, there is a canonical transformation that negates the effects of such unimportant rescalings. It is standard terminology to say that $\mathcal{K}$ is in *John's position* if the largest ellipsoid contained in $\mathcal{K}$ is a scalar multiple of the unit Euclidean ball (see Ball (1997); Vershynin (2009)). So consider $\mathcal{K}$ to be the convex hull of $\mathcal{X}$, and assume that the center of $\mathcal{K}$ is μ. Then the quantity r defined above is the radius of the largest ball inside $\mathcal{K}$. John's theorem asserts that $\mathcal{K}$ is

1. This assumption is used in the proof of Theorem 5.
2. The bound claimed in that paper appears to be inaccurate. In Appendix C, we prove that the bound that we state here is correct.

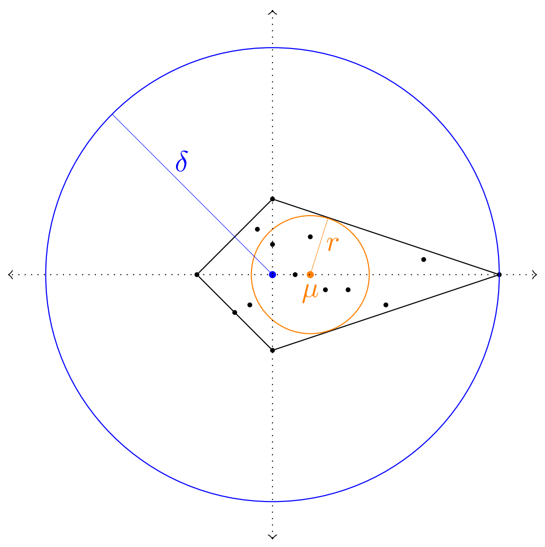

Figure 1: The black colour shows the set $\mathcal{X}$ and its convex hull $\mathcal{K}$. The blue colour shows the smallest circle centred at the origin that contains $\mathcal{K}$ and its radius δ. The orange colour shows the biggest circle centred at μ that is contained in $\mathcal{K}$ and its radius r.

contained in the ball centered at μ of radius rd. Assuming that $\mu = 0$, it follows that the quantity δ defined above is at most rd, and hence[3] $\delta/r \leq d$. Thus, under these assumptions, the error of the Herding algorithm is

$$O\big((\|w_0\|_2 + \delta d)/t\big). \tag{2}$$

Instead of simply assuming that $\mathcal{X}$ is in John's position, one could imagine performing a preprocessing step to transform $\mathcal{X}$ to John's position, then running the Herding algorithm in the transformed space. This would yield a 2-norm error bound in transformed space. However, translating back to the original space, one would obtain an error bound in a different norm. It is unclear whether such an approach would provide a desirable 2-norm error bound in the original space.

In this paper, we are interested in bounds on α that essentially depend only on d, and are not disrupted by unimportant rescalings. This suggests two natural questions. First, does a bound depending only on d hold without assuming that the set $\mathcal{X}$ has been transformed to some canonical position? Second, is the linear dependence on d in (2) optimal? Our first, and rather simple, observation is:

Theorem 3 *For each d, there exists a set $\mathcal{X} \subset \mathbb{R}^d$ with $\max_{x\in\mathcal{X}} \|x\|_2 = 1$ such that every solution to the Sampling Problem on $\mathcal{X}$ has error $\Omega(\sqrt{d}/t)$ for $t = \Theta(d^2)$.*

If we allow error bounds that depend on n, then we can show a nearly-matching upper bound.

Theorem 4 *There is an algorithm with running time $\operatorname{poly}(d, n)$ that solves the Sampling Problem with error $O(\sqrt{d}\delta \log^{2.5}(n)/t)$.*

3. This bound is tight: a simplex centered at the origin has δ/r exactly equal to d.

Theorem 3 is proven in Section 4 and Theorem 4 is proven in Section 3. We do not know whether the algorithm of Theorem 4 has strictly better error than the Herding algorithm. However, if we measure error in the ℓ_∞-norm then our algorithm is seen to be strictly better.

Theorem 5 *For each d, we can construct a set $\mathcal{X} \in \mathbb{R}^d$ such that the Herding algorithm has $\max_t \|t \cdot (\sum_{i=1}^t x_i/t - \mu)\|_\infty = \Omega(\sqrt{d})$. On the other hand, the algorithm of Theorem 4 has $\max_t \|t \cdot (\sum_{i=1}^t x_i/t - \mu)\|_\infty = O(\log^{2.5}(n))$.*

Theorem 5 is proven in Appendix A. We now begin discussing the proof of Theorem 4. In order to do so, let us introduce a more stringent problem.

Definition 6 (Permutation Problem) *Let $\mathcal{X}$, μ, d and n be as before. We wish to find a bijection $\pi : \{1, ..., n\} \to \mathcal{X}$. The error of the first t points is $\|\sum_{i \le t} \pi(i)/t - \mu\|_2$. The goal is to ensure that, for all $t \geq 1$, the error of the first t points is at most α/t*

It is easy to see that any solution to the Permutation Problem yields a solution to the Sampling Problem with the same parameter α. To see this, consider the infinite sequence obtained by concatenating copies of $\mathcal{X}$, each ordered by π. Every contiguous subsequence of length n sums to $n\mu$. So, for every length-t prefix of this infinite sequence, only the last $t \bmod n$ vectors contribute to the error. The error of those last vectors is exactly $\|\sum_{i=1}^{t \bmod n} (\pi(i) - \mu)/t\|_2$, which is assumed to be at most α/t.

Thus, to give an upper bound on the error in the Sampling Problem, it suffices to give an upper bound on the error in the Permutation Problem. This is the approach used to prove Theorem 4. To explain this approach in more detail, we must introduce tools from discrepancy theory.

2.2. Discrepancy Theory

Discrepancy theory is a major area of study in both combinatorics and geometry (Chazelle (2000); Matousek (1999)). The powerful results in this area have applications in many theoretical areas of computer science. We begin by mentioning one of the oldest results in this area Steinitz (1913).

Fix a convex body $B \subset \mathbb{R}^d$ that is 0-symmetric (i.e., $B = -B$) For any finite set $V \subset B$ with $\sum_{v \in V} v = 0$, let β_V be the smallest value for which there is an ordering $v_1, v_2, ...$ of V with $\sum_{i \le t} v_i \in \beta_V \cdot B$ for all t. The *Steinitz constant* of B, denoted $S(B)$, is the supremum of β_V over all finite $V \subset B$. Note that $S(B)$ depends only on d and B, and is not a function of $|V|$.

The connection between the Permutation Problem and discrepancy theory is now apparent. Let B_2^d be the Euclidean unit ball in $\mathbb{R}^d$. Let $\mathcal{X}'$ be $\mathcal{X}$ translated so that its centroid μ is the origin, and scaled so that $\mathcal{X} \subset B_2^d$. There is an ordering $x_1, x_2, ...$ of $\mathcal{X}'$ such that $\sum_{i \le t} x_i \in S(B_2^d) \cdot B_2^d$ for all t. This gives a solution to the Permutation Problem on $\mathcal{X}$ with $\alpha = S(B_2^d) \cdot \max_{x \in \mathcal{X}} \|x\|_2$.

It is conjectured that $S(B_2^d) = O(\sqrt{d})$ (see Bergström (1931); Bařaňy (2008)). If true, that would imply a solution to the Permutation Problem, and hence to the Sampling Problem, with $\alpha = O(\sqrt{d}) \cdot \max_{x \in \mathcal{X}} \|x\|_2$. This would match our lower bound in Section 4. The best known bound on $S(B_2^d)$ is the Steinitz Lemma: $S(B_2^d) \le d$ (Bařaňy (2008)). Furthermore, the proof is algorithmic, so this gives an efficient solution to the Sampling Problem with $\alpha = d \cdot \max_{x \in \mathcal{X}} \|x\|_2$.

This result is of mild interest, in that it answers the question raised in the previous section on whether there is a solution with error linear in d *without* the various assumptions on $\mathcal{K}$.

Our next result is as follows:

Theorem 7 *Let $V \subset \mathbb{R}^d$ satisfy $\max_{v \in V} \|v\|_2 \le 1$ and $\sum_{v \in V} v = 0$. Let $n = |V|$. Then there is an ordering $v_1, v_2, \ldots$ of V such that $\|\sum_{i \le t} v_i\|_2 = O(\sqrt{d} \log^{2.5} n)$ for all $t \ge 1$. Furthermore, there is an algorithm with running time $\operatorname{poly}(d, n)$ to find such an ordering.*

Due to the connection from the Sampling Problem to the Permutation Problem described above, Theorem 7 implies a solution to Theorem 4. It does not imply anything about the Steinitz constant $S(B_2^d)$ because that quantity must not depend on n. To explain the proof of Theorem 7, we must introduce some further definitions and notation (Banaszczyk (2012)).

Let $B \subset \mathbb{R}^d$ be a full-dimensional convex body that is 0-symmetric. Let $B' \subset \mathbb{R}^d$ be bounded. For $n \in \mathbb{N}$, we define $\mathbf{sv}(B, B'; n)$, $\mathbf{ss}(B, B'; n)$ and $\mathbf{st}(B, B'; n)$ as follows:

- $\mathbf{sv}(B, B'; n)$ is the smallest $\lambda > 0$ such that for all $x_1, \ldots, x_n \in B'$ there exist $\epsilon_1, \ldots, \epsilon_n \in \{-1, 1\}$ such that $\sum_{i=1}^n \epsilon_i x_i \in \lambda B$. The symbol $\mathbf{sv}$ is an abbreviation of "signed vectors".
- $\mathbf{ss}(B, B'; n)$ is the smallest $\lambda > 0$ such that for all $x_1, \ldots, x_n \in B'$ there exist $\epsilon_1, \ldots, \epsilon_n \in \{-1, 1\}$ such that $\sum_{i=1}^k \epsilon_i x_i \in \lambda B$ for all $k \in [n]$. The symbol $\mathbf{ss}$ is an abbreviation of "signed series".
- $\mathbf{st}(B, B'; n)$ is the smallest $\lambda > 0$ such that for all $x_1, \ldots, x_n \in B'$ with $\sum_{i=1}^n x_i = 0$ there exists permutation π of $[n]$ such that $\sum_{i=1}^k x_{\pi(i)} \in \lambda B$ for all $k \in [n]$. The symbol $\mathbf{st}$ is an abbreviation of "Steinitz constant".

Let us now state some results using this notation. Let B_p be the unit ball of the ℓ_p-norm in $\mathbb{R}^d$. Theorem 7 states that $\mathbf{st}(B_2, B_2; n) \le O(\sqrt{d} \log^{2.5} n)$. It is known that $\mathbf{sv}(B_\infty, B_\infty; n) \le O(\sqrt{d \log{(2n/d)}})$ and that $\mathbf{sv}(B_\infty, B_1; n) \le 2$. These results can be found in standard references (Alon and Spencer (2000)). It is also known that $\mathbf{sv}(B_\infty, B_2; n) \le O(\sqrt{\log d})$ (Banaszczyk (1998)). The central open question in this area is Komlós' conjecture that $\mathbf{sv}(B_\infty, B_2; n) \le O(1)$ (Spencer (1985)).

Algorithmic proofs are known for some of the results mentioned above. Recent algorithmic breakthroughs proved that $\mathbf{sv}(B_\infty, B_\infty; n) \le O(\sqrt{d \log(2n/d)})$ (Bansal (2010); Lovett and Meka (2012)). These techniques also algorithmically prove $\mathbf{sv}(B_\infty, B_2; n) \le O(\log d)$, which is slightly worse than Banaszczyk's result. The core of these algorithmic results is the following lemma (Lovett and Meka (2012)).

Lemma 8 (Partial Coloring Lemma) *Let $v_1, \ldots, v_d \in \mathbb{R}^n$ be vectors and let $x_0 \in [-1, 1]^n$. Let $c_1, \ldots, c_d \ge 0$ be scalars such that $\sum_{i=1}^d \exp(-c_i{}^2/16) \le n/16$. Let $\epsilon > 0$. Then there exists a randomized algorithm which with probability at least 0.1 finds a point $x \in [-1, 1]^n$ such that*

(i): $|\langle x - x_0, v_i \rangle| \le c_i \|v_i\|_2$ *for all $i \in [d]$*

(ii): $|x_j| \ge 1 - \epsilon$ *for at least $n/2$ indices $j \in [n]$.*

The running time of the algorithm is $\operatorname{poly}(n, d, 1/\epsilon)$.

We will also use this lemma as a key ingredient in our proof in Section 3.1.

3. Main Algorithmic Result

3.1. Bounding ss

The first step to proving Theorem 7 is to bound $\mathbf{ss}(B_\infty, B_2; n)$. Theorem 9 constructively proves that $\mathbf{ss}(B_\infty, B_2; n) \leq O(\log^{2.5} n)$. The vector χ appearing in this theorem is called a "coloring".

Theorem 9 *Let V be a matrix of size $d \times n$ such that $\max_j \|V_{*,j}\|_2 \leq \delta$. Then there exists $\chi \in \{-1,1\}^n$ such that $\|\sum_{j=1}^k V_{*,j}\chi_j\|_\infty \leq O(\delta \log^{2.5} n)$ for all $k \in [n]$. Furthermore, there exists a randomized algorithm to compute the coloring χ in time $\mathrm{poly}(n,d)$.*

Proof sketch. The partial coloring lemma can be used iteratively to control the discrepancy of the rows of matrix V; this is the standard technique of Spencer used to bound $\mathbf{sv}(B_\infty, B_2; n)$. However, it does not suffice to bound $\mathbf{ss}(B_\infty, B_2; n)$. We must additionally control the discrepancy of all *prefixes* of rows of V. The idea is to construct from V a new matrix W by adding rows that indirectly control the prefixes of rows of V. Some care is required to ensure that δ does not increase dramatically. Finally, we iteratively apply the partial coloring lemma to W to obtain the desired coloring.

Proof The proof has several steps:

Construction of W. We construct a new matrix W from V by adding new rows which are "sub-rows" of the rows of V. Roughly speaking, for every row of V, we make many new copies of that row. Each copy has most of its entries zeroed out, except those in a contiguous region of length 2^k that ends at a multiple of 2^k.

More formally, define $\mathcal{I} := \{\ [j \cdot 2^k + 1, (j+1)\cdot 2^k] \cap [n] \ :\ j \geq 0, k \geq 0\ \}$, and note that $|\mathcal{I}| \leq 2n-1$. We may enumerate the entries of $\mathcal{I}$ as $I_1, I_2, \ldots$. For each $i \in [d]$ and for every $I_a \in \mathcal{I}$, let W^i be a matrix of size $|\mathcal{I}| \times n$, constructed from the row $V_{i,*}$ as follows.

$$\forall a \in \{1, \ldots, |\mathcal{I}|\} \qquad\qquad W^i_{a,l} = \begin{cases} V_{i,l} & \text{if } l \in I_a \\ 0 & \text{otherwise} \end{cases}$$

Note that $V_{i,*}$ itself is one of the rows of W^i. Finally, the matrix W is constructed by "stacking" the matrices W^i on top of each other. That is, the rows of W are precisely the set of rows that appears in any W^i. So the matrix W has size $d' \times n$, where $d' = d \cdot |\mathcal{I}| = O(nd)$.

Let δ' be $\max_j \|W_{*,j}\|_2$, the largest 2-norm of any column of W. We claim that $\delta' \leq \delta \cdot \sqrt{\lceil \log n \rceil}$. To see this, note that for every $l \in [n]$, we have $|\{\ I \in \mathcal{I} \ :\ l \in I\ \}| \leq \lceil \log n \rceil$. Thus, each entry $V_{i,j}$ appear in at most $\lceil \log n \rceil$ of the copies of row $V_{i,*}$.

Applying the partial coloring lemma. The next step of the proof is to apply the partial coloring lemma to the matrix W. We set the parameter $c_i = \delta' / \|W_{i,*}\|_2$, where $C = 32$ and δ' is the maximum 2-norm of any column of W.

We now check the hypotheses of the partial coloring lemma. The argument is similar to existing arguments (Bansal (2010); Lovett and Meka (2012)). We have

$$\sum_{i=1}^{d'} \|W_{i,*}\|_2^2 \;=\; \sum_{j=1}^{n} \|W_{*,j}\|_2^2 \;\leq\; n\delta'^2.$$

In particular, there are at most $n/2^r$ rows $W_{i,*}$ with $\|W_{i,*}\|_2^2 \in [2^r\delta'^2, 2^{r+1}\delta'^2]$. So

$$\sum_{i=1}^{d'} \exp\left(\frac{-c_i{}^2}{16}\right) \;<\; \sum_{r\in\mathbb{Z}} \frac{n}{2^r} \exp\left(\frac{-C^2}{2^{r+5}}\right). \tag{3}$$

We prove in Appendix B that the right-hand side (3) is at most $n/16$ if $C = 32$. This establishes the hypotheses of the partial coloring lemma.

Iteratively applying the partial coloring lemma. This step is identical to a previous result (Bansal (2010); Lovett and Meka (2012)). The idea is that a partial coloring ensures that at least half of the coordinates are very close to either 1 or -1. If we recurse on the remaining coordinates we can obtain a full coloring. The depth of the recursion is only $\lceil \log n \rceil$ since the number of remaining coordinates halves in each step. Each application of the partial coloring lemma increases the discrepancy of row $W_{i,*}$ by at most $c_i \|W_{i,*}\|_2 = C\delta'$. The final vector resulting from this recursion is a vector $\chi \in [-1,1]^n$ satisfying $|\chi_j| \geq 1-\varepsilon$ for all $j \in [n]$, and:

$$|\langle\, W_{i,*},\, \chi \,\rangle| \;\leq\; C\delta' \cdot \lceil \log n \rceil \;=\; O(\delta' \log n) \qquad \forall i \in [d']. \tag{4}$$

For further explanation, see Lovett and Meka (2012).

Rounding. The next step is to produce an integral coloring. As in prior work, we simply round each coordinate of χ to either $+1$ or -1. The additional discrepancy introduced by this rounding can be easily bounded. Every entry $V_{i,j}$ satisfies $V_{i,j} \leq \|V_{*,j}\|_2 \leq \delta$. Increasing $|x_i|$ from $1-\varepsilon$ to 1 changes the discrepancy by at most $\varepsilon\delta n$. Choosing $\varepsilon \leq 1/n\delta$, the additional discrepancy produced by this rounding is at most 1.

Discrepancy of prefixes. Finally, we must show that every prefix of every row of V has low discrepancy under the coloring χ. Observe that any set $P = \{1, \ldots, k\}$, $k \leq n$, can be written as the disjoint union $P = \cup_{a\in S} I_a$ for some $S \subseteq [\,|\mathcal{I}|\,]$ with $|S| \leq O(\log n)$. Thus, for any $i \in [d]$,

$$\begin{aligned} |\textstyle\sum_{j=1}^{k} V_{i,j}\chi_j| \;&\leq\; \textstyle\sum_{a\in S} |\sum_{j\in I_a} V_{i,j}\chi_j| \\ &=\; \textstyle\sum_{a\in S} |\langle\, W^i_{a,*},\, \chi \,\rangle| \\ &\leq\; |S| \cdot O(\delta' \log n) \\ &=\; O(\delta' \log^2 n) \;=\; O(\delta \log^{2.5} n). \end{aligned}$$

This completes the proof. ∎

3.2. Relating ss and st

The previous section gave an algorithmic proof that $\mathbf{ss}(B_\infty, B_2; n) \leq O(\log^{2.5} n)$. The next step in proving Theorem 7 is to relate $\mathbf{ss}$ to $\mathbf{st}$; furthermore, the proof should be *constructive*. The Chobanyan theorem is a classic theorem in discrepancy that relates $\mathbf{ss}$ and $\mathbf{st}$. Unfortunately, the standard proof of this theorem (see Bařány (2008)) is not constructive. We derive an algorithmic version of the Chobanyan theorem, leading to the following result.

Theorem 10 *Let V be a real matrix of size $d \times n$ with $\sum_j V_{*,j} = 0$. Let $\delta = \max_j \|V_{*,j}\|_2$. Assume that for every permutation π of $[n]$ there exist $\epsilon_1, ..., \epsilon_n \in \{-1, 1\}$ (depending on π) such that $\max_{1\leq k\leq n} \|\sum_{j=1}^k \epsilon_j V_{*,\pi(j)}\|_\infty \leq A$. Furthermore, assume that there is an algorithm with running time $\mathrm{poly}(n, d)$ to compute the signs $\epsilon_1, ..., \epsilon_n$. Then there is an algorithm to construct a permutation π^* of $[n]$ such that $\max_{1\leq k\leq n} \|\sum_{j=1}^k V_{*,\pi^*(j)}\|_\infty \leq A + \delta$. This algorithm also has running time $\mathrm{poly}(n, d)$.*

Proof We describe an algorithm that outputs the desired permutation π^*. For each permutation π of $[n]$, define $\mathrm{disc}(\pi) = \max_{1\leq k\leq n} \|\sum_{j=1}^k V_{*,\pi(j)}\|_\infty$.

Consider an arbitrary permutation π_1 of $[n]$, and let $B_1 = \mathrm{disc}(\pi_1)$. If $B_1 \leq A$ then simply output π_1. Otherwise, by the hypothesis of the theorem, there exist $\epsilon^1_1, ..., \epsilon^1_n \in \{-1, 1\}$ such that $\max_{1\leq k\leq n} \|\sum_{j=1}^k \epsilon^1_j V_{*,\pi_1(j)}\|_\infty \leq A$. We construct permutation π_2 from π_1 as follows. Let

$$M^+ = \{\, j \in [n] \,:\, \epsilon^1_j = +1 \,\} \qquad \text{and} \qquad M^- = \{\, j \in [n] \,:\, \epsilon^1_j = -1 \,\}.$$

We have

$$\begin{aligned} \textstyle\sum_{j=1}^k V_{*,\pi_1(j)} + \sum_{j=1}^k \epsilon^1_j V_{*,\pi_1(j)} &= 2 \cdot \textstyle\sum_{j\in M^+\cap[k]} V_{*,\pi_1(j)} \\ \textstyle\sum_{j=1}^k V_{*,\pi_1(j)} - \sum_{j=1}^k \epsilon^1_j V_{*,\pi_1(j)} &= 2 \cdot \textstyle\sum_{j\in M^-\cap[k]} V_{*,\pi_1(j)}. \end{aligned}$$

Hence

$$\begin{aligned} \|\textstyle\sum_{j\in M^+\cap[k]} V_{*,\pi_1(j)}\|_\infty &\leq \frac{A + B_1}{2} \\ \|\textstyle\sum_{j\in M^-\cap[k]} V_{*,\pi_1(j)}\|_\infty &\leq \frac{A + B_1}{2}. \end{aligned}$$

The new permutation π_2 is formed by concatenating the elements of M^+, in the order given by π_1, followed by the elements of M^-, in the *reverse* of the order given by π_1. It follows from the assumption $\sum_j V_{*,j} = 0$ that

$$\max_{1\leq k\leq n} \|\textstyle\sum_{j=1}^k V_{*,\pi_2(j)}\|_\infty \leq \frac{A + B_1}{2}. \tag{5}$$

Let $B_2 := \mathrm{disc}(\pi_2)$. If $B_2 \leq A$, we output π_2. Otherwise, (5) gives $B_2 \leq \frac{A+B_1}{2}$, and so $A < B_2 < \frac{A+B_1}{2} < B_1$. We then repeat the procedure explained above to get another ordering π_3 from π_2.

Let $\ell = \log\big((B_1 - A)/\delta\big)$ and suppose we repeat this process ℓ times. Possibly some ordering π_i has $\mathrm{disc}(\pi_i) \leq A$, in which case the algorithm sets $\pi^* = \pi_i$ and terminates. If not, then since

$B_{i+1} - A < (B_i - A)/2$, we have $\operatorname{disc}(\pi_\ell) = B_\ell \leq A + \delta$. So the algorithm sets $\pi^* = \pi_\ell$ and terminates.

As a final remark, we should specify how to choose the initial ordering π_1. A good choice is to use the ordering produced by the algorithmic proof of the Steinitz lemma (Bar̃any (2008)). This choice ensures that $B_1 = d\delta$, and so the number of iterations ℓ is at most $\log(B_1/\delta) \leq \log d$. ■

3.3. Proofs of Theorem 4 and Theorem 7

The previous two sections have presented algorithmic proofs that $\mathbf{ss}(B_\infty, B_2; n) \leq O(\log^{2.5}(n))$ and that $\mathbf{st}(B_\infty, B_2; n) \leq \mathbf{ss}(B_\infty, B_2; n)$. The proof of Theorem 7 now follows easily.

Proof (of Theorem 7). We may think of the set V as a $d \times n$ matrix by imposing an arbitrary order on the vectors. Applying Theorem 9 with $\delta = 1$ shows that there exists an efficient randomized algorithm that, for any ordering on the columns of V, computes a coloring $\chi \in \{-1, 1\}^n$ with $\max_{1 \leq k \leq n} \|\sum_{j=1}^{k} V_{*,j}\chi_j\|_\infty \leq O(\log^{2.5} n)$.

The existence of this algorithm satisfies the key hypothesis of Theorem 10. Consequently Theorem 10 asserts that there exists an algorithm with running time $\operatorname{poly}(n, d)$ that constructs a permutation π of these vectors with

$$\max_{1 \leq t \leq n} \|\textstyle\sum_{j=1}^{t} V_{*,\pi(j)}\|_\infty \;\leq\; O(\log^{2.5} n). \tag{6}$$

Using standard norm inequalities, we have that

$$\max_{1 \leq t \leq n} \|\textstyle\sum_{j=1}^{t} V_{*,\pi(j)}\|_2 \;\leq\; O(\sqrt{d} \log^{2.5} n),$$

as required. ■

As observed earlier, Theorem 7 easily implies Theorem 4.

Algorithm 1: Our new algorithm for the Sampling Problem

```
Input: a set $\mathcal{X} = \{x_1, ..., x_n\} \subseteq \mathbb{R}^d$.
    Let $\delta = \max_{i \in [n]} \|x_i\|_2$ and $\mu = \sum_{i \in [n]} x_i/n$.
    Let $V = \{v_1, ..., v_n\}$ where $v_i = (x_i - \mu)/2\delta$.
    Since $\sum_{i \in [n]} v_i = 0$ and $\max_{i \in [n]} \|v_i\|_2 \leq 1$, we may apply the algorithm of Theorem 7 to
    get an ordering $\pi : [n] \to [n]$ of the elements of $V$.
Output: The infinite sequence $y_1, y_2, ...$ obtained by ordering the vectors in $\mathcal{X}$ according to the
    ordering $\pi$, and then repeating this ordering infinitely many times. Formally, for all $t \geq 1$,
    the $t^{\text{th}}$ output point is $y_t = x_{\pi((t-1) \bmod n)+1}$.
```

Proof (of Theorem 4). Let $\mathcal{X} \subset \mathbb{R}^d$ and let us denote its members as $x_1, ..., x_n$. Let $y_1, y_2, ...$ be the infinite sequence computed by Algorithm 1. We must show that this sequence has error $O\big(\sqrt{d} \log^{2.5}(n)/t\big)$.

Consider any $t \geq 0$, and let p and $0 \leq r \leq n-1$ be integers such that $t = pn + r$. Note that $\sum_{j=1}^{n} x_j - n\mu = 0$. Consequently, the error of $y_0, ..., y_t$ is

$$\begin{aligned} \|\textstyle\sum_{j=1}^{t} y_j/t - \mu\|_2 &= \|\textstyle\sum_{j=1}^{t} y_j - t\mu\|_2/t \\ &= \|(\textstyle\sum_{j=1}^{pn} y_j - pn\mu) + (\textstyle\sum_{j=pn+1}^{t} y_j - r\mu)\|_2/t \\ &= \|p\underbrace{(\textstyle\sum_{j=1}^{n} x_j - n\mu)}_{=0} + (\textstyle\sum_{j=1}^{r} x_{\pi(j)} - r\mu)\|_2/t \\ &= \|\textstyle\sum_{j=1}^{r} x_{\pi(j)} - r\mu\|_2/t \end{aligned}$$

By Theorem 7, the ordering π satisfies

$$\begin{aligned} &\max_{1\leq r\leq n} \|\textstyle\sum_{j=1}^{r} v_{\pi(j)}\|_2 \leq O(\sqrt{d}\log^{2.5} n) \\ \implies \quad &\max_{1\leq r\leq n} \|\textstyle\sum_{j=1}^{r} x_{\pi(j)} - t\mu\|_2 \leq O(\sqrt{d}\delta\log^{2.5} n). \end{aligned}$$

It follows that error is

$$\|\textstyle\sum_{j=1}^{t} y_j/t - \mu\|_2 \leq O\big(\sqrt{d}\delta\log^{2.5}(n)/t\big),$$

as required. ■

4. Lower Bound for the Sampling Problem

In this section we prove that *every* algorithm for the Sampling Problem has error at least $\Omega(\sqrt{d}\delta/t)$, proving Theorem 3. This shows that the algorithm of Theorem 4 is optimal, up to logarithmic factors.

Theorem 11 *For any d, let $n = d^2$ and define a matrix V of size $d \times (n+d)$ by*

$$V_{d\times(n+d)} = \begin{bmatrix} -1/n & \cdots & -1/n & 1 & \cdots & & 0 \\ -1/n & \cdots & -1/n & \vdots & \ddots & & \vdots \\ \vdots & & \vdots & & & 1 & 0 \\ -1/n & \cdots & -1/n & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

Here, the first n columns are $-1/n \cdot \mathbf{1}$ and the last d columns are $e_1, ..., e_d$. Then, for any sequence $x_1, x_2, ...$ generated from columns of this matrix, there exists $t \in \mathbb{N}$ such that $\|\sum_{i=1}^{t} x_t\|_2 \geq \Omega(\sqrt{d})$.

Proof Consider any sequence $x_1, x_2, ...$ Fix $t = n/2$ and consider the first t elements $x_1, ..., x_t$ in the sequence. Let a_i be the number of occurrences of e_i in these t elements, and let $a = \sum_{i=1}^{d} a_i$. Define $z := \sum_{i=1}^{t} x_i$. Then

$$z = \begin{bmatrix} a_1 - (t-a)/n \\ a_2 - (t-a)/n \\ \vdots \\ a_d - (t-a)/n \end{bmatrix}$$

The i^{th} coordinate of z is equal to $a_i - 1/2 + a/n$. We consider two cases:

Case 1: $a \geq n/3$. Then, there exists an index l for which $a_l \geq n/3d$. So

$$z_l \;\geq\; n/3d - 1/2 + 1/3 \;=\; n/3d - 1/6 \;\geq\; d/3 - 1/6.$$

Thus $\|z\|_2 \geq \Omega(d) = \Omega(\sqrt{d})$.
Case 2: $a \leq n/3$. Then, for all $i \in [d]$

$$a_i - 1/2 \;\leq\; z_i \;\leq\; a_i - 1/2 + 1/3 \;=\; a_i - 1/6.$$

Note that $a_i \in \mathbb{N}$ so $|z_i| \geq 1/6$. So every coordinate of z has absolute value greater than $1/6$. Thus

$$\|\textstyle\sum_{i=1}^{t} x_t\|_2 \;=\; \|z\|_2 \;=\; \Omega(\sqrt{d}),$$

as required. ■

Proof (of Theorem 3). Let $\mathcal{X}$ consist of the columns of V. Note that $\mu = \sum_{x \in X} x = 0$, and that $\delta = \max_{x \in X} \|x\|_2 = 1$. Assume that we have an algorithm that solves the Sampling Problem on $\mathcal{X}$ and generates pseudosamples $x_1, x_2, \ldots$. By Theorem 11 there exists an index $t \in \mathbb{N}$ such that $\|\sum_{i=1}^{t} x_t\|_2 \geq \Omega(\sqrt{d})$. Therefore

$$\|\textstyle\sum_{i=1}^{t} x_i/t - \mu\|_2 \;\geq\; \Omega(\sqrt{d}\delta/t),$$

as required. ■

5. Conclusions

5.1. Remarks

Our algorithm in Section 3 is described as a randomized algorithm. It can be derandomized by replacing our use of the Lovett-Meka algorithm with a derandomized form of Bansal's algorithm (Bansal and Spencer (2013)). It is also plausible to improve the running time of our algorithm by using recent results (T. Rothvoss (2014)) that should be faster than the Lovett-Meka algorithm.

Our bound $\mathbf{ss}(B_\infty, B_2; n) = O(\log^{2.5} n)$ can be improved if we do not require an efficient algorithm. It is also known (Banaszczyk (1998)) that $\mathbf{sv}(B_\infty, B_2; n) = O(\sqrt{\log n})$. Our results in Section 3 show that

$$\mathbf{ss}(B_\infty, B_2; n) \;=\; O(\log^{1.5} n) \cdot \mathbf{sv}(B_\infty, B_2; n). \tag{7}$$

Combining our result with Banaszczyk's yields a solution to the Sampling Problem with error $O(\delta\sqrt{d}\log^2(n)/t)$, but no known efficient algorithm achieves that bound.

Our lower bound in Theorem 3 is for the particular case that $t = d^2/2$. It would be interesting to get a similar bound for the cases $t \gg d^2$ and $t \ll d^2$.

5.2. Open questions

Our work leaves several questions unanswered. We do not prove that our algorithm of Section 3 has better error than the Herding algorithm. In particular, it is conceivable that the Herding algorithm has optimal error $O(\delta\sqrt{d}/t)$, matching our lower bound. Alternatively, it is also conceivable that the $O(\delta d/t)$ bound on the Herding algorithm's error is actually tight.

We do not know whether our bound in (7) is tight. It is conceivable that

$$\mathbf{ss}(B_\infty, B_2; n) \;=\; O(1) \cdot \mathbf{sv}(B_\infty, B_2; n). \tag{8}$$

It seems that techniques substantially different than ours would be required to prove this.

A major open question in discrepancy theory (Bergström (1931); Bařaňy (2008)) is whether

$$\mathbf{ss}(B_2, B_2; n) \;\leq\; O(\sqrt{d}). \tag{9}$$

It would be very interesting if (8) is true, as then the Komlós conjecture would imply (9).

Our algorithm in Section 3 assumes that the input set $\mathcal{X}$ is finite. It is interesting to study if we can generalize our algorithm for case of infinite $\mathcal{X}$. It would also be interesting to study if our algorithm gives good results for the intended applications of Herding in Markov random fields.

Our definition of the Sampling Problem states that μ is the mean of $\mathcal{X}$, whereas a more general formulation could incorporate a distribution p on $\mathcal{X}$. It is interesting to see if one can modify our algorithm for the more general case. Here we briefly suggest one possible approach. It seems that the more general case should be reducible to the uniform distribution case. Assume that p assigns probability $p(x_i)$ to each point $x_i \in \mathcal{X}$. If each $p(x_i) = \frac{a_i}{b_t}$ for natural numbers a_i and b_i, then one can set $c = LCM(b_1, ..., b_n)$ and generate a new multi-set $\mathcal{Y}$ with $c \cdot \frac{a_i}{b_i}$ copies of x_i. The uniform distribution on $\mathcal{Y}$ is the same as the distribution p on the original set $\mathcal{X}$. Now we only need to feed set $\mathcal{Y}$ to our algorithm. There are two caveats with this approach. First, we have ignored irrational numbers and second, we may have $|\mathcal{Y}| \gg |\mathcal{X}|$. It seems possible to mitigate the runtime increase and to deal with irrational numbers through the use of standard sampling lemmas (Althöfer (1994)), which can approximate p by a distribution for which c is small.

Acknowledgements

We thank Nando de Freitas, Nikhil Bansal, Mohit Singh, David Kirkpatrick and anonymous reviewers.

References

N. Alon and J. Spencer. *The Probabilistic Method.* John Wiley,, 2000.

I. Althöfer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and its Applications*, 199:339–355, 1994.

Edoardo Amaldi and Raphael Hauser. Boundedness theorems for the relaxation method. *Mathematics of Operations Research*, 30:939–955, 2005.

F. Bach, S. Lacoste-Julien, and G. Obozinsk. On the equivalence between herding and conditional gradient algorithms. In *In Proc. ICML*, 2012.

K. Ball. An elementary introduction to modern convex geometry. *Random Structures and Algorithms*, 31, 1997.

W. Banaszczyk. Balancing vectors and Gaussian measures of n-dimensional convex bodies. *Random Structures and Algorithms*, 12(4):351–360, 1998.

W. Banaszczyk. On series of signed vectors and their rearrangements. *Random Structures and Algorithms*, 40:301–316, 2012.

N. Bansal. Constructive algorithms for discrepancy minimization. In *FOCS*, 2010.

N. Bansal and J. Spencer. Deterministic discrepancy minimization. *Algorithmica*, 67(4):451–471, 2013.

I. Bařańy. On the power of linear dependencies. *Building Bridges, Bolyai Society Mathematical Studies*, 19:31–45, 2008.

Viktor Bergström. Zwei sätze über ebene vektorpolygone. *Abh. Math. Sem. Univ. Hamburg*, 8: 148–152, 1931.

H. D. Block and S. A. Levin. On the boundedness of an iterative procedure for solving a system of linear inequalities. *Proceedings of the American Mathematical Society*, 26:229–235, 1970.

L. Bornn, Y. Chen, N. de Freitas, M. Eskelin, J. Fang, and M. Welling. Herded Gibbs sampling. In *International Conference on Learning Representations*, 2013.

Bernard Chazelle. *The Discrepancy Method: Randomness and Complexity.* Cambridge University Press, 2000.

S. Chen, J. Dick, and A. B. Owen. Consistency of Markov chain quasi-monte carlo on continuous state spaces. *Annals of Statistics*, 39:673–701, 2011.

Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. In *In Proc. UAI*, 2010.

A. Gelfand, Y. Chen, L. van der Maaten, and M. Welling. On herding and the perceptron cycling theorem. In *In Proc. NIPS*, 2010.

Alexander E. Holroyd and James Propp. Rotor walks and Markov chains. *Algorithmic Probability and Combinatorics*, 520:105–126, 2010.

Ferenc Huszár and David Duvenaud. Optimally-weighted herding is Bayesian quadrature. In *In Proc. UAI*, 2012.

S. Lovett and R. Meka. Constructive discrepancy minimization by walking on the edges. In *FOCS*, 2012.

Jiri Matousek. *Geometric Discrepancy: An Illustrated Guide*. Springer, 1999.

J. Spencer. Six standard deviations suffice. *Trans. Amer. Math. Soc.*, 289:679–706, 1985.

E. Steinitz. Bedingt konvergente reihen und konvexe systeme. *J. Reine Ang. Mathematik*, 143: 128–175, 1913.

T. Rothvoss. Constructive discrepancy minimization for convex sets. *CoRR*, abs/1404.0339, 2014.

R. Vershynin. Lecture notes in geometric functional analysis, 2009. manuscript.

M. Welling. Herding dynamical weights to learn. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

Appendix A. Proof of Theorem 5

Theorem 5 states that the algorithm of Section 3 has error $O(\log^{2.5}(n)/t)$ in the ℓ_∞-norm. This can easily be seen from (6). On the other hand, the following theorem shows that the Herding algorithm has error $\Omega(\sqrt{d}/t)$ in the ℓ_∞-norm. These facts prove Theorem 5.

Theorem 12 *Let V be a matrix of size $d \times n$ with*

$$V = \begin{array}{c|ccc|ccc|cccc|ccc}
\overbrace{\quad}^{A} & \multicolumn{3}{c|}{\overbrace{\qquad\qquad}^{B}} & \multicolumn{3}{c|}{\overbrace{\qquad\qquad}^{C}} & \multicolumn{4}{c|}{\overbrace{\qquad\qquad\qquad}^{D}} & \multicolumn{3}{c}{\overbrace{\qquad\qquad}^{E}} \\
1/\sqrt{d} & -\epsilon & \cdots & -\epsilon & 0 & \cdots & 0 & -1 & 0\cdots & & 0 & \epsilon & \cdots & \epsilon \\
 & & & & & & & 0 & \ddots & & & & & \\
\vdots & \vdots & & \vdots & & & & \vdots & & & 0 & \vdots & & \\
1/\sqrt{d} & -\epsilon & \cdots & -\epsilon & 0 & \cdots & 0 & 0 & \cdots & 0 & -1 & \epsilon & \cdots & \epsilon \\
1 & 0 & \cdots & 0 & -\epsilon & \cdots & -\epsilon & 0 & \cdots & 0 & 0 & 0 & \cdots & 0
\end{array}$$

in which $|A| = 1$, $|B| = 1/(\sqrt{d}\epsilon)$, $|C| = 1/\epsilon$, $|D| = d - 1$, and $|E| = 1/\epsilon$. So $n = d + 1/\sqrt{d}\epsilon + 1/\epsilon + 1/\epsilon$. This definition ensures that μ, the average of the columns, is 0. We define $\epsilon = /d\sqrt{d}$. Let $x_1, x_2, \ldots$ be the pseudosamples generated by the Herding algorithm when we apply it to matrix V, with $w_0 = 0$. Then there exists $T \in \mathbb{N}$ such that $\|\sum_{i=1}^{T} x_i\|_\infty \geq \Omega(\sqrt{d})$.

Proof Set $T = (2\sqrt{d}+1)(\sqrt{d}/2 - 1)$. Assume that the Herding algorithm has chosen pseudosamples $x_1, x_2, \ldots, x_T$ in the first T iterations. From the definition of the Herding algorithm we know that $w_i = w_{i-1} - x_i$ for all $1 \leq i \leq T$. Summing these T equalities $w_T = w_0 - (x_1 + x_2 + \ldots + x_T)$. Substituting $w_0 = 0$, we get that $w_T = -(x_1 + x_2 + \ldots + x_T)$. So in order to compute the value of $\sum_{i=1}^{T} x_i$ we only need to compute the vector w_T.

We claim that for all $1 \leq t \leq T$, w_t is as follows. Choose $0 \leq k \leq \sqrt{d}/2 - 1$ such that $(2\sqrt{d}+1)(k-1)+1 < t \leq (2\sqrt{d}+1)k+1$ then

$$w_t = \big[\overbrace{1-k/\sqrt{d}, \cdots, 1-k/\sqrt{d}}^{t-1}, \overbrace{-k/\sqrt{d}, \cdots, -k/\sqrt{d}}^{d-t}, \overbrace{-k}^{1}\big]^\mathsf{T}$$

in which there are $t-1$ coordinates of value $1-k/\sqrt{d}$ on the top, $d-t$ coordinates of value $-k/\sqrt{d}$ in the middle and one coordinate of value $-k$ at the bottom. We prove our claim by induction:

Initial case of induction; $t = 1$. From the definition of the Herding algorithm, x_1 is defined to be $\arg\max_{j \in [n]} \langle w_0, V_{*,j} \rangle$. Since $w_0 = 0, x_1 = V_1$. So $w_1 = w_0 - x_1 = -V_1$. This is the same as w_1 in the above formula for $k = 1$.

Inductive step. Assuming that the hypothesis is true for some $t = 1, 2, ..., l$, we prove that it is also true for $t = l + 1$. We consider two cases:

- $t = (2\sqrt{d}+1)k+1$ for some $0 \leq k \leq \sqrt{d}/2 - 1$. In order to get x_{t+1} we need to compute $\langle w_t, V_j \rangle$ for all $j \in [n]$ and get the column that maximizes this value. We consider five cases:

  Case 1: V_j is selected from block A. In this case $\langle w_t, V_j \rangle = k/\sqrt{d} + k/d$.
  Case 2: V_j is selected from block B. In this case $\langle w_t, V_j \rangle = -\epsilon k(\sqrt{d} + 1 + 1/\sqrt{d})$.
  Case 3: V_j is selected from block C. In this case $\langle w_t, V_j \rangle = k\epsilon$.
  Case 4: V_j is selected from block D. In this case $\langle w_t, V_j \rangle = -1 + k/\sqrt{d}$ or $k/\sqrt{d}$.
  Case 5: V_j is selected from block E. In this case $\langle w_t, V_j \rangle = \epsilon k(\sqrt{d} + 1 + 1/\sqrt{d})$.

  Substituting the values of ϵ, ϵ and ϵ, it is easy to check that the maximum value is achieved by choosing $x_{t+1} = V_1$. So $w_{t+1} = w_t - x_{t+1} =$

  $$\big[\overbrace{1-(k+1)/\sqrt{d}, \cdots, 1-(k+1)/\sqrt{d}}^{t-1}, \overbrace{-(k+1)/\sqrt{d}, \cdots, -(k+1)/\sqrt{d}}^{d-t}, \overbrace{-(k+1)}^{1}\big]^\mathsf{T}$$

  Note that $t = (2\sqrt{d}+1)k+1$ which gives $t+1 = (2\sqrt{d}+1)k+2$. Thus $(2\sqrt{d}+1)((k+1)-1)+1 < t < (2\sqrt{d}+1)(k+1)+1$ which gives the same value for w_{t+1} in the hypothesis.
- $(2\sqrt{d}+1)(k-1)+2 \leq t \leq (2\sqrt{d}+1)k$ for some $1 \leq k \leq \sqrt{d}/2-1$. Similar to the previous case, we need to compute $\langle w_t, V_j \rangle$ for all $j \in [n]$ and get the column that maximizes this value. We consider five cases:

  Case 1: V_j is selected from block A. In this case $\langle w_t, V_j \rangle \leq \epsilon(k-1)/\sqrt{d} + k/d$.
  Case 2: V_j is selected from block B. In this case $\langle w_t, V_j \rangle \leq \sqrt{d}(2-k) - k - k/\sqrt{d}$.
  Case 3: V_j is selected from block C. In this case $\langle w_t, V_j \rangle = k\epsilon$.
  Case 4: V_j is selected from block D. In this case $\langle w_t, V_j \rangle = -1 + k/\sqrt{d}$ or $k/\sqrt{d}$.
  Case 5: V_j is selected from block E. In this case $\langle w_t, V_j \rangle \leq \epsilon(\sqrt{d}k + k + k/\sqrt{d} - 1)$.

  Similar to the previous case, it is easy to check that the maximum value is achieved in the third case by choosing the $x_{t+1} = -e_t$ from block D. (Recall that e_t denotes the t^{th} standard basis vector). So $w_{t+1} = w_t - x_{t+1} =$

$$\Big[\overbrace{1-k/\sqrt{d},\cdots,1-k/\sqrt{d}}^{t},\ \overbrace{1-k/\sqrt{d},\cdots,-(k+1)/\sqrt{d}}^{d-1-t},\ \overbrace{-(k+1)}^{1}\Big]^{\mathsf{T}}$$

in which the top t coordinate are $1-k/\sqrt{d}$ and the middle $d-1-t$ coordinates are $-k/d$ and the last coordinate is $-k$. Note that $t+1$ is still in the same interval as before and therefore the above vector gives the same value for w_{t+1} as in the hypothesis. This completes the proof of our claim.

The proof completes by setting $T = (2\sqrt{d}+1)(\sqrt{d}/2-1)$ and $k = \sqrt{d}/2-1$. In this case $\|w_T\|_\infty \geq \sqrt{d}/2 - 1 = \Omega(\sqrt{d})$. ■

Appendix B. Bounding (3)

In this appendix, we show that (3) is satisfied. This calculation is just a slight modification of a calculation appearing in Lovett and Meka (2012).

Dividing by n, it suffices to show that

$$\sum_{r\in\mathbb{Z}} \frac{1}{2^r}\exp\left(\frac{-C^2}{2^{r+5}}\right) \leq 1/16. \tag{10}$$

First, assuming only that $C \geq 0$, we have

$$\sum_{r\geq 6} \frac{1}{2^r}\exp\left(\frac{-C^2}{2^{r+5}}\right) \leq \sum_{r\geq 6}\frac{1}{2^r} = 1/32. \tag{11}$$

Next assume that $C = 32$. Then

$$\sum_{r\leq 0} \frac{1}{2^r}\exp\left(\frac{-C^2}{2^{r+5}}\right) = \sum_{i\geq 0} 2^i\cdot\exp\left(\frac{-C^2\cdot 2^i}{32}\right) \leq \sum_{i\geq 0}\exp\left(i-32\cdot 2^i\right) < \sum_{i\geq 1}\exp(-31i)$$

$$= \frac{e^{-31}}{1-e^{-31}} < 1/200.$$

Finally,

$$\sum_{r=1}^{5} \frac{1}{2^r}\exp\left(\frac{-C^2}{2^{r+5}}\right) = \frac{e^{-16}}{2}+\frac{e^{-8}}{4}+\frac{e^{-4}}{8}+\frac{e^{-2}}{16}+\frac{e^{-1}}{32} < 1/44. \tag{12}$$

Summing (11), (B) and (12) shows (10).

Appendix C. Herding algorithm error

Proof This is a modification of the proof in Chen et al. (2010). Assume that the Herding algorithm generates pseudosamples $x_1, x_2, ..., x_t, ...$ and weights $w_0, w_1, w_2, ..., w_t, ...$. We have that $w_i = w_{i-1} + \mu - x_i$ for all $1 \leq i \leq t$. Summing these above equations we get that $w_t = w_0 + t\mu - \sum_{i=1}^{t} x_i$. The error of the Herding algorithm for the first t samples is

$$\left\| \frac{1}{t} \sum_{i=1}^{t} x_i - \mu \right\|_2 = \frac{1}{t} \left\| \sum_{i=1}^{t} x_i - t\mu \right\|_2 = \frac{1}{t} \|w_0 - w_t\|_2 \leq \frac{1}{t} (\|w_0\|_2 + \|w_t\|_2)$$

So in order to bound the error, we only need to bound $\|w_t\|_2$. We claim that $\|w_t\|_2 \leq O(\delta^2/r)$.

Let's define the convex body $\mathcal{C} = \mathcal{K} - \mu$. Since $\|x\|_2 \leq \delta$ for all $x \in \mathcal{K}$, $\|c\|_2 \leq 2\delta$ for all $c \in \mathcal{C}$. Let $c_t = \arg\max_{c \in \mathcal{C}} \langle w_t, c \rangle$ for all t. The Herding algorithm update equation becomes $w_{t+1} = w_t - c_t$. Since there is a ball of radius r with centre μ inside $\mathcal{K}$ and $\mathcal{C}$ is a shifting of $\mathcal{K}$, there is a ball of radius r with centre 0 inside $\mathcal{C}$. Thus $r \cdot \frac{w_t}{\|w_t\|_2} \in \mathcal{C}$.

Consider the sequence $w_0, w_1, ...$ and let k be the first index for which $\|w_k\|_2 > \frac{2\delta^2}{r}$. Then

$$\|w_{k+1}\|_2^2 - \|w_k\|_2^2 = \|c_k\|_2^2 - 2\langle w_k, c_k \rangle \leq 4\delta^2 - 2\langle w_k, r \cdot \frac{w_k}{\|w_k\|_2} \rangle = 4\delta^2 - 2r\|w_k\|_2 < 0.$$

Therefore $\|w_k\|_2 > \|w_{k+1}\|_2$. This means that as soon as the norm of a weight point gets bigger than $\frac{2\delta^2}{r}$, the norm starts decreasing until it gets smaller than $\frac{2\delta^2}{r}$ again. Since $w_k = w_{k-1} - c_{k-1}$, we have

$$\|w_k\|_2 \leq \|w_{k-1}\|_2 + \|c_{k-1}\|_2 \leq \frac{2\delta^2}{r} + 2\delta = O(\frac{\delta^2}{r})$$

Note that this is the maximum value that the norm of any vector w_t can get. Therefore the error of Herding algorithm is bounded by

$$\left\| \frac{1}{t} \sum_{i=1}^{t} x_i - \mu \right\|_2 \leq \frac{1}{t} (\|w_0\|_2 + \|w_t\|_2) \leq O(\|w_0\|_2 + \frac{\delta^2}{r})/t.$$

■