Towards Efficient Evaluation of Risk via Herding

Zelai Xu [*1] **Tiancheng Yu** [*2] **Suvrit Sra** [2]

Abstract

We introduce a novel use of herding to address the problem of selecting samples from a large unlabeled dataset to efficiently evaluate the risk of a given model. Herding is an algorithm which elaborately draws samples to approximate the underlying distribution. We use herding to select the most informative samples and show that the loss evaluated on k samples produced by herding converges to the expected loss at a rate $\mathcal{O}(1/k)$, which is much faster than $\mathcal{O}(1/\sqrt{k})$ for iid random sampling. We validate our analysis on both synthetic data and real data, and further explore the empirical performance of herding-based sampling in different cases of high-dimensional data.

1. Introduction

A key component of any machine learning pipeline is performance evaluation. Traditionally, performance is evaluated by comparing the prediction to ground truth on a test set. For large-scale machine learning, the scale of the test set also has to grow to achieve high evaluation accuracy (Bennett & Carvalho, 2010). Thus, manually labeling the ground truth becomes costly and depends on experts with domain knowledge. Compared with training, performance evaluation usually requires less data, while being used more widely since many users do not train their own model but only choose from existing ones.

Since in many applications, it is easier to obtain unlabeled data, it is desirable to find a systematic scheme to choose the most informative data points to label (Katariya et al., 2012; Sawade et al., 2010; Welinder et al., 2013). Herding is a natural candidate for this problem. Given a set $\mathcal{X} \subset \mathbb{R}^d$, herding constructs an infinite sequence of samples $\{x^{(i)}\}_{i=1}^{\infty}$ from $\mathcal{X}$, such that the mean of a suitable function $h(x)$ over the first k samples $\{x^{(i)}\}_{i=1}^{k}$ converges to the mean of $h(x)$ over the whole set $\mathcal{X}$ with a rate of $\mathcal{O}(1/k)$. Herding motivates the central question of our paper:

Can herding achieve the convergence rate of $\mathcal{O}(1/k)$ for the task of performance evaluation?

This problem has many connections to statistical learning theory. A central problem in learning theory is to achieve the optimal sample complexity. That is, given accuracy level ε, how many labeled samples are required to train/test a machine learning model? Learning theory has shown that, for PAC learning, the sample complexity is at least $\mathcal{O}(1/\varepsilon^2)$ and can be achieved by the popular ERM paradigm within a logarithm factor. However, in practice, we usually have some side information which can be used to reduce sample complexity (Vapnik & Izmailov, 2015). A particularly common source of side information is unlabeled samples. As shown in (Golovnev et al., 2019), knowing the distribution of unlabeled samples provably reduces sample complexity.

Similar settings have been studied in experiment design and active learning literature, where we have a large pool of unlabeled samples and pick a small number of them to label (Angluin, 1988). In experiment design, the statistical model is assumed to be known and leveraged to design an optimal criterion (Gevers & Ljung, 1986; Allen-Zhu et al., 2017). This is usually equal to a combinatorial optimization task and the solution corresponds to the samples to label. In active learning setting, the unlabeled data is usually used to explore either the cluster structure of the samples or the hypothesis space. This usually involves a complicated interactive process of query and labeling.

This problem is also related to affine invariant optimization. An optimization method is called affine invariant if, under affine transformation of the input data, every new iterate value remains exactly the transform of the old value, yielding an unchanged convergence rate (Lacoste-Julien & Jaggi, 2013). An affine invariant method is preferred in model evaluation since the performance is unaffected by normalization of the original data.

Can we choose samples to label agnostically and still reduce the sample complexity? Generally speaking, this is impossible according to learning theoretic arguments above. However, if we assume that the loss function is a determinis-

[*]Equal contribution [1]Department of Electronic Engineering, Tsinghua University, Beijing, China [2]MIT EECS, Cambridge, MA, USA. Correspondence to: Zelai Xu <xuzl16@mails.tsinghua.edu.cn>, Tiancheng Yu <yutc@mit.edu>, Suvrit Sra <suvrit@mit.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

tic function of the samples and belongs to some reproducing kernel Hilbert space (RKHS), then herding can be used to achieve a fast convergence rate of $\mathcal{O}(1/k)$.

1.1. Contribution

- We introduce the use of herding to select test samples for performance evaluation. We show how herding can use fewer samples to efficiently evaluate the loss than uniform random sampling.
- We run several experiments to explore the empirical performance of herding-based sampling. We validate our analysis by numerical experiments on both synthetic and real data. We also explored the influence of dimensionality by doing experiments on several high-dimensional datasets with low-dimensional structure. Furthermore, we run experiments to show the affine invariance problem in herding-based sampling.

2. Herding-based Sampling

2.1. Setting

We first give a more concrete description of our setting. Let $\mathcal{X}, \mathcal{Y}$ be the input and output space, and let $\mathcal{F}$ be the hypothesis space of functions $f : \mathcal{X} \to \mathcal{Y}$. Suppose we have n unlabeled data from some underlying distribution p, the expected risk of any function $f \in \mathcal{F}$ is defined as

$$\mathcal{E}(f) = \mathbb{E}_{x\sim p}[L(f(x), y(x))],$$

and the empirical risk of f over k selected samples $\{x^{(i)}\}_{i=1}^{k}$ is

$$\hat{\mathcal{E}}_k(f) = \frac{1}{k}\sum\nolimits_{i=1}^{k} L(f(x^{(i)}), y(x^{(i)})).$$

Goal: select k out of k unlabeled sample to label, and use in test phase to efficiently evaluate the expected risk of the current model such that

$$\|\mathcal{E}(f) - \hat{\mathcal{E}}_k(f)\| = \mathcal{O}(1/k).$$

2.2. Problem Formulation

We show how herding can be used to address the sampling problem. Herding (Chen et al., 2012) is a generic method for estimating the expectation of function h in some reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with kernel function $K(\cdot,\cdot)$. Herding generates samples $\{x^{(i)}\}_{i=1}^{k}$ by the following iterations:

$$x^{(t+1)} = \underset{x}{\text{argmax}}\langle w^{(t)}, \phi(x)\rangle_{\mathcal{H}},$$
$$w^{(t+1)} = w^{(t)} + \mathbb{E}_{x\sim p}\phi(x) - \phi(x^{(t+1)}),$$

where $\{w^{(i)}\}_{i=1}^{k}$ is an auxiliary sequence initialized at 0 and $\phi(\cdot)$ is the corresponding feature map of the kernel $K(\cdot,\cdot)$. An intriguing property of herding is that for any function $h \in \mathcal{H}$, we have

$$\left\|\frac{1}{k}\sum\nolimits_{i=1}^{k} h(x^{(i)}) - \mathbb{E}_{x\sim p}h(x)\right\| = \mathcal{O}(1/k).$$

In our sampling problem, we consider the case of test phase, where the learned function f is fixed. We assume the loss function $L(x) = L(f(x), y(x))$ is deterministic. This is true for most large-scale applications, and the random part cannot be handled by wisely selecting samples to label. We also assume $L(x) \in \mathcal{H}$, then we have

$$\left\|\frac{1}{k}\sum\nolimits_{i=1}^{k} L(x^{(i)}) - \mathbb{E}_{x\sim p}L(x)\right\| = \mathcal{O}(1/k).$$

2.3. Curse of Dimensionality

The argument above show that herding is able to accelerate the convergence rate to $\mathcal{O}(1/k)$. However, the performance of herding-based sampling becomes drastically worse when the samples live in a high-dimensional space. This is because the convergence rate of herding is exponential in dimensionality d (Amaldi & Hauser, 2005), and the performance of herding-based sampling can even be worse than random sampling when the dimensionality is high enough. Although another herding method has been proposed to achieves a better convergence rate of $\mathcal{O}(\sqrt{d}\log^{2.5} n/k)$ (Harvey & Samadi, 2014), it is not computationally practical since the running time is at least $\mathcal{O}(n^5 d^5)$, where n is the total number of unlabeled samples and can be very large in real-world applications.

However, though the input samples usually live in a high-dimensional space, the underlying distribution p of the samples may still mainly concentrate on some low-dimensional structure, for example, a low dimensional manifold. In this case, herding may still be useful to reduce sample complexity. In the next section, we designed several experiments to show herding's empirical performance in this scenario.

3. Experiments

In this section, we run herding and random sampling on synthetic and real data to compare their performance by plotting error= $\left\|\frac{1}{k}\sum_{i=1}^{k} L(x^{(i)}) - \mathbb{E}_{x\sim p}L(x)\right\|$ with respect to sample number k as well as dimensionality d. In the following experiments, unless otherwise stated, we set $n = 10000, d \in [2, 30], k \in [1, 100]$. The results are averaged over $m = 100$ independent experiments.

3.1. Synthetic Data

Consider the linear regression model $y = \beta^T x$, the corresponding loss function is $L(x, y) = (\hat{\beta}^T x - y)^2$. In this experiment, we first generate $\{x^{(i)}\}_{i=1}^{n}$ where

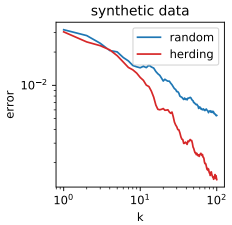

Figure 1. Result of linear regression on synthetic data.

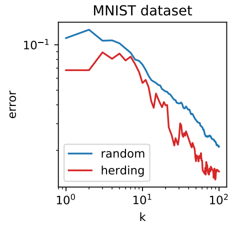

Figure 2. Result of SVM classification on MNIST dataset.

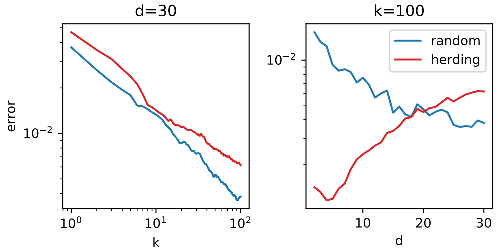

Figure 3. Results of full components.

$x_j^{(i)} \sim \mathcal{U}[0,1]$ for all $i \in [n], j \in [d]$. Then generate $\beta \sim \mathcal{U}[0,1]$ and calculate $y^{(i)} = \beta^T x^{(i)} + \epsilon^{(i)}$ where $\epsilon^{(i)} = 1/d \sum_{j=1}^{d} \log(2x_j^{(i)})$ is a noisy term. The result of error w.r.t k when $d = 10$ is shown in Figure 1.

As shown in Figure 1, the error of sampling by herding is much smaller than random sampling with the same number of test samples. Also, the convergence rate of herding is close to the theoretical $\mathcal{O}(1/k)$.

3.2. Real Data

Consider the SVM classification with RBF kernel on MNIST dataset, the corresponding loss function is the 0-1 loss $L(x,y) = \mathbb{1}_{\hat{y}(x) \neq y}$. We first train a SVM classifier on 10000 samples from the MNIST training set, and then use herding to sample $k = 100$ from $n = 3000$ samples for performance evaluation. Since herding is a deterministic process, we randomly select the $n = 3000$ samples from the MNIST test set to get an averaged result shown in Figure 2.

Similar to the results in Figure 1, the error of herding is smaller than random sampling, but the convergence rate appears to be a little worse than the theoretical $\mathcal{O}(1/k)$. On the one hand, this is because the 0-1 loss function on MNIST data is not necessarily a deterministic function in the RKHS $\mathcal{H}$; on the other hand, the dimensionality of MNIST dataset is relatively high (784). As shown in the previous arguments, the convergence rate of herding is exponential in dimensionality, resulting in the drop in performance in high-dimensional space.

3.3. Influence of Dimensionality

To further explores the influence of dimensionality, we consider a nonlinear loss function $L(x) = \|x\|/\sqrt{d}$ which is a little more complex than previous ones. We run several experiments on different distributions of samples and compare their results.

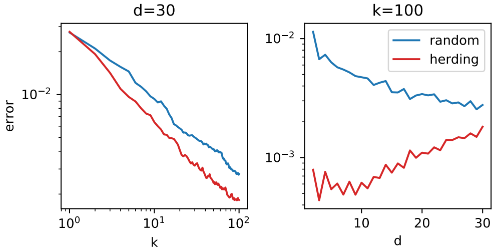

Figure 4. results of partial zero components.

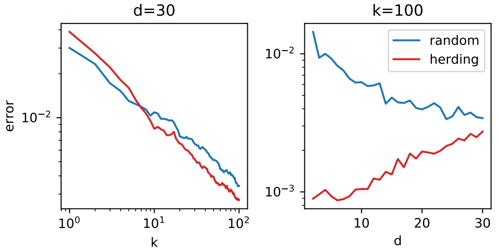

Figure 5. results of partial compressed components.

3.3.1. Case 1: Full Components

This experiment serves as a baseline to compare with the results of following experiments. The unlabeled samples $\{x^{(i)}\}_{i=1}^{n}$ are generated the same as the previous section, that is $x_j^{(i)} \sim \mathcal{U}[0,1]$ for all $i \in [n], j \in [d]$. The results of error w.r.t. k and d are shown in Figure 3.

In the left plot of Figure 3, we see that when $d = 30$, the performance of herding is already worse than that of random sampling. The right plot better demonstrates the curse of dimensionality in herding-based sampling, where the error of herding increases w.r.t. to d while random sampling decreases.

3.3.2. Case 2: Partial Zero Components

In this experiment, we set half of the components of x to be 0. More specifically, we generate $\{x^{(i)}\}_{i=1}^{n}$ where when $j \leq d/2$, $x_j^{(i)} \sim \mathcal{U}[0,1]$ for all $i \in [n]$, and when $j > d/2$, $x_j^{(i)} = 0$ for all $i \in [n]$. In this case, although the dimensionality of the samples is high, they actually live on a subspace with much lower dimensionality. The results are shown in Figure 4.

From the first plot of Figure 4, herding still outperforms random sampling even when $d = 30$ in this case. Compare the plots in Figure 3 and Figure 2, we see that herding performs better in case 2 than case 1 when d is the same. This is quite straightforward since the sample set with dimensionality d in case 2 is equivalent to the set with dimensionality $\lfloor d/2 \rfloor$ in case 1. And the first plot of Figure 3 is actually very similar to the plot of error w.r.t. k when $d = 15$ in case 1.

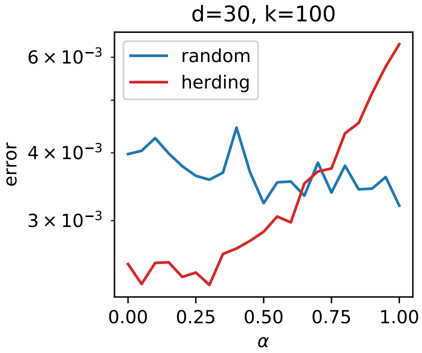

Figure 6. error w.r.t. α when $d = 30, k = 100$.

3.3.3. Case 3: Partial Compressed Components

To see the results when the distribution is somewhere between case 1 and case 2, we consider the distribution where half of the components of x are compressed towards 0. More specifically, we generate $\{x^{(i)}\}_{i=1}^n$ where when $j \le d/2$, $x_j^{(i)} \sim \mathcal{U}[0,1]$ for all $i \in [n]$, and when $j > d/2$, $x_j^{(i)} \sim \mathcal{U}[0,\alpha]$ for all $i \in [n]$. In this case, the samples still live on a high-dimensional space, but is distributed more concentrated in some dimensions while less on the others. We can think of this samples set as the set in case 2 being added some noise, i.e. the true underlying samples still live on a low-dimensional subspace, but the observed data was polluted by some high-dimensional noise. The results when $\alpha = 0.5$ are shown in Figure 5.

The results are quite similar to that of case 2. To further explore the influence of α, we fix $d = 30, k = 100$ and let α varies from 0 to 1. Notice that when $\alpha = 1$, this case becomes case 1; when $\alpha = 0$, this case reduces to case 2. The plot of error w.r.t. α is shown in Figure 6.

As Figure 6 shows, when α gets larger, the performance of herding becomes worse, while the performance of random sampling remains approximately the same. Also, error grows relatively slower when $\alpha \le 0.5$ compared to $\alpha > 0.5$. This suggests that as long as the magnitude of noise is not too large compared to the true data, herding can still perform well even in high-dimensional case.

3.4. Affine Invariance

In this subsection, we show a problem of herding-based sampling that it is not affine invariant. A common preprocessing of data in machine learning is normalization, that is applying an affine transformation $f(x) = Ax + b$ on the original data. We hope sampling on data after affine transformations still efficiently evaluates the loss. However, it turns out that the performance of herding-based sampling can be much worse after affine transformations. In fact, both shifting and rescaling of the original data may have an influence on herding's performance.

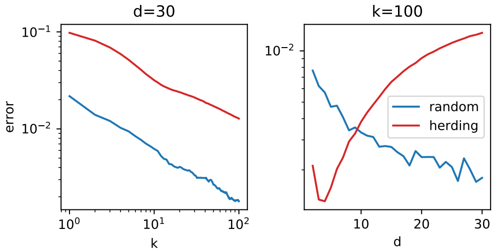

Figure 7. Results of shifted data.

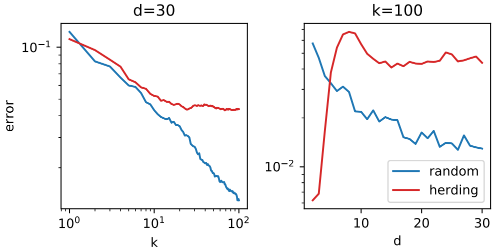

Figure 8. Results of rescaled data.

In the following experiments, we first generate $\{x^{(i)}\}_{i=1}^n$ the same as subsection 3.3.1. In the first experiment, we shift the original data $\overline{x} = x - 0.5 \cdot \mathbb{1}_d$ so that they are zero mean. In the second experiment, we rescale the data $\tilde{x} = 2\sqrt{3} \cdot x$ so that they are unit variance. We perform herding on the shifted and rescaled data respectively, and the results are shown in Figure 7 and 8.

Compare the results in Figure 7, 8 with Figure 3, we can see that the performance of herding-based sampling is much worse after both shifting and rescaling. Although there is some work (Lacoste-Julien & Jaggi, 2013) on the affine invariance of herding's equivalent algorithm conditional gradient algorithms (Bach et al., 2012), it cannot be extended to herding since the corresponding affine invariance is in the feature space instead of input space of herding.

4. Discussion

Herding-based sampling accelerates the convergence rate of empirical loss over k selected samples from $\mathcal{O}(1/\sqrt{k})$ to $\mathcal{O}(1/k)$, making the performance evaluation of machine learning models more efficient. Experiments on simulated data show that though herding-based sampling suffers from the curse of dimensionality, it still works well on high-dimensional datasets with low-dimensional structure. However, herding-based sampling is not affine invariant, which may cause a drop of performance when data are normalized.

The empirical performance of herding-based sampling leads to questions that can be further studied, that is to propose herding-style sampling algorithms which are more robust to dimensionality and affine transformation. In addition, theoretical analysis of herding on high-dimensional data with low-dimensional structure can also be further explored.

References

Allen-Zhu, Z., Li, Y., Singh, A., and Wang, Y. Near-optimal discrete optimization for experimental design: A regret minimization approach. *arXiv preprint arXiv:1711.05174*, 2017.

Amaldi, E. and Hauser, R. Boundedness theorems for the relaxation method. *Math. Oper. Res.*, 30(4):939–955, November 2005.

Angluin, D. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

Bach, F., Lacoste-Julien, S., and Obozinski, G. On the equivalence between herding and conditional gradient algorithms. 2012.

Bennett, P. N. and Carvalho, V. R. Online stratified sampling: evaluating classifiers at web-scale. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1581–1584. ACM, 2010.

Chen, Y., Welling, M., and Smola, A. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012.

Gevers, M. and Ljung, L. Optimal experiment designs with respect to the intended model application. *Automatica*, 22(5):543–554, 1986.

Golovnev, A., Pál, D., and Szörényi, B. The information-theoretic value of unlabeled data in semi-supervised learning. *arXiv preprint arXiv:1901.05515*, 2019.

Harvey, N. and Samadi, S. Near-optimal herding. In *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pp. 1165–1182, Barcelona, Spain, 13–15 Jun 2014. PMLR.

Katariya, N., Iyer, A., and Sarawagi, S. Active evaluation of classifiers on large datasets. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pp. 329–338. IEEE, 2012.

Lacoste-Julien, S. and Jaggi, M. An affine invariant linear convergence analysis for frank-wolfe algorithms. 2013.

Sawade, C., Landwehr, N., Bickel, S., and Scheffer, T. Active risk estimation. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 951–958. Citeseer, 2010.

Vapnik, V. and Izmailov, R. Learning using privileged information: similarity control and knowledge transfer. *Journal of machine learning research*, 16(2023-2049):2, 2015.

Welinder, P., Welling, M., and Perona, P. A lazy man's approach to benchmarking: Semisupervised classifier evaluation and recalibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3262–3269, 2013.