Bayesian Methods of Machine Learning 2020 1–8

Super-Samples from Kernel Herding

Georgii Novikov

Abstract

Project report on reproducing the paper "Super-samples from Kernel Hearding" Chen et al. (2012) – paper about kernel version of herding procedure, that iteratively builds a subset, empirical distribution of which approximates the true distribution, and that has a rate of convergence $\mathcal{O}(T^{-1})$.

1. Introduction

The considered paper Chen et al. (2012) is a follow-up paper in the chain of several papers (Chen and Welling (2010), Welling and Chen (2010), Welling and Bren, Welling) which develop the so-called "herding" procedure. It is a method of generating pseudo-samples which resembles original data and could be used for dataset compression, estimate quantities of interest, compression of model ensembles. The key benefit of the herding procedure is that it has an $\mathcal{O}(T^{-1})$ convergence rate, which is significantly better than $\mathcal{O}(T^{-\frac{1}{2}})$ convergence rate of i.i.d samples.

In the paper of interest, the kernel version of herding algorithm was described. Then, it was proved, that the same asymptotic guarantees still hold for the kernelized version. And then the developed procedure was tested on several model tasks.

In this report, I first describe the herding procedure and how to obtain the kernelized version of it in section 2. Then in section 3, I reproduce the experiments from the original paper. In section 4, I describe my criticism regarding the paper and sum up my work done.

2. Theory

2.1. Herding

Let $\mathcal{X} \subset \mathbb{R}^n$ is some space with a probability distribution $p(\boldsymbol{x})$ on it, $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{\phi}(\boldsymbol{x}) : \mathcal{X} \to \mathcal{H}$ is some feature map function to the Hilbert space $\mathcal{H}$ with inner product $\langle \cdot ; \cdot \rangle$. Then herding is a weakly-chaotic, non-linear dynamical system, consisted of the following update equations for a weight-vector $\boldsymbol{w} \in \mathcal{H}$ (Chen and Welling (2010), Welling and Chen (2010)):

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \arg\max_{\boldsymbol{x} \sim \mathcal{X}} \langle \boldsymbol{w}_t, \boldsymbol{x} \rangle \\ \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t + \mathbb{E}_{\boldsymbol{x} \sim p} \left[\boldsymbol{\phi}(\boldsymbol{x}) \right] - \boldsymbol{\phi}(\boldsymbol{x}). \end{aligned} \tag{1}$$

For easy of intuitive understanding of herding, under some assumptions, the iterative procedure (1) could be viewed as a greedy optimization of the squared error ε_T^2 defined as

$$\varepsilon_T^2 = \left\| \mu_p - \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\phi}(\boldsymbol{x}) \right\|^2, \tag{2}$$

where $\mu_p = \mathbb{E}_{\boldsymbol{x}\sim p}\,\boldsymbol{\phi}(\boldsymbol{x})$. The key result about herding is that if we manage to find optimal $\boldsymbol{x}_t$ on each iteration of (1), then the error in (2) decreases at a rate of $\mathcal{O}(T^{-1})$. It is a very nice result, because, for instance, randomly generated i.i.d samples would have a $\mathcal{O}(T^{-1})$ rate of convergence, and samples, produced by MCMC, would have positive autocorrelation and thus have even worse rate of convergence.

2.2. Kernel Trick

In order to apply herding procedure (1), we have to store state vector $\boldsymbol{w}$ explicitly, which means, that we can not work in an infinite-dimensional Hilbert space. To overcome this, we can make a standard kernel trick: substitute explicit function $\boldsymbol{\phi(x)}$ with an implicit one, given by the kernel function $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}')\rangle$. Making an assumption, that $\boldsymbol{w}_0 = \mu_p$, we can rewrite (1) in the following way:

$$\begin{aligned} \boldsymbol{x}_{T+1} &= \arg\max_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{w}_0, \boldsymbol{\phi}(\boldsymbol{x})\rangle + T\,\mathbb{E}_{\boldsymbol{x}'\sim p}\left[k(\boldsymbol{x}, \boldsymbol{x}')\right] - \sum_{t=1}^{T} k(\boldsymbol{x}, \boldsymbol{x_t}) \\ &= \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\boldsymbol{x}'\sim p} - \frac{1}{T+1}\sum_{t=1}^{T} k(\boldsymbol{x}, \boldsymbol{x}'). \end{aligned} \tag{3}$$

It can be shown, that kernel herding is performing greedy minimization of

$$\begin{aligned} \varepsilon_T^2 &= \left\| \mu_p - \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\phi}(\boldsymbol{x}) \right\|^2 \\ &= \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim p}\left[k(\boldsymbol{x}, \boldsymbol{x'})\right] - \frac{2}{T}\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}\sim p}\, k(\boldsymbol{x}, \boldsymbol{x_t}) + \frac{1}{T^2}\sum_{t,t'=1}^{T} k(\boldsymbol{x}_t, \boldsymbol{x}_{t'}). \end{aligned}$$

The error measures the distance between p and the empirical distribution $\hat{p}_T(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} \delta(\boldsymbol{x}, \boldsymbol{x}')$, and the kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ determines how we should measure distances between distributions. The central result of the observed paper is that kernel herding still has $\mathcal{O}(T^{-1})$ rate of convergence.

3. Experiments

3.1. Toy example

To begin with, authors show how herding works on a toy 2D mixture of Gaussians. On a Figure 1 you can see the difference between first 20 herded samples and 20 i.i.d samples. All modes are getting attentions and samples are distributed overall in a "pleasant way".

3.2. Matching the True Distribution

A mixture of 100 5-dimensional Gaussian distributions was generated. Error was computed on four functions: the first three moments, and a non-linear function. Firstly average value

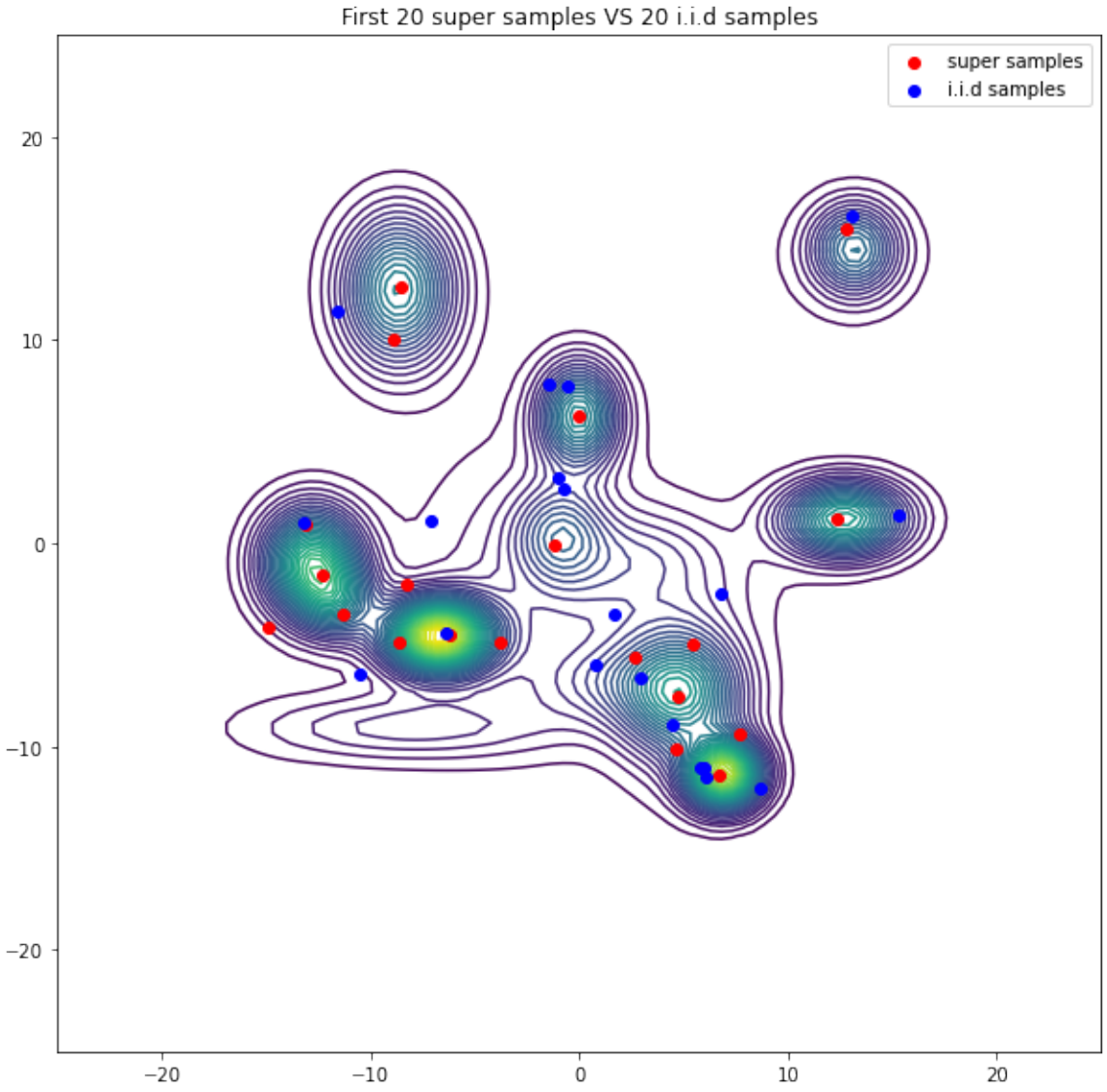

Figure 1: Comparision of 20 super-samples (obtained by herding) and 20 i.i.d samples for mixture of 10 2-dimensional Gaussians

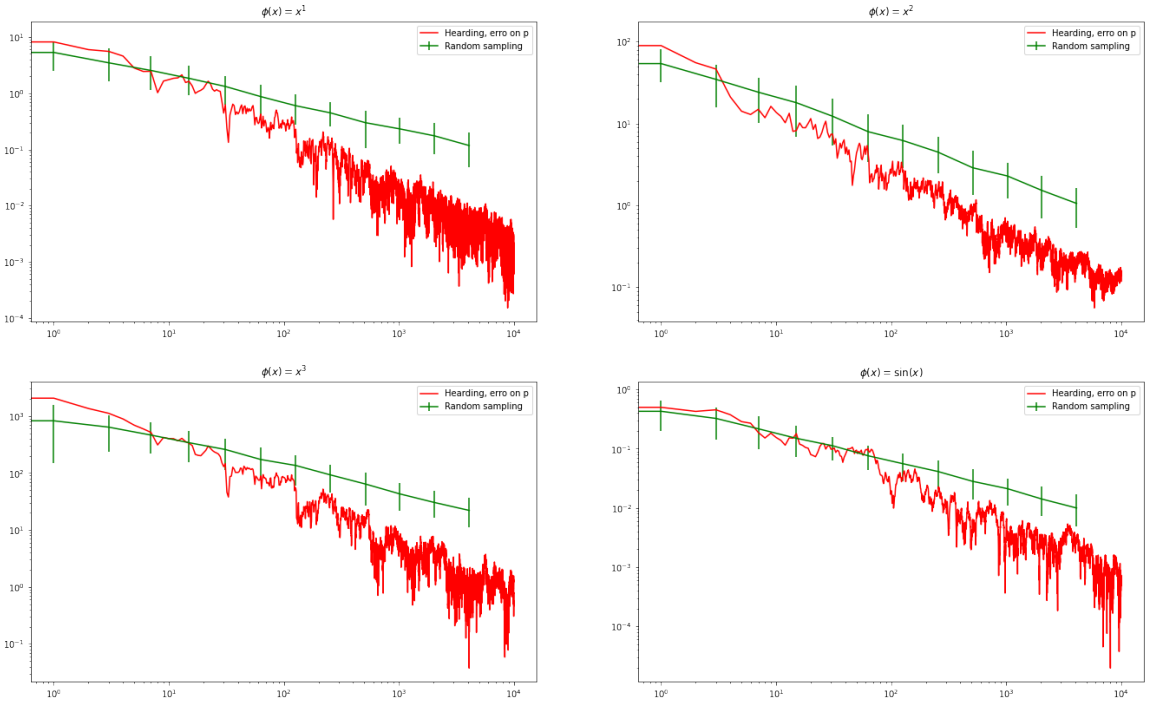

Figure 2: Dependence of approximation error (4) on the number of samples for Gaussian Mixture. Red line is samples acquired by herding. Green line is i.i.d samples from mixture.

of $f(x_{i,t})$ was calculated (where $f(x) = x^m, m = 1, 2, 3$ for moments and $f(x) = \sin(x)$ for sine function) and then the RMSE of the estimated value was computed:

$$err(\mathcal{S}_T) = \left(\frac{1}{d} \sum_{i=1}^{d} \left(\langle f(x_i) \rangle_{\int_T} - \langle f(x_i) \rangle_p \right)^2 \right). \tag{4}$$

The results are shown on the Figure 2.

3.3. Matching Empirical Distribution

The expectation in (3) can not be computed explicitly in the general case. But if we have a set $\mathcal{D}$ of i.i.d samples from the distribution, we can run herding to match the empirical distribution. A set of 10^5 samples is drawn from the mixture and then used as a true distribution for herding. In this case expectation in (3) is just an average over all samples. The estimation of function is again compared between herding and random samples. On Figure 3 we can see, that only a couple of thousands super samples is required to achieve similar approximation quality of the true mixture quantity.

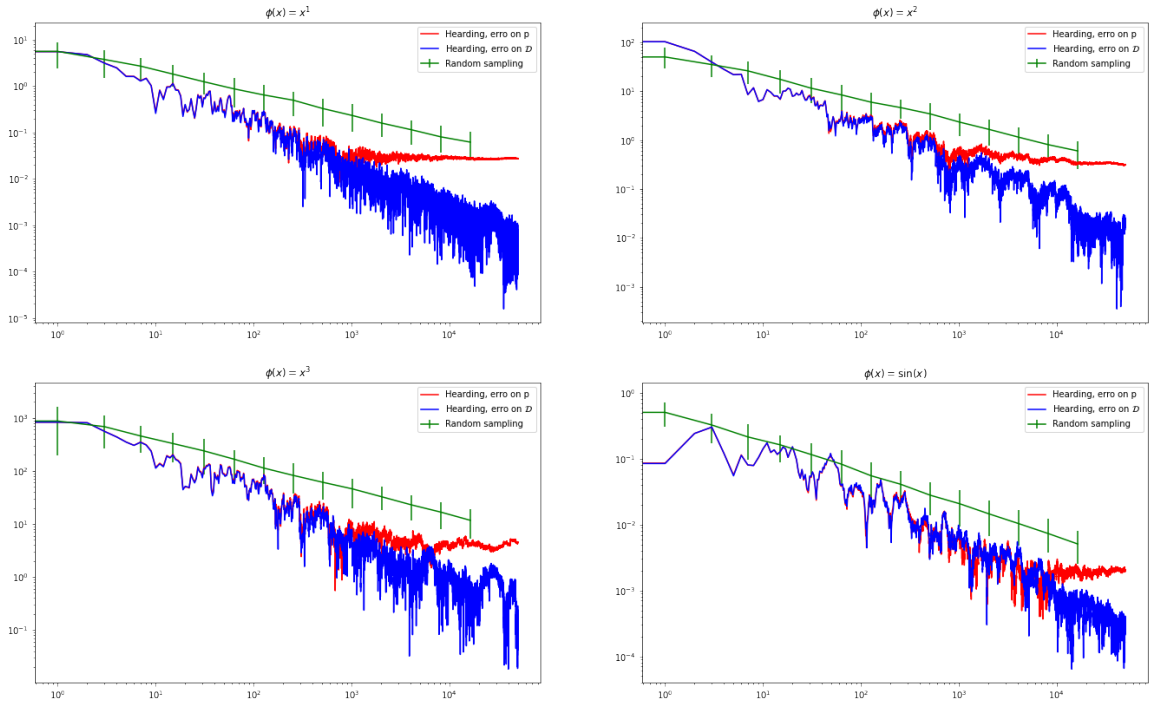

Figure 3: Dependence of approximation error (4) on the number of samples for Gaussian Mixture in the empirical distribution matching setup. Red and blue lines are samples acquired by herding for the approximating true value (red) and empirical value (blue). As the empirical value goes to the zero with an increasing number of herded elements, red line stagnates on several thousands (as we can not approximate better than the initial sample pool $\mathcal{D}$. Which means, that we can significantly compress the dataset $\mathcal{D}$ without huge quality loss. Green line is i.i.d samples from mixture (measured on true distribution).

3.4. Approximating the Bayesian Posterior

The final experiment in the original paper is an approximating the predictive distribution of a Bayesian model. Authors took the UCI spambase dataset that has 4601 instances (which were split into 3000 instances for train and 1601 for test) with 57 real attributes and 1 binary class label. The training set was whitened by a PCA and then fed into logistic regression model with Gaussian prior to draw a set of 10^7 samples by Metropolis-Hasting procedure with Gaussian proposal distribution. And the resulting set was subsampled by a factor of 100 to reduce the autocorrelation. This set $\mathcal{D}$ was whitened with PCA and fed into herding procedure with an isotropic Gaussian kernel with $\sigma = 10$. The quality of the resulting set $\mathcal{S}_T$ of supersamples was tested with the following error:

$$\mathrm{RMSE}^2(\mathcal{S}_T, \mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{T}\sum_{t=1}^{T} p(y_n|x_n,\theta_t) - \frac{1}{|D|}\sum_{d=1}^{|D|} p(y_n|x_n,\theta_d)\right]^2$$

For comparision, random samples was drawn from $\mathcal{D}$ by bootstrapping and computed error in the same way to model the performance of the i.i.d samples. Also, the much larger set of $2.5 * 10^6$ posterior samples were drawn in the same way as $\mathcal{D}$ to estimate the error of $\mathcal{S}_T$ on p by $\mathrm{RMSE}(\mathcal{S}, p)$ (the red line on Figure 4). With that we can see that a couple of thousands of super-samples are enough to achieve similar predictive distribution approximation quality.

4. Discussion

In this project I was managed to successfully reproduce all the experiments, that was presented by the authors in the original paper. All pictures in this report are plotted by me, the code for exact reproduction could be found in the jupyter notebook file in https://github.com/PgLoLo/kernel-herding.

Paper does not contain all the required information: a lot of small details are missing: parameters of Gaussian Mixtures, an exact algorithm of generating mixtures, parameters of Gaussian kernels for kernel herding, parameters of logistic regression prior and exact Metropolis-Hasting proposal distribution. But anyway, pretty much everything was reproducible with some guessing involved.

An important moment, about which I did not manage to find information in the paper is the dependence on the hyperparameters of the kernel for kernel herding. Namely, it turned out, that the scale of the Gaussian kernel has a great influence on the resulting graphics of the approximation quality. On the Figure 5 you can see the counterpart of the Figure 3 – the repetition of the same empirical distribution matching experiment, but with Gaussian kernel with $\sigma = 1$. As you can see, we do not see the same superiority of herding over i.i.d samples. That fact added difficulties to the reproduction of the paper, but with trial and error everything worked out in the end.

References

Yutian Chen and Max Welling. Parametric Herding. 2010.

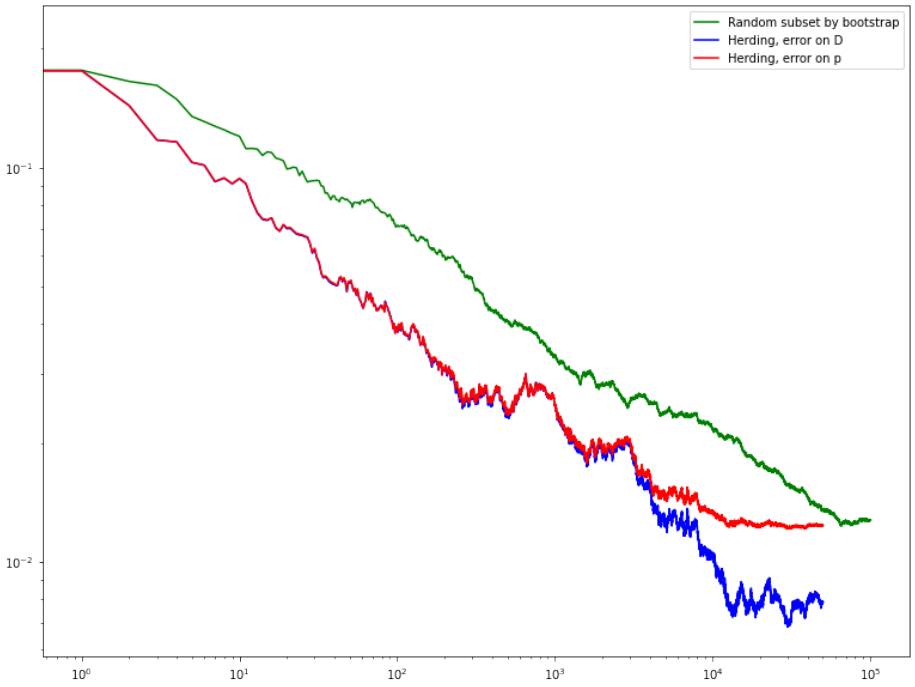

Figure 4: RMSE of the predicted probability of herding (blue - measured with respect to $\mathcal{D}$, red - measured with respect to much larger set, which imitates the true distribution p) and a random subset (green)

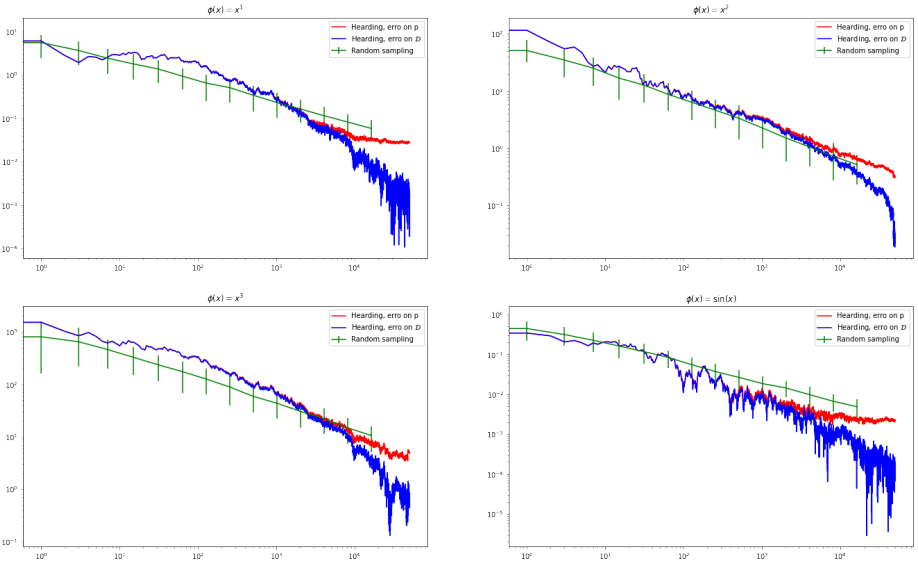

Figure 5: Repeated matching empirical distribution experiment (section 3.3 and Figure 3) with Gaussian kernel with $\sigma = 1$.

Yutian Chen, Max Welling, and Alex Smola. Super-Samples from Kernel Herding. mar 2012. URL http://arxiv.org/abs/1203.3472.

Max Welling. Herding Dynamic Weights for Partially Observed Random Field Models.

Max Welling and Donald Bren. Herding Dynamical Weights to Learn.

Max Welling and Yutian Chen. Conference Series OPEN ACCESS To cite this article: Max Welling and Yutian Chen. *J. Phys.: Conf. Ser*, 233:12005, 2010. doi: 10.1088/1742-6596/233/1/012005.