MNRAS **000**, 1–?? (2017) Preprint 14th July 2017 Compiled using MNRAS LATEX style file v3.0

Big Data vs. complex physical models: a scalable inference algorithm

J. Buchner[1,2,3]⋆
[1] *Millenium Institute of Astrophysics, Vicuña. MacKenna 4860, 7820436 Macul, Santiago, Chile*
[2] *Pontificia Universidad Católica de Chile, Instituto de Astrofísica, Casilla 306, Santiago 22, Chile*
[3] *Excellence Cluster Universe, Boltzmannstr. 2, D-85748, Garching, Germany*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

The data torrent unleashed by current and upcoming instruments requires scalable analysis methods. Machine Learning approaches scale well. However, separating the instrument measurement from the physical effects of interest, dealing with variable errors, and deriving parameter uncertainties is usually an after-thought. Classic forward-folding analyses with Markov Chain Monte Carlo or Nested Sampling enable parameter estimation and model comparison, even for complex and slow-to-evaluate physical models. However, these approaches require independent runs for each data set, implying an unfeasible number of model evaluations in the Big Data regime. Here we present a new algorithm based on nested sampling, deriving parameter probability distributions for each observation. Importantly, in our method the number of physical model evaluations scales sub-linearly with the number of data sets, and we make no assumptions about homogeneous errors, Gaussianity, the form of the model or heterogeneity/completeness of the observations. Our method has immediate application in speeding up analyses of large surveys, integral-field-unit observations, and Monte Carlo simulations.

Key words: methods: statistics – methods: data analysis

1 INTRODUCTION

Big Data has arrived in astronomy. In the previous century it was common to analyse a few dozen objects in detail. For instance, one would use Markov Chain Monte Carlo to forward fold a physical model and constrain its parameters. This would be repeated for each member of the sample. However, current and upcoming instruments provide a wealth of data (∼millions of independent sources) where it becomes computationally difficult to follow the same approach, even though it is embarrassingly parallel. Currently, much effort is put into studying and applying machine learning algorithms such as (deep learning) neural networks for the analysis of massive datasets. This can work well if the measurement errors are homogeneous, but typically these methods make it difficult to insert existing physical knowledge into the analysis, to deal with variable errors and missing data points, and generally to separate the instrument measurement process from the physical effects of interest. Furthermore, we would like to derive probability density distributions of physical parameters for each object, and do model comparison between physical effects/sources classes.

⋆ E-mail: johannes.buchner.acad@gmx.com

In this work I show how nested sampling can be used to analyse N data sets simultaneously. The key insight is that nested sampling allows effective sharing of evaluation points across data sets, requiring much fewer model evaluations than if the N data sets were analysed individually. I only assume that the model can be split into two components: a slow-to-evaluate physical model which performs a prediction into observable space, and a fast-to-compute comparison to the individual data sets (e.g. the likelihood of a probability distribution). Otherwise, the user is free to chose arbitrary physical models and likelihoods. In §3, I present as an example the line fitting of a hypothetical many-object spectroscopic survey. A more advanced example could include broadened H/O/C line emissions from various ionisation states under red noise errors, without modification of our algorithm.

2 METHODOLOGY

2.1 Introduction to Classic Nested Sampling

Nested sampling (Skilling 2004) is a global parameter space exploration algorithm, which zooms in from the entire volume towards the best-fit models by steadily increasing

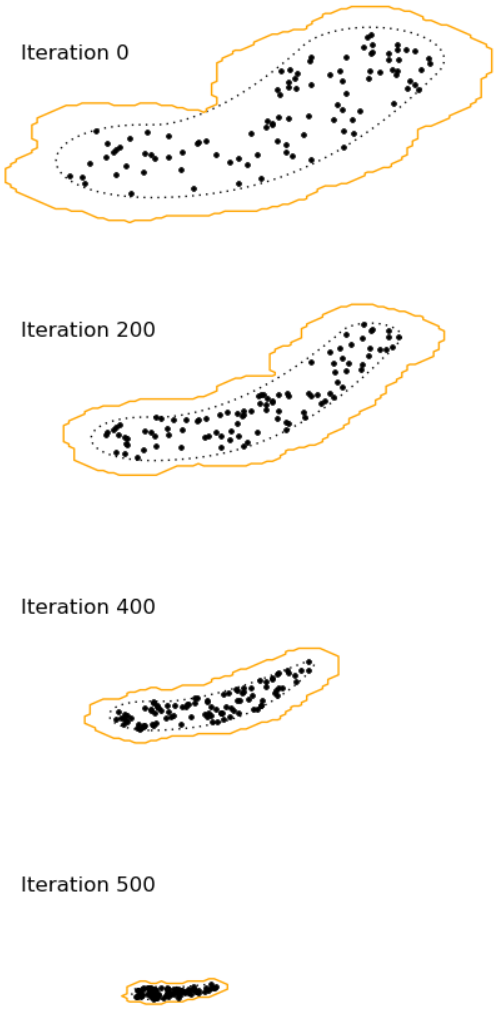

Figure 1. Illustration of nested sampling. At a given iteration of the nested sampling algorithm, the live points (black) trace out the current likelihood constraint, a region (dashed) which is unknown. The RadFriends algorithm conservatively reconstructs the region (orange) by including everything within a certain, adaptively chosen radius of the current live points. Between iterations, the likelihood contour is elevated, making the sampled volume smaller and smaller.

the likelihood threshold. In the process it produces parameter posterior probability distributions and computes the integral over the parameter space. Assume that the parameter space is a k-dimensional cube. A number of live points $N_{\rm live}$ are randomly[1] placed in the parameter space. Their likelihood is evaluated. Each point represents $1/N_{\rm live}$ of the entire volume. The live point with the lowest likelihood $L_{\rm min}$ is then removed, implying the removal of space with likelihood below $L_{\rm min}$ and shrinkage of the volume to $1 - \exp(-1/N_{\rm live})$, on average. A new random live point is drawn, with the requirement that its likelihood must be above $L_{\rm min}$. This replacement procedure is iterated, shrinking the volume exponentially. Each removed ("dead") point and its likelihood L_i is stored. The integral over the parameter space can then be approximated by $Z = \sum_i L_i \times w_i$, where w_i is the removed volume at the iteration. At a late stage in the algorithm the volume probed is tiny and the likelihood L_i increase is negligible, so that the weights $L_i \times w_i$ of the remaining live points becomes small. Then the iterative procedure can be stopped (the algorithm converged). The posterior probability distribution of the parameters is approximated as importance samples of weight $L_i \times w_i$ at the dead point locations, and can be resampled into a set of points with equal weights, for posterior analyses similar to those with Markov Chains. More details on the convergence and error estimates can be found in Skilling (2009).

Efficient general solutions exist for drawing a new point above a likelihood threshold in low dimensions ($n_{\rm dim} < 20$). The idea is to draw only in the neighbourhood of the current live points, which already fulfill the likelihood threshold. The best-known algorithm in astrophysics and cosmology is MultiNest (Shaw et al. 2007; Feroz et al. 2009). There, the contours traced out by the points are clustered into ellipses, and new points drawn from the ellipses. To avoid accidentally cutting away too much of the parameter space, the tightest-fitting ellipses are enlarged by an empirical (problem-specific) factor. Another algorithm is RadFriends (Buchner 2014), which defines the neighbourhood as all points within a radius r of an existing live point. By leaving out randomly a portion of the live points, and determining their distance to the remaining live points, the largest nearest-neighbour radius r is determined. The worst-case analysis through bootstrapping cross-validation over multiple rounds makes RadFriends robust, independent of contour shapes and free of tuning parameters. Figure 1 illustrates the generated regions. RadFriends is efficient if one chooses a standardised euclidean metric (i.e. normalise by the standard deviation of the live points along each axis). I use this algorithm in this work, although any other method can be substituted.

2.2 Simplified description of the idea

Consider two independent nested sampling runs on different data sets, but initialised to the same random number generator state. Initially points are generated from across the entire parameter space, typically giving bad fits. If the data sets are somewhat similar, this phase of zooming to the relevant parameter space will be the same for the two runs. Importantly, while the exact likelihood value will be different for the same point, the ordering of the points will be similar. In other words, for both, the worst-fitting point to be removed is the same. The next key insight is that new points can be drawn efficiently from a contour which is the union of the likelihood contours from both runs. Ideally, the point can be accepted by both runs, keeping the runs similar (black points in Figure 2). When a point is shared, the (slow) predicting model has to be only evaluated once, speeding up the run. The model prediction is then compared against the data to produce a likelihood for each data set, an operation which I presume to be fast (e.g. simply com-

[1] In general, following the prior. For most problems one can assume uniform sampling with appropriate stretching of the parameter space under the inverse cumulative of the prior distributions.

Figure 2. The analysis of two similar data sets yields at the same iteration similar likelihood contours (the two dotted ellipses). In the presented algorithm a large fraction of live points are shared across data sets (black points), which reduces the number of model evaluations. The differences (cyan crosses and magenta pluses) requiring additional draws.

puting $\mathcal{L}_j = -\sum_i (x_{ij} - m_i)^2 / 2\sigma_{ij}^2$ where $m_i / x_{ij} / \sigma_{ij}$ are the predictions/measurements/errors in data space respectively for data set j).

What if the point can be accepted by only one run? It cannot simply be rejected, otherwise the uniform sampling property of nested sampling is broken. Instead, accepted points are stored in queues, one for each run/data set. Once both runs have a non-empty queue, the first accepted point is removed from each queue and replaces the dead point of each data set. Joint sampling also helps even if a point is not useful right away. If a point was only accepted by one run, but the following point is accepted by both runs, the latter becomes a live point immediately for one run, but can later also become a live point for the other run (if it suffices the likelihood threshold at that later iteration). This technique allows sustained sharing of points, decreasing the number of unique live points and increasing the speed-up.

At a later point in the algorithm, the contours may significantly diverge and not share any live points. This is because the best-fit parameters of data sets will differ. Then, nested sampling runs can continue as in the classic case, without speed-up, falling back to a linear scaling. This happens earlier, the more different the data sets are. The run is longer for data sets with high signal-to-noise, making the algorithm most efficient when most observations are near the detection limit. This is typically the case in surveys (a consequence of powerlaw distributions).

2.3 Complete description of the algorithm

I now describe the algorithm for simultaneously analysing N data sets, where N is a large number. The algorithm components are the nested sampling integrator, the constrained sampler and the likelihood function, as in classic nested sampling, except that work on N data sets. The constrained sampler behaves substantially different in our algorithm. A proof-of-concept reference implementation is available at https://github.com/JohannesBuchner/massivedatans/.

2.3.1 Likelihood Function

The likelihood function receives a single parameter vector, and information which data sets to consider. It calls the physical model with the parameter vector to compute into a prediction into data space. The physical model may perform complex and slow numerical computations/simulations at this point.

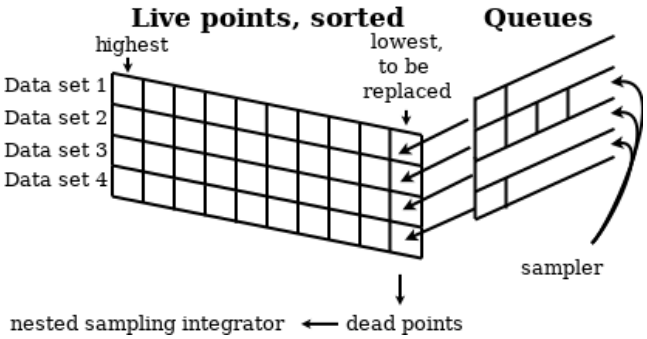

Figure 3. To replace the least likely live point, new points are sampled and placed in queues if they have a high enough likelihood. Once every data set has a non-empty queue, the lowest points are pushed out and stored as dead points by the integrator.

Finally the prediction is compared with the individual data sets to produce a likelihood for each considered data set. The likelihood at this point can be Gaussian (see above), Poisson (comparing the predicted counts to observed counts), a red noise Gaussian process, or any other probability distribution as appropriate for the instrument. In any case, this computation must be fast compared with producing the model predictions to receive any performance gains.

2.3.2 Nested Sampling Integrator

The integrator essentially deals with each run individually as in standard nested sampling, keeping track of the volume at the current iteration, and storing the live points and their weights for each data set individually. It calls the constrained sampler (see below), which holds the live points, to receive the next dead point (for all data sets simultaneously). The integrator must also test for convergence, and advance further only those runs which have not yet converged. Here I use the standard criterion that the nested sampling error is $\delta Z < 0.5$ (from last equation in Skilling 2009). Once all runs have terminated, corresponding to each data set the integral estimates Z and posterior samples are returned, giving the user the same output as e.g. a MULTINEST analysis.

2.3.3 Constrained Sampler

The sampler initially draws $N_{\rm live}$ live points and stores their likelihoods in an array of size $N \times N_{\rm live}$. Sequential IDs are assigned to live points and the mapping between live point IDs and data sets ($N \times N_{\rm live}$ indices) is stored. The integrator informs the sampler when it should remove the lowest likelihood point and replace it. The integrator also informs the sampler when some data sets have finished and can be discarded from further consideration, in which case the sampler works as if they had never participated.

The main task of the constrained sampler is to do joint draws under likelihood constraint $L > L_{\rm min}$ to replace the lowest likelihood point in each of the d data sets. For this, d initially empty queues are introduced (see Figure 3). First,

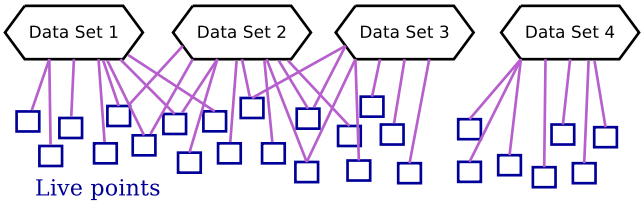

Figure 4. Association of live point objects with data sets. In this illustration, some live points are shared between the group of Data Set 1-3; these form a connected subgraph. Data Set 4 has separate live points and can be treated independently.

it is attempted to draw from the joint contour over all data sets (*superset draw*), i.e. letting RadFriends define a region based on the all unique live points. From this region a point is drawn which has $L > L_{\rm min}$ for at least one data set. Some will accept and the corresponding queues are filled. If this fails to fill all queues after several (e.g. 10) attempts, a *focussed draw* is done. In that case, only the data sets with empty queues are considered, the region is constructed from their live points, and the likelihood only evaluated for these data sets. For example, in the illustration of Figure 3, only Data Set 3 would be considered. Once all queues have at least one entry, nested sampling can advance: For each data set, the first queue entry is removed and replaces the dead live point. In Figure 3 this is illustrated by the queues pushing out the lowest live points. These dead points are returned to the integrator.

Storing queue entries is only useful if they can replace live points in the upcoming iterations. Playing nested sampling forward this implies that to be accepted into the end of the queue at position j, it must have a likelihood higher than j points from the runs live points and previous entries of the queue. In other words, the first entry must merely beat a single existing live point, the second entry must beat both a live point and either another live point or the first queue entry (which will become a live point in the next iteration).

2.3.4 Data Set Clustering

It can occur that between two groups of data sets the live points are not shared any more, i.e. the live point sets are disjoint. For example, one may have a dichotomy between broad and narrow line objects, and the contours identify some of the data sets in the former, some in the latter class. Then it is not useful to consider all live points when defining the region, because it introduces unnecessary multi-modality. Instead, subsets can be identified which share live points (see Figure 4), and these subsets can be processed independently. Algorithms for identifying connected subsets of graphs are well-known. The necessary graph can be constructed with nodes corresponding to the data sets, nodes corresponding to the live points, and connecting the graph according to the current live point statuses. This introduces some computational overhead, especially for large N and $N_{\rm live}$. However, one can lazily defer graph construction and maintenance until they are needed. A few simple checks can trivially rule out disjoint subsets: If there are fewer unique live points across all data sets than $2 \times N_{\rm live}$, some must be shared and there are no disjoint subsets. We can also track

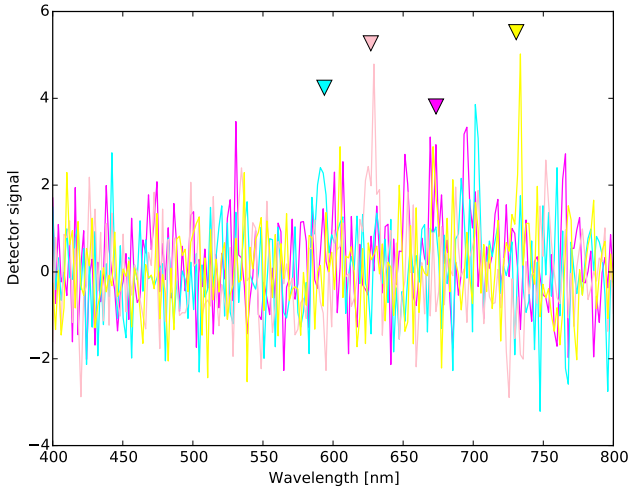

Figure 5. Simulated noisy data. The location, width and amplitude of a single line is sought in Gaussian noise for the illustrative problem. Here we show four spectra, with the true line locations indicated by triangles. The cyan data set shows a random fluctuation at 700nm.

any live point which has been accepted by all data sets. Any such a *superpoint*, until one data set removes it as a dead point, guarantees that there are no disconnected subsets.

The described analysis of divergence of contours effectively clusters data sets by similarity. Interestingly, the similarity is not defined in data space, nor in parameter space, but only through the constraints in parameter space. This is important because clustering in data space can be non-trivial for data with varying errors and completeness, and clustering in parameter space would scale poorly with model dimensionality because it is metric dependent. Instead, the likelihood ordering that makes nested sampling unique is taken advantage of. The clustering improves the efficiency of region draws, as it eliminates space between clusters, and improves super-set draws because unrelated data sets, which cannot benefit, are not pulled in.

3 RESULTS

A simple example problem illustrates the use and scaling of the algorithm. We consider a spectroscopic survey which collected N spectra in the $400-800$nm wavelength range. We look for a Gaussian line at 654nm rest frame (but randomly shifted) with standard deviation of 0.5nm. The amplitudes vary with a powerlaw distribution with index 3, and a minimum value of 2 in units of the Gaussian noise standard deviation. We generate a large random data set and analyse the first N data sets simultaneously to understand the scaling of the algorithm, with $N = 1$ to $N = 10^4$. Figure 5 presents some high and low signal-to-noise examples of the simulated data set.

The parameter space of the analysis has three dimensions: The amplitude, width and location of a single Gaussian line, with log-uniform/log-uniform/uniform priors from 10^{0-2}, $0.15-15$nm and $600-1000$nm respectively. The Gaussian line is our "slow-to-compute" physical model. The likelihood function is a simple product of Gaussian errors. A

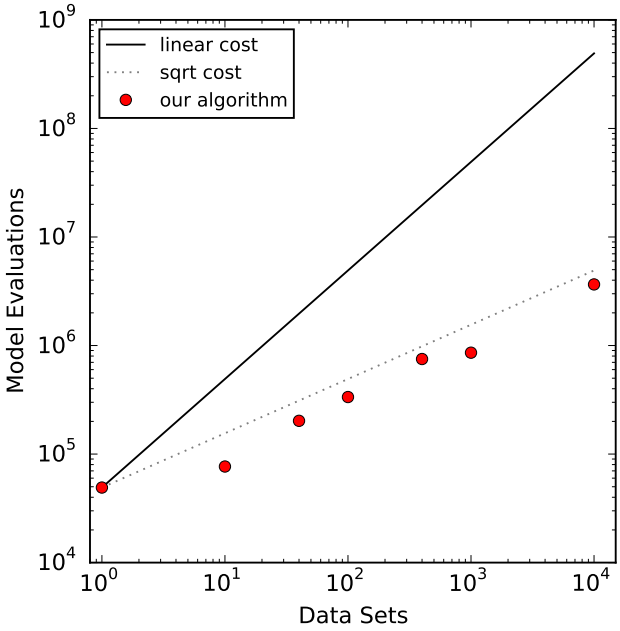

Figure 6. Number of model evaluations of the algorithm. A naive approach of independent analyses would have a linear scaling (black line). The algorithm (red points) scales substantially better, similar to $O(\sqrt{N})$ in the considered problem.

more elaborate example would include physical modelling of an ionised outflow emitting multiple lines with Doppler broadening and red detector noise, without necessitating any modification of the presented algorithm.

Figure 6 shows the number of model evaluations necessary for analysing N data sets. The algorithm scales much better than the baseline linear scaling, i.e. analysing the data sets individually one-by-one. For instance, it takes only twenty times more model evaluations to analyse 1000 observations than a single observations, a 50-fold speedup. Indeed, the algorithm scales in this problem much better than the naive linear cost $O(N)$ of parallel analyses.

We can now plot the posterior distributions of the found line locations. Figure 7 demonstrates the wide variety of uncertainties. The spectra of Figure 5 are shown in the same colours. For many, the line could be identified and characterised with small uncertainties (cyan, black), for others, the method remains unsure (pink, yellow, magenta, gray). Figure 8 shows that the input redshift distribution is correctly recovered.

After parameter estimation we can consider model comparison: is the line significantly detected? For this, lets consider the Bayes factor, $B = Z_1/Z_0$, where Z_1 is the integral computed by nested sampling under the single-line model, and Z_0 is the same for the null hypothesis (no line). The latter can be analytically computed as $\ln Z_0 = -\frac{1}{2}\left[\sum(x_i/\sigma_i)^2 + \ln 2\pi\sigma_i^2\right]$. Figure 9 shows in black the derived Bayes factors (truncated at 10^4). To define a lower threshold for significant detections, we Monte Carlo simulate a dataset with $N = 10,000$ spectra without signal, and derive Z_1 values. This can be done rapidly with the presented algorithm (gray squares in Figure 6). The red histogram in Figure 9 shows the resulting Bayes factors. The 99.9% quantile of B-values in this signal-free data set is

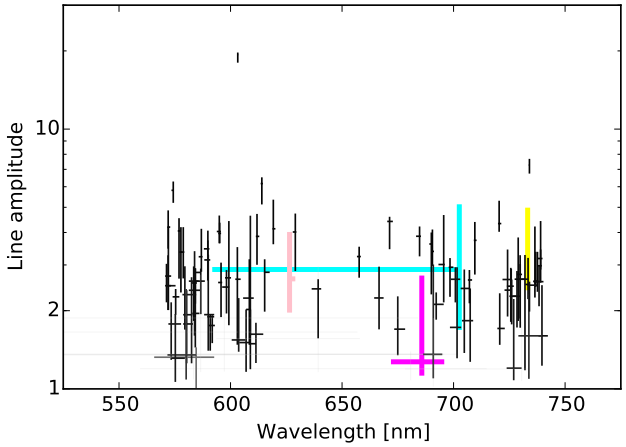

Figure 7. Parameter posterior distributions. The four examples from Figure 5 are shown in the same colours. Gray is used for the larger errorbars and black otherwise. The line in the pink and yellow data sets have been well-detected and characterized, while the magenta has larger uncertainties. The cyan constraints cover two solutions (see Figure 5). Error bars are centred at the median of the posteriors with the line lengths reflecting the 1-sigma equivalent quantiles.

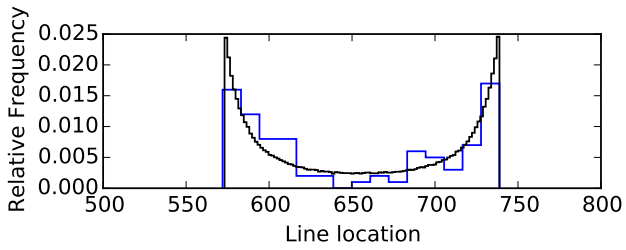

Figure 8. Line location distribution for objects where the line was well-constrained (blue) compared to the input distribution (black).

$B \approx 10$. Therefore, in the "real" data, those with a Bayes factor $B > 10$ can be securely claimed to have a line, with a small false detections fraction ($p < 0.001$).

4 DISCUSSION

A scalable algorithm for analysing massive data sets with arbitrarily detailed physical models and complex, inhomogeneous noise properties was presented. The algorithm brings to the Big Data regime parameter estimation with uncertainties, classification of objects and distinction between physical processes. A reference implementation is available at https://github.com/JohannesBuchner/massivedatans/.

The key insight in this work is to take advantage of a property specific to nested sampling: The contours at a given iteration are sub-volumes which can look similar across similar data sets, and it is permitted to draw new points from the union of contours[2]. These joint draws reduce drastically

[2] As in classic nested sampling, the volume shrinkage estimates are valid on average. Multiple runs can test whether this leads

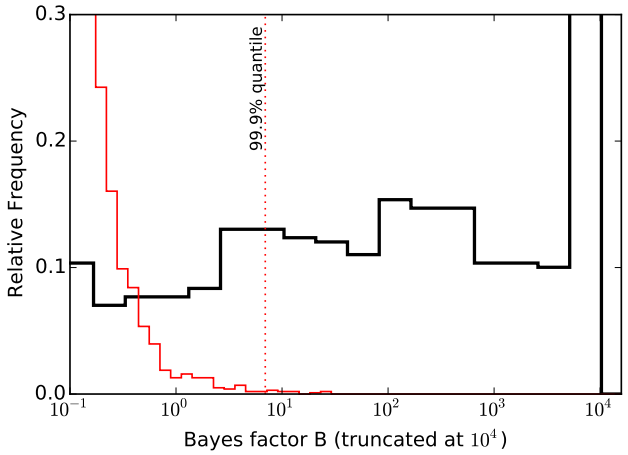

Figure 9. Bayes factors between the single-line model and a no-line model. The black histogram shows Bayes factors from analysing the test data set. The red histogram shows Bayes factors from a Monte Carlo data set without any signal. Because the latter has very few values above $B \gtrapprox 10$, for the black histogram at $B \gtrapprox 10$, a line can be claimed detected with a low false positive fraction.

the number of unique model evaluations, in particular at the beginning of the nested sampling run. The same approach cannot be followed with Markov Chain proposals: There, the following proposal depends on the current points, and different acceptances prohibit a later joint proposal. The management of many nests suggests *Bird Colony Exploration Algorithm* as a name.

The algorithm has some overhead related to the management of live points, in particularly to determine the unique set of live points across a dynamically selected subgroup of data sets. The memory usage also grows if big data sets have to be held in the same machine. If only chunks of N are managable, the analyses can be split into such sizes and analysed in parallel across multiple machines. In that case, one can take advantage of the scaling of the algorithm until N.

Applications: The algorithm can be applied immediately to any existing large data sets, such as spectra from the Sloan Digital Sky Survey (Eisenstein et al. 2011) which inspired the example presented here. Low signal-to-noise data or exhaustive searches for lines with multiple solutions are addressed in the algorithm. Aside from surveys with large data sets, individual integral-field-unit observations, where each pixel contains a spectrum, can be analysed with the algorithm, permitting also the fitting of complex ionisation or radiative transfer models. There the optimal speed-up case for the algorithm is satisfied, because often many pixels will share lines with similar positions and widths. Compared to existing approaches, nested sampling naturally allows model comparisons (e.g. detection of additional lines) and multiple fit solutions.

As another example, *eROSITA* (Predehl et al. 2014) requires the source classification and characterisation of 3 million point sources in its all-sky survey (Kolodzig et al. 2012). The desire to use existing physical models, the complex detector response and non-Gaussianity of count data make standard machine learning approaches difficult to apply. Furthermore, standard fitting techniques can fail to correctly converge and individual visual inspection in the Big Data regime is impractical.

Even in the analysis of single objects the presented algorithm can help. One might test the correctness of selecting a more complex model, e.g. based on Bayes Factors, as in the toy example presented. Large Monte Carlo simulations of a null hypothesis model can be quickly analysed with the presented method, with a model evaluation cost that is essentially independent of the number of generated data sets, i.e. comparable to the single-object analysis.

to additional scatter in the integral estimate. In practice, single runs already give correct uncertainties for many problems.

ACKNOWLEDGEMENTS

I thank Surangkhana Rukdee and Frederik Beaujean for reading the manuscript.

I acknowledge support from the CONICYT-Chile grants Basal-CATA PFB-06/2007, FONDECYT Postdoctorados 3160439 and the Ministry of Economy, Development, and Tourism's Millennium Science Initiative through grant IC120009, awarded to The Millennium Institute of Astrophysics, MAS. This research was supported by the DFG cluster of excellence "Origin and Structure of the Universe".

References

Buchner J., 2014, Statistics and Computing, pp 1–10
Eisenstein D. J., et al., 2011, AJ, 142, 72
Feroz F., Hobson M. P., Bridges M., 2009, MNRAS, 398, 1601
Kolodzig A., Gilfanov M., Sunyaev R., Sazonov S., Brusa M., 2012, preprint, (arXiv:1212.2151)
Predehl P., et al., 2014, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. p. 1, doi:10.1117/12.2055426, http://proceedings.spiedigitallibrary.org/pdfaccess.ashx?ResourceID=7241319&PDFSource=24
Shaw J. R., Bridges M., Hobson M. P., 2007, MNRAS, 378, 1365
Skilling J., 2004, in AIP Conference Proceedings. p. 395, http://proceedings.aip.org/resource/2/apcpcs/735/1/395_1
Skilling J., 2009, in BAYESIAN INFERENCE AND MAXIMUM ENTROPY METHODS IN SCIENCE AND ENGINEERING: The 29th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering. pp 277–291, http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.3275625