Proceedings of the International Workshop on Statistical-Mechanical Informatics
March 7–10, 2010, Kyoto, Japan

Statistical Inference using Weak Chaos and Infinite Memory

Max Welling and Yutian Chen

Donald Bren School of Information and Computer Science, University of California Irvine
CA 92697-3425 USA

E-mail: welling@ics.uci.edu, yutian.chen@uci.edu

Abstract. We describe a class of deterministic weakly chaotic dynamical systems with infinite memory. These "herding systems" combine learning and inference into one algorithm, where moments or data-items are converted directly into an arbitrarily long sequence of pseudo-samples. This sequence has infinite range correlations and as such is highly structured. We show that its information content, as measured by sub-extensive entropy, can grow as fast as $K \log T$, which is faster than the usual $\frac{1}{2} K \log T$ for exchangeable sequences generated by random posterior sampling from a Bayesian model. In one dimension we prove that herding sequences are equivalent to Sturmian sequences which have complexity exactly $\log(T+1)$. More generally, we advocate the application of the rich theoretical framework around nonlinear dynamical systems, chaos theory and fractal geometry to statistical learning.

1. Introduction

There are two theoretical frameworks for statistical inference: the frequentist and the Bayesian paradigm. A frequentist assumes a true objective value for some parameter and tries to estimate its value from from samples. Except for the simplest models, estimation usually involves an iterative procedure where the value of the parameter is estimated with increasing precision. In information theoretic terms, this means that more and more information from the data is accumulated in more decimal places of the estimate. With a finite data-set, this process should stop at some scale because there is not enough information in the data that can be transferred into the decimal places of the parameter. If we continue anyway, we will overfit to the dataset at hand.

In a Bayesian setting we entertain a posterior distribution over parameters, the width of which determines the amount of information it encodes. In Bayesian estimation, the width automatically adapts itself to the amount of available information in the data. In both cases, the information contained in our parameter (distribution) can be estimated to leading order as $H(N) = \frac{1}{2} K \log(N)$ where K is the number of parameters and N is the number of data-cases. Note that the learnable information in the data grows sub-linearly (or sub-extensively).

The learning process itself can be viewed as a dynamical system. For a frequentist this means a convergent series of parameter estimates $w_1, w_2, ...$ For a Bayesian running a MCMC procedure this means a stochastic process converging on some equilibrium distribution. In this paper we introduce a third possibility especially suitable for Markov random field models for which maximum likelihood (ML) learning and even more so Bayesian posterior inference is highly intractable. The idea is to model the correlations in data by a deterministic weakly

chaotic nonlinear dynamical system which was inspired by taking a zero temperature limit of the ML gradient update rules. Unlike ML learning however, the weights never converge but trace out a quasi-periodic trajectory on an attractor set which is often found to be of fractal dimension.

In this paper we study this dynamical system using the tools developed for chaotic systems. We estimate its topological entropy and find the characteristic behavior of weak chaos: $H(T) = K\log(T)$. We then relate this quantity to the sub-extensive entropies of the frequentist and Bayesian methods and find to our surprise that is can be higher by a factor of two. This remarkable result is explained by the fact that herding generates highly structured sequences with infinite range negative auto-correlations. This infinite memory effect renders the herding samples more efficient in encoding the information present in the data and can have implications for designing better MCMC methods.

2. Herding Dynamics

We will first introduce the herding system in its simplest form and show its equivalence to some well-studied theories in mathematics. Next, the system will be generalized and its relevance to machine learning explained in more detail.

2.1. A single Neuron Model

Consider a single (artificial) neuron, which can take on two distinct states: either it fires ($s = 1$) or it doesn't fire ($s = 0$).[1] Assume, we want this neuron to represent an irrational number, $p \in [0, 1]$, by firing at a rate that is asymptotically equal to p on average: $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} s_t = p$. One way to achieve that is to introduce a "synaptic strength", w, which keeps track of the current error. Assume that at time $t = 0$ we start with some arbitrary initial value w_0. By using the following updates for w and s we achieve the averaging property above:

$$s_{t+1} = \mathbb{I}[w_t > \alpha], \tag{1}$$

$$w_{t+1} = w_t + \gamma(p - s_{t+1}) \tag{2}$$

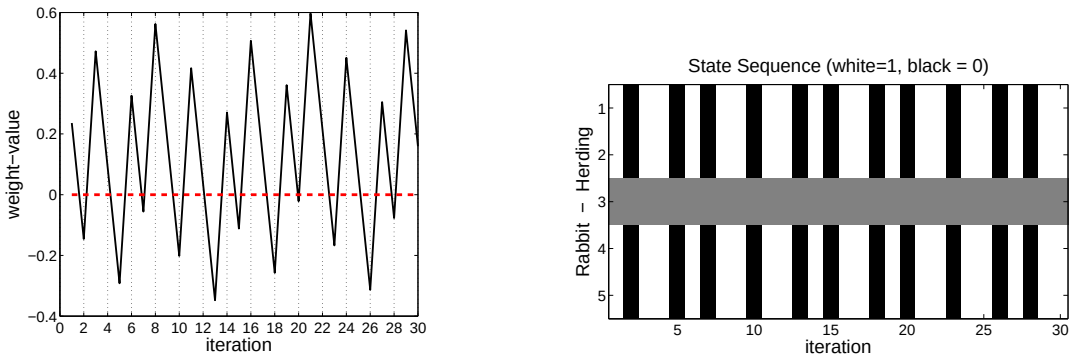
where $\mathbb{I}[\cdot]$ is the indicator function (equal to 1 if the argument is true and 0 if not). These updates are easily interpreted: at each iteration the synaptic strength increases by an amount γp. When the synaptic strength rises above a certain threshold, α, the neuron fires. If it fires its synaptic strength is depressed by a factor γ. The actual values of α and γ will only have some effect on the transient behavior of the resulting sequence $s_1, s_2, ...$ (in fact, for two different herding systems α, γ and α', γ' you can find initial conditions w_0 and w_0' such that the resulting sequences are identical). In Figures 1 we plot the weights (a) and the states (b) resulting from herding with a 1-neuron model. One may think of the synaptic strength as an error potential that keeps track of the total error so far.

It is perhaps interesting to note that by setting $p = \phi$ with ϕ the golden mean $\phi = \frac{1}{2}(\sqrt{5}-1)$ and initializing the weights at $w_0 = 2\phi - 1$, we exactly generate the "Rabbit Sequence": a well studied Sturmian sequence which is intimately related with Fibonacci numbers[2]). For a proof, please see Appendix A.

We can view this as a dynamical system on the state space s, i.e. $s_{t+1} = F_t(s_{[1:t]}; w_0)$. We note that the map depends on the entire history, $\{s_1, ..s_t\}$. This same information is equivalently

[1] Physicists may want to think of (classical) spin systems.

[2] Imagine two types of rabbits: young rabbits (0) and adult rabbits (1). At each new generation the young rabbits grow up ($0 \to 1$) and old rabbits produce offspring ($1 \to 10$). Recursively applying these rules we produce the rabbit sequence: $0 \to 1 \to 10 \to 101 \to 10110 \to 10110101$ etc. The total number of terms of these sequences and incidentally also the total number of 1's (lagged by one iteration) constitutes the Fibonacci sequence: $1, 1, 2, 3, 5, 8, ...$



(a) (b)

Figure 1. (a) Sequence of weight values for the "Fibonacci neuron" based on herding dynamics. Note that the state results by checking if the weight value is larger than 0 (in which case $s_t = 1$) or smaller than 0 (in which case $s_t = 0$). By initializing the weights at $w_0 = 2\phi - 1$ and using $p = \phi$, with ϕ the golden mean, we obtain the Rabbit sequence (see main text). (b) Top stripes show the first 30 iterates of the sequence obtained with herding. For comparison we also show the Rabbit sequence below it (white indicates 1 and black indicates 0). Note that these two sequences are identical as we show in Appendix A.

stored in the current value of the parameter w_t. However, marginalizing over w, the system has infinite memory. One can also compute the complexity of the state sequences, which is defined as the total number possible sequences of length T. This number turns out to be exactly $T + 1$, which is the absolute bare minimum for sequences that are not eventually periodic. These facts imply that our neuron model generates Sturmian sequences for irrational values of p which are precisely defined to be the non-eventually periodic sequences of minimal complexity (Lu & Wang, 2005). (For a proof, please see Appendix A.)

2.2. A Hopfield Network Model

Now let's generalize the one neuron model to a network of N neurons (a Hopfield network (Hopfield, 1982)). Each neuron can be in a firing or non-firing state ($s_i = 0/1$) and with each neuron we associate a dynamic weight, w_i, as before but we also introduce dynamic weights (or synapses) between two neurons, w_{ij} . The update equations for this entire network become:

$$s_{i,t+1} = \mathbb{I}\left[\sum_{j \setminus i} w_{ijt} s_{jt} + w_{it} > \alpha_i\right] \tag{3}$$

$$w_{i,t+1} = w_{it} + \gamma_i(p_i - s_{i,t+1}) \tag{4}$$

$$w_{ij,t+1} = w_{ijt} + \gamma_{ij}(p_{ij} - s_{i,t+1}s_{j,t+1}). \tag{5}$$

The first equation sums up all the synaptic weights from all neighboring, spiking neurons and adds its own weight. It emits a spike when the total is larger than a threshold. This equation is typically run until all states, s, have stabilized. After that, we perform updates on the synaptic strengths as indicated. Once again, synapses grow linearly in strength but may get instantaneously depressed when the neurons associated with that synapse fire. It is not hard to show that this type of dynamics guarantees that averages over trajectories will converge to their

associated population averages:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} s_{it} = p_i, \tag{6}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} s_{it} s_{jt} = p_{ij}. \tag{7}$$

Here, $\{p_i\}$ and $\{p_{ij}\}$ represent the probability of a single neuron firing and the probability of a pair of neurons firing respectively. These values should be mutually consistent and can for instance be obtained as averages over data-cases.

To guarantee that the weights in herding dynamics remain contained in a bounded region of space (i.e. do not run away to infinity), we need to find the exact maximizing argument of the Hopfield energy at every iteration. This is arguably hard in most cases. However, we claim (empirically for now) that we can replace exact maximization with approximate local maximization, initializing at the final state of the previous iteration, and still obtain excellent results. This approximate version is in fact the algorithm described above.

One way to understand dynamical synapses is as bookkeeping devices: they keep track of the number of times a state was sampled relative to other states. When a state is under-sampled it will attract the system to this state and when it is over-sampled it will repel the system away from this state. In this sense, the dynamics is similar to the weakly chaotic system of (Aihara, 1994) with destabilizing attractors.

2.3. General Features

We now generalize one step further and extend herding dynamics to the case of general features. In the examples described above we have been using the features $f_i(s_i) = s_i, f_{ij}(s_i s_j) = s_i s_j$, but there is no reason to restrict ourselves to that. In the following we will use general features denoted with $f_\alpha(S_\alpha)$, where the sub-indices fulfill the dual role of labeling both feature functions and (possibly overlapping) subsets of variables (e.g. $f_5(S_5) = f_5(s_1, s_4, s_8)$). We start by writing an energy function as follows: $E(S, w) = -\sum_k w_\alpha f_\alpha(S_\alpha)$. Herding dynamics is now defined through the repeated application of the following update equations:

$$S_{t+1} = \arg\min_S E(S, w_t), \tag{8}$$

$$w_{\alpha,t+1} = w_{\alpha,t} + < f_\alpha >_P - f_\alpha(S_{\alpha,t+1}), \tag{9}$$

where P is the probability distribution from which the data was sampled (the empirical distribution can be used instead). For tractability, we can again replace the minimization by a local minimization initialized at the final state of the last iteration.

This system shares similar properties with the Hopfield model described above. In particular we can show that under certain mild conditions the weights stay contained in a finite region of parameter space and the average of the features over the samples S_t will converge to the population average.

2.4. Hidden Variables

The models described above have the property that the generated state sequence reproduces the average of the sufficient statistics of the data. This can sometimes be restrictive in terms of the capacity to model arbitrary higher order statistical dependencies. A well-known method to model higher order dependencies is to introduce hidden variables, i.e. variables that are not directly observed in the data-sequence. The main question is whether we can extend herding dynamics to incorporate hidden variables as well.

The answer turns out to be affirmative. The only thing we have to change is the term $< f_\alpha >_P$ in the herding updates. Instead, it has to be replaced with the term $< f(Z^*, X) >_{P(X)}$, where $S = (Z, X)$ and Z^* is the hidden state that minimizes the energy,

$$Z^* = \arg\min_Z E(X, w, Z) \tag{10}$$

In practice we use for $P(X)$ the empirical distribution. This implies that at every iteration for every data-case separately we impute the most likely value for Z by minimizing the energy, and then average over all data-cases. (For more details see (Welling, 2009a).)

Conceptually, the situation has changed relative to the fully observed model of the previous sections. In particular, in those cases we only needed to store a small number of sufficient statistics and we could subsequently delete the actual dataset. In other words, the model was simply given in terms of these average sufficient statistics and we provided a way to probe this model through the generation of samples. For hidden variables we seem to actually have to store all the data-cases for our herding dynamics to run. We note that this might still be a useful exercise because the Z variables can often be interpreted as more abstract and semantically meaningful representations of the input data. These representations are often useful for subsequent processing such as classification or retrieval tasks. Moreover, the visible subset of pseudo-samples samples (X_t) can be used to interpolate between the data-samples. However, it is sometimes desirable to decouple the data altogether by replacing the data dependent terms $< f >_P$ with parameterized functions (i.e. a "model"). We have indeed performed a preliminary study and found that these functions can be effectively learned from the data. We have shown that the herding system that emerges from that process can be used to compress data and predict attributes of new data (Chen & Welling, 2010). In Figure 2 (c) we show a herding sequence trained on the digit "2" from the USPS digits dataset.

3. Learning in the Zero Temperature Limit

Consider the Kullback-Leibler divergence between the target distribution P_0 and the model distribution P_w (where w stands for parameters of the model),

$$KL[P_0||P_w] = -\sum_S P_0(S) \log P_w(S) - H[P_0] \tag{11}$$

The goal of learning would be to adjust the parameters w in such a way as to make this KL-divergence as small as possible.

In the following we will consider a Hopfield network (or Ising model) for definiteness, but the same conclusions hold for the more general systems described above. For these models we may write,

$$\begin{aligned} KL[P_0||P_w] &= < E(S; w) >_{P_0} + \log(Z) - H[P_0] & (12) \\ &= (< E(S; w) >_{P_0} - < E(S; w) >_{P_w}) - (H[P_0] - H[P_w]) & (13) \end{aligned}$$

with

$$E(S; w) = -(\sum_{ij} w_{ij} s_i s_j + \sum_i w_i s_i) \tag{14}$$

This represents a difference of two free energies, ΔF or "contrastive free energy" (Welling & Sutton, 2005).

Next, we introduce a temperature by substituting $E \to E/T$ for the energy. We will consider the quantity,

$$T\Delta F = \Delta E - T\Delta H \tag{15}$$

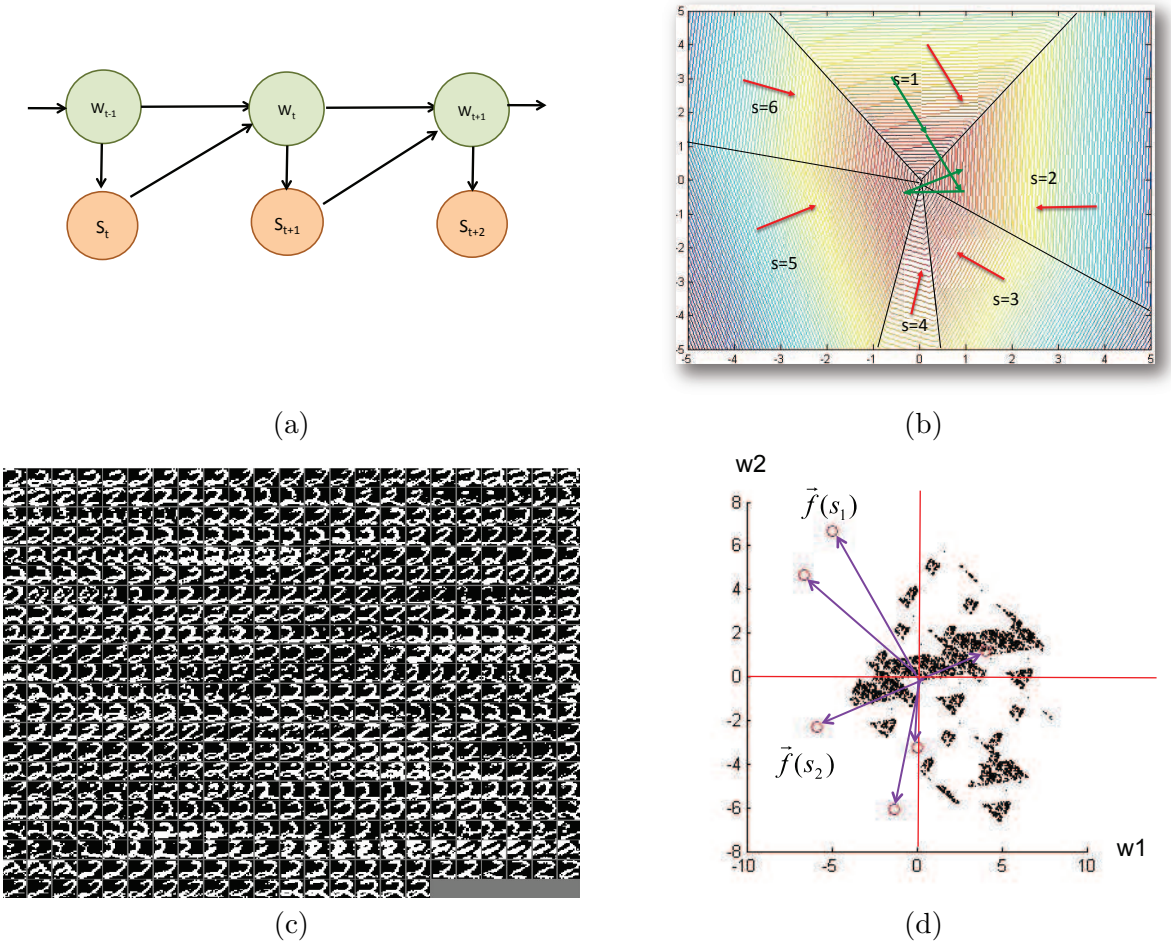


Figure 2. (a) Herding as a nonlinear dynamical system over the joint space (w, s). (b) Cones in parameter space $\{w_1, w_2\}$ that correspond to the discrete states $s_1, ...s_6$. Arrows indicate the translation vectors associated with the cones. (c) Sequence of "two's" generated using herding. (d) Fractal attractor set for herding with two parameters. The circles represent the feature-vectors evaluated the states $s_1, ...s_6$. Hausdorff dimension for this example is between 0 and 1.

and take the limit $T \to 0$. The result of this limit is then that all entropy terms vanish and averaging over P_w turns into maximization,

$$\ell_0 \doteq < E(S; w) >_{P_0} - \max_S E(S; w) \tag{16}$$

This function has a number of interesting properties (see (Welling, 2009b; Welling, 2009a)), namely: A) it is concave with a maximum at the origin, B) it is piecewise linear where the flat faces are formed by cones starting at the origin, C) it is scale free in that $\ell_0(\lambda \mathbf{w}) = \lambda \ell_0(\mathbf{w})$.

Using this function, we now propose to generate samples by alternating two steps: I) *maximize* ℓ_0 over $\mathbf{w}$ by taking a gradient step $\mathbf{w}_{t+1} = \mathbf{w}_t + \nabla_w \ell_0$ (It turns out that due to the scale-free property (C) the stepsize will not matter for the sequence S_t and will only scale $\mathbf{w}$-space by a constant factor.), II) *minimize* the energy $E(S, w)$ over S to compute the second term of

Eqn. 16. This is like performing gradient descent with a finite stepsize on a surface that looks like a "Tipi" tent. Because it will continuously overshoot the tip of the tent the sequence of weights will never converge to a fixed point. Instead we find an attractor set with (often) fractal dimension (see Figure 2 (d)). The sequence itself can be characterized as "weakly chaotic" as we will further explain in the next sections.

4. A Simple Example

For illustrative purposes only, we will now assume a finite discrete state space, $S = \{y_1, .., y_K\}$ and choose features: $f_k(S, y_k) = \delta(S, y_k),\ k = 1..K$. Every feature is also associated with a parameter, i.e the energy is given as:

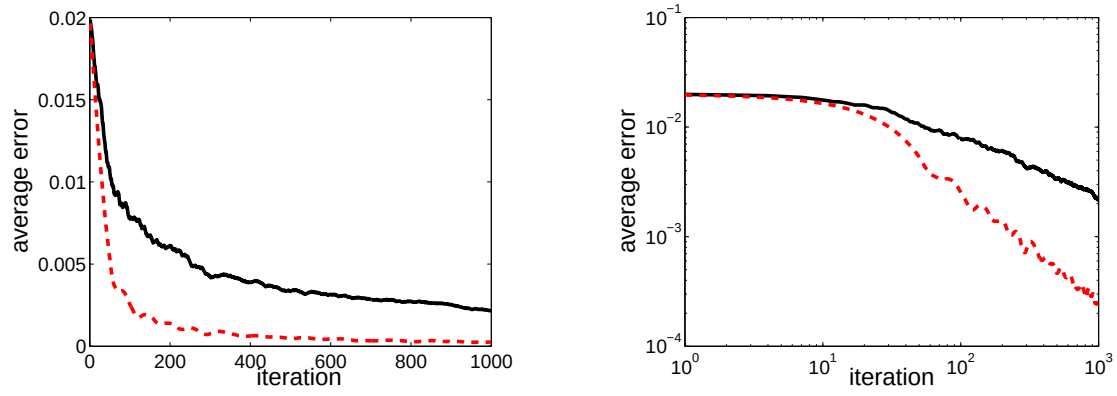
$$E(S; w) = -\sum_k w_k \delta(S, y_k) \tag{17}$$

Hence, every degree of freedom has its own feature and the model is unconstrained. We will denote the probabilities $P(S = y_k) = p_k$. In this situation the herding equations become,

$$S_{t+1} = \arg\max_S \sum_k w_{kt} \delta(S, y_k) \tag{18}$$

$$w_{k,t+1} = w_{k,t} + p_k - \delta(S_{t+1}, y_k) \tag{19}$$

In this case it is not hard to see that this will produce a sequence of states, S_t, where each symbol y_k will appear asymptotically with exactly the correct probability p_k. In figures 3 (a) and (b) we have compared the speed with which this sampling procedure converges onto the correct probabilities with drawing independent samples from the (correct) distribution. Interestingly, the above deterministic dynamical system has much better convergence properties.



(a) (b)

Figure 3. 100 probabilities were uniformly drawn between $[0, 1]$ and subsequently normalized. We then drew 1000 samples either independently from the correct distribution or using the method described in the main text (herding). We measured the mean absolute difference between the true probabilities and the estimated probabilities computed as averages over samples. The black solid curve represents IID sampling while the red dashed curve represent herding. Left figure (a) shows the errors as a function of iteration while the right plot (b) shows the same on a log-log plot. Clearly, the error converges much faster for herding both initially and asymptotically.

5. Herding as a Nonlinear Dynamical System

Herding can be viewed as a nonlinear dynamical system in K dimensions where K is the total number of parameters involved (Figure 2 (a)). In the previous section we learned that herding may be viewed as a series of translations where each discrete state s corresponds to one translation (i.e. $\rho(s) =< f >_P - f(s)$). Parameter space is partitioned into cones emanating from the origin (Figure 2 (b)). If the current location of the weights is inside cone i, then one applies the translation corresponding to that cone and moves along ρ_i to the next point. This system is an example of what is known as a piecewise translation (or piecewise isometry more generally) (Goetz, 2000).

It is clear that this system has zero Lyapunov exponents everywhere (except perhaps on the boundaries between cones but since this is a measure zero set we will ignore these). One can also show that that the evolution of the weights will remain bounded inside some finite ball (Welling, 2009b; Welling, 2009a) so the evolution will converge on some attractor set. Moreover, the dynamics is non-periodic in the typical case (more formally, the translation vectors must form an incommensurate (possibly over-complete) basis set; for a proof see Appendix B). It can often be observed that this attractor has fractal dimension (see Figure 2 (d) for an example). All these facts clearly point to the idea that herding is on the edge between full chaos (with positive Lyapunov exponents) and regular periodic behavior (with negative Lyapunov exponents). In fact, herding is an example of "weak chaos", which is usually defined through its (topological) entropy. We will study the entropy of herding in the next sections.

Herding is a dissipative system in the sense that there are many locations where a point has more than one pre-image. At these places it becomes obviously impossible to determine $\mathbf{w}_{t-1}$ from $\mathbf{w}_t$. In other words, we can not run the system backwards in time, i.e. it is *time irreversible*. We interpret these regions as "entropy sinks", i.e. regions where entropy is destroyed. These entropy sinks are full dimensional (i.e. of dimension K). On the other hand, entropy is also generated on the boundaries between the cones. We call these $K-1$ dimensional areas "entropy sources". It is here that a region gets split into two (or more) regions and the points in one half become completely uncorrelated with points in the other half.[3].

Note that in breaking a region in two we do not expand the volume of our space at a microscopic level. However at the sinks, regions overlap and we do indeed loose volume there. So, we can conclude that the volume will only decrease as time moves on. Whether the volume will become zero or reach a pedestal remains an open question. We conjecture that (under the right conditions) the volume will shrink to zero leading to fractal attractors[4].

Although the $K-1$-dimensional entropy sources do not generate K-dimensional volume at a microscopic level, they do generate volume at a coarse grained level by pushing apart the two halves of the region that it splits. It is this process which is usually referred to when we speak of the second law of thermodynamics, namely that the entropy of a system always increases. Starting with a small (infinitely dense) region we will indeed find that at a coarse grained level it will expand and converge on a stable attractor set. This despite the fact that it will loose K-dimensional volume. Again this picture suggests a fractal structure of the attractor set.

6. Topological Entropy

In this section we will estimate the entropy production rate of herding. This will inform us further of the properties of this system and how it processes information. We will first focus on topological entropy.

[3] Amusingly, herding therefore satisfies its own version of the "holographic principle" where entropy is generated on two-dimensional surfaces (such as black-hole horizons) and entropy gets destroyed in the three dimensional bulk of the universe.

[4] We have numerically verified this statement for some two dimensional cases.

From Figure 2 (b) we see that the sequence $s_1, s_2, ...$ can be interpreted as the symbolic system of the continuous dynamical system defined for the parameters $\mathbf{w}$. A sequence of symbols (states) is sometimes referred to as an "itinerary". Every time $\mathbf{w}$ falls inside a cone we record its label which equals the state s. The topological entropy for the symbolic system can be easily defined by counting the total number of subsequences of length T, which we will call $M(T)$. One may think of this as a dynamical language where the subsequences are called "words". The topological entropy is defined as,

$$h = \lim_{T\to\infty} h(T) = \lim_{T\to\infty} \frac{\log M(T)}{T} \tag{20}$$

It was rigorously proven in (Goetz, 2000) that $M(T)$ grows polynomially in T for general piecewise isometries, which implies that the topological entropy vanishes for herding. It is however interesting to study the growth of $M(T)$ as a function of T to get a sense of how chaotic its dynamics is.

To count the number of subsequences of length T, we can study the T-step herding map that results from applying herding T steps at a time. The original cones are now further subdivided into smaller convex polygons, each one labeled with the sequence $s_1, s_2, ..., s_T$ that the points inside the polygon will follow during the following T steps. Thus as we increase T, the number of these polygons will increase and it is exactly the number of those polygons which partition our parameter space that is equal to the number of possible subsequences. We first claim that every polygon, however small, will break up into smaller sub-pieces after a finite amount of time. This is proven in Appendix C. In fact, because we can show that in general every pair of points will break up as well, we infer that the diameter of the polygons must shrink. A partition with this property is called a *generating partition*.

Thus, to estimate the rate with which the number of sequences grow, we need to estimate how quickly a partition breaks up into smaller pieces. Define $c(t)$ to be the total number of cells (polygons) at time t. Recall that the cells split when they overlap with a cone-boundary. The probability that that happens is proportional to the diameter of the cell, which scales as $d(t) \sim c(t)^{-\frac{1}{K}}$ where K is the number of dimensions (which is equal to the number of herding parameters). This follows because $c(t) \approx (D/d(t))^K$ with D the size of the initial cell. At time t the probability of splitting a particular cell is $\pi \sim d(t)/L$ where L is the diameter of the volume over which the cells are distributed. Therefore, the change in the expected number of cells between time t and $t + \mathrm{d}t$ equals the number of cells at time t times the probability of a cell splitting: $\mathrm{d}c(t) = c(t)\pi(t)\mathrm{d}t$. Which then gives,

$$\mathrm{d}c(t) = c(t)\pi(t)\mathrm{d}t \sim c(t)d(t)\mathrm{d}t \sim c(t)^{1-\frac{1}{K}}\mathrm{d}t \tag{21}$$

which leads to the solution: $c(t) \sim t^K$. It has been rigorously proven that the growth rate must have an exponent less or equal than K (Goetz, 2000), but we conjecture based on the above argument and some preliminary numerical simulations that it is actual equal to K for herding in the typical case (a.k.a. with an incommensurate translation basis, see Appendix B). Thus, if we accept this conjecture, then the entropy $H(T) = Th(T)$ of a sequence of length T (for T large enough) is given by,

$$H(T) = K\log(T) \qquad \alpha \in (0, 1] \tag{22}$$

This result is interesting, because it implies that there may be more *learnable* information in a herding sequence than in a sequence of IID samples. We will return to this issue in section 8.

7. Dynamical Tsallis Entropy

Dynamical entropies are defined by first dividing space into small cells and initializing a large number of particles in 1 such cell. We subsequently measure (or estimate) the total accumulated

probability mass in each cell of our partition. The definition for discrete time systems is,

$$h_{\text{Dyn}} = \lim_{\ell \to 0} \lim_{T \to \infty} \lim_{N \to \infty} \frac{H(T) - H(0)}{T} \tag{23}$$

where N is the number of particles in the simulation, ℓ is the diameter of the cells in the partition and $H(T)$ counts the total entropy of the ensemble at time T. If we use the usual Shannon entropy $H_{\text{Shannon}}(t) = -\sum_i p_i(t) \log p_i(t)$ then the Kolmogorov-Sinai entropy follows. (Note that if we *count* the number of occupied cells instead of using the Shannon entropy we retrieve the topological entropy). From the previous section we know however, that $h_{\text{KS}} = 0$ because $H_{\text{KS}}(T)$ would grow logarithmically in T.

There is however an alternative definition of entropy specifically designed for systems with powerlaw growth in their complexity called the Tsallis entropy (Tsallis et al., 2006) defined as,

$$H^q_{\text{Tsallis}} = \frac{1 - \sum_i p_i^q}{q - 1} \tag{24}$$

The Tsallis entropy converges to the well known Shannon entropy as $q \to 1$. By demanding that the Tsallis entropy grows *linearly* in time (Tsallis et al., 2006; van Beijeren, 2004) we can compute the exponent q. For that we recall the result: $c(t) \sim t^K$. Thus, because the number of occupied cells grows as t^K, each cell itself will contain $p_{\text{occupied}} \sim t^{-K}$ probability mass. Therefore:

$$\sum_i p_i^q \sim t^K (t^{-K})^q \propto t \quad \Rightarrow \quad q = \frac{K - 1}{K} \tag{25}$$

This result is interesting because it tells us that as we increase the dimensionality, K, of the system, we will approach the Shannon entropy. Therefore, very high dimensional systems are expected to behave increasingly like random systems and will be more accurately described by a Gibbs distribution. This conclusion is indeed what we observe in learning systems with a very high number of variables.

8. Learning with Long Range Order

Consider the task of finding a binary sequence (a.k.a. a code) such that the mean of that sequence represents an irrational number p, i.e. $\frac{1}{T}\sum_{t=1}^{T} s_i = p$. The squared error is given as,

$$\text{error}^2 = (\frac{1}{T}\sum_{t=1}^{T} s_t - p)^2 \tag{26}$$

We now wish to find the sequence that has minimal error *at every iteration* t. It is not hard to show that at time t the optimal decision is given by,

$$s_t = \mathbb{I}\left[(p - \frac{1}{2}) - (\sum_{t=1}^{T-1} s_t - (t-1)p) > 0\right] \tag{27}$$

This sequence results from herding with an initial weight $w_0 = p - \frac{1}{2}$. (Note that this is different from the initialization of the Rabbit sequence by a factor of 2.)

Alternatively, one can generate IID samples from a Bernoulli distribution $P(s) = p^s(1-p)^{1-s}$. While independent samples from a Bernoulli distribution also converge to p, they do so at a slower rate (see Figure 3). This phenomenon can be understood by the fact that a herding sequence has infinite range auto-correlations while IID samples have zero auto-correlations. The long range auto-covariances of herding are depicted in Figure 4. We observe strong negative

correlations between time steps which are responsible for the improved convergence rate. Note that we have become used to the positive auto-correlations of MCMC sampling methods which have the opposite effect of decreasing the convergence rate to the mean. It's interesting that a deterministic dynamical system with long range order manages to improve the convergence rate.

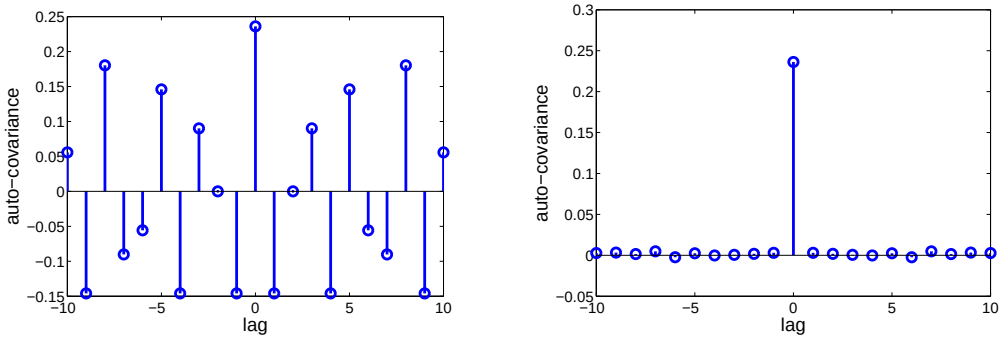


Figure 4. Autocovariance function for herding (left) and IID sampling (right) for the two-state single neuron problem of section 2.1 with $p = \phi$ the golden mean. We observe that the long range order with negative autocorrelation is causing the faster convergence to the (golden) mean.

Recall that in section 6 we estimated the dominant contribution to the entropy of a herding sequence to be $H_{\text{top}}(T) \sim K \log(T)$ with K the number of parameters. We can also compute the entropy of a *posterior* sample from a Bernoulli distribution where we will use a Beta-prior on the parameter p. This results in an exchangeable sequence of samples (no longer IID due to the fact we integrate over the p which induces dependencies). The dominant part of that sequence grows linearly in T, i.e. $H_{\text{ext}} = hT$. As explained in (Bialek et al., 2001) however, this extensive contribution will not contribute the predictive information defined as the mutual information between the future and past sequence. The authors argue that it is the sub-extensive part of the entropy that should be interpreted as the amount of predictive information about the future. In fact the authors go one step further in stating that sub-extensive entropy is a good measure of the *complexity* of the sequence where both regular and random sequences have low complexity. The dominant contribution of the sub-extensive part for posterior sampling can easily be computed as $H_{sub}(T) = \frac{1}{2} K \log T$ (with $K = 1$ for the Beta-Bernoulli distribution). This sub-extensive entropy is equal the minimum description length as it measures the number of bits required to encode the learnable (predictable) part of the sequence. It can also be derived as the *bits-back* term (Hinton & Zemel, 1994) in the Bayesian evidence but now applied to the parameters instead of the hidden variables.

Comparing to the entropy of herding[5] we can draw two important conclusions: I) The entropy of herding does not have an extensive component due to its long range order. Thus, all the information in a herding sequence is useful for prediction and no bits are wasted on unpredictable randomness. II) The entropy of herding can grow faster by a factor of two than the sub-extensive part of the entropy of the Bayesian evidence. From this we conclude that the samples of herding can be interpreted as a filtered version of samples obtained from sampling from the corresponding Bayesian model (a Beta-Bernoulli distribution in this case) where the useful signal has been separated from a noisy background. (The term "noisy" is being used in its more general interpretation as useless for prediction.)[6].

[5] We do not expect that the topological entropy and the Kolmogorov-Sinai entropy will differ significantly in their leading order behavior. The KS-entropy can be identified with the H_{sub}.

[6] It is interesting to note that the authors of (Bialek et al., 2001) comment on the equivalence of information

Herding becomes more practical, if we use it for higher dimensional learning problems like the ones discussed in sections 2.2,2.3 and 2.4. However, for these problems the number of possible states is exponentially larger than the number of parameters. We can therefore not hope to apply herding to all degrees of freedom of the problem (this would mean that we define one feature per state as in section 3). In this case it becomes important to think about the behavior of the moments which are not forced to converge to their true values. This will namely reflect our *inductive bias*. The usual approach without hidden variables is that of maximizing the entropy subject to the enforced constraints. This makes intuitive sense because we remain maximally ignorant about the unconstrained degrees of freedom. We have strong indications that herding does not follow a maximum entropy distribution. We are currently looking into alternative variational principles perhaps based on the Tsallis framework. The conclusion is thus that herding sacrifices the maximum entropy to some extend in order to achieve faster convergence on estimating some sufficient statistics relative to say posterior sampling from a hierarchical Bayesian model.

9. Conclusions

The information flow for herding is similar to that of Bayesian posterior sampling: We take in data and convert that into a sequence of parameters (sampled from the posterior $P(w|\text{data})$) and given these weights we can produce samples from the model $P(x|w)$. The overall effect of this is samples from $P(x|\text{data})$. However, Bayesian sampling is stochastic while herding is deterministic. This has interesting consequences: firstly, herding is more tractable than posterior sampling, particularly for Markov random field models. Secondly, herding has no extensive entropy leading to highly structured samples. This structure causes the sequence of samples to potentially carry more useful (predictable/learnable) information than posterior random samples. We believe that this observation underlies the success of the herding in real world learning problems as demonstrated in (Welling, 2009b) and (Welling, 2009a). In large state spaces, when the number of states is much higher than the number of moments, herding is no longer maximizing entropy for the unconstrained degrees of freedom. Elucidating the effective distribution from which herding is sampling is important because it relates to the inductive bias of the modeler.

Herding in Hopfield networks shows strong similarities to the relatively recently discovered dynamical weights in real neurons (Maass & Zador, 1998; Tsodyks et al., 1098; Pantic et al., 2002; Pfister et al., 2009). There are clear differences between herding and the phenomenon of depression in neurons, most markedly the fact that real neurons depress when the pre-synaptic neuron fires while "herding neurons" depress when both pre- and post-synaptic neurons fire. Yet, it can be argued in both cases that the depression is causing the dynamics to be on the edge between regular and chaotic behavior (Levina et al., 2007) which is often associated with high complexity and efficient information processing. Our results clearly underscore these conclusions and we believe that a good deal of progress is yet to be made in building networks of artificial neurons with depressing synapses. In this context the work of (Tieleman & Hinton, 2009) should also be mentioned where fast weights fulfil a similar role.

In summary, we believe that we have identified a problem that has components in physics, mathematics, cognitive science, neuroscience, statistics and machine learning and it could greatly benefit from the combined wisdom of these disciplines.

10. Aknowledgements

We thank Romain Thibaux, Anton Goredetski and Richard Palais for discussions and suggestions.

generated through parameter tying or through other types of infinite range correlations in the data.

Appendix A. All Sturmian sequences can be generated by herding

We wish to proof that any Sturmian sequence can be generated using herding with irrational probabilities p. We first recall the definition of Sturmian sequences. A sequence is Sturmian if it can be written as,

$$r_k = \lfloor (k+1)\gamma + \alpha \rfloor - \lfloor k\gamma + \alpha \rfloor, \qquad k = 1, ... \tag{A.1}$$

with $0 < \gamma < 1$ an irrational real and α real. Rewrite the second term as,

$$\lfloor k\gamma + \alpha \rfloor = \left\lceil \sum_{i=1}^{k-1} r_i \right\rceil + \lfloor \gamma + \alpha \rfloor. \tag{A.2}$$

Next we note that r_k can only take on values 0 or 1, and that we can equivalently write

$$r_k = \mathbb{I}\left[(k+1)\gamma + \alpha - \lfloor k\gamma + \alpha \rfloor > 1 \right], \qquad k = 1, ... \tag{A.3}$$

with $\mathbb{I}[\cdot]$ a indicator function. Combining with A.2 we get the condition,

$$r_k = \mathbb{I}\left[(k+1)\gamma + \alpha - \sum_{i=1}^{k-1} r_i - \lfloor \gamma + \alpha \rfloor > 1 \right], \qquad k = 1, ... \tag{A.4}$$

Lets consider a herding sequence with initial weight w_0. A herding sequence is given as,

$$s_k = \mathbb{I}\left[w_{k-1} > 0 \right] \tag{A.5}$$

$$w_k = w_{k-1} + \gamma - s_k \tag{A.6}$$

Hence, we find that,

$$w_k = w_0 + k\gamma - \sum_{i=1}^{k} s_i \tag{A.7}$$

$$s_k = \mathbb{I}\left[w_0 + (k-1)\gamma - \sum_{i=1}^{k-1} s_i > 0 \right] \tag{A.8}$$

where in the second line we used the first line. If we now compare this with Eqn.A.4 then we find that we need to identify,

$$w_0 = 2\gamma + \alpha - \lfloor \gamma + \alpha \rfloor - 1 \tag{A.9}$$

We conclude that for every Sturmian sequence there is a herding algorithm. However, there are more herding algorithms than Sturmian sequences because we can find initializations for which there is no corresponding for γ, α (note that γ) is fixed by the requirement that $p = \gamma$). However, the difference is only a transient effect and solely due to initialization. Finally, the Rabbit sequence (connected to the Fibonacci sequence) is obtained by choosing $p = \gamma = \phi$, $\alpha = 0$, and $w_0 = 2\phi - 1$ (with ϕ the golden mean).

Appendix B. Herding is non-periodic

We define the translation vector for every cone i to be $\boldsymbol{\varrho}_i$. It is given by

$$\boldsymbol{\varrho}_i = \langle \mathbf{f} \rangle_p - \mathbf{f}(s_i^*) \tag{B.1}$$

where s_i^* is the maximizing state in that cell. We call the collection of translation vectors a translation basis. Note that we assume that the number of basis vectors is at least equal to the number of dimensions. Herding will shift the current weight vector $\mathbf{w}$ over $-\boldsymbol{\varrho}_i$ for some (optimal) i.

Proposition: If there is no solution to the Eqn. $\sum_i n_i \boldsymbol{\varrho}_i = 0$ for $n_i \in \mathcal{N}$, $\forall k$, i.e. the translation vectors form an incommensurate basis, then every orbit is non-periodic.

Proof: Assume some orbit starting at $\mathbf{w}_0$ is periodic with period T. Then $\sum_{t=1}^{T} \delta\mathbf{w}_t = 0$, with $\delta\mathbf{w}_t = \mathbf{w}_t - \mathbf{w}_{t-1}$. However, by definition, $\delta\mathbf{w}_t = \boldsymbol{\varrho}_{\iota[t]}$ where $\iota[t]$ is the index of the translation vector picked at time t. Thus, if we denote with n_i the number of times cell i was visited in the finite time sequence under consideration, i.e. $n_i = \sum_{t=1}^{T} \mathbb{I}[\iota[t] = i]$ then we find that the following equation must hold for a periodic orbit: $\sum_{t=1}^{T} n_i \boldsymbol{\varrho}_i = 0$, which is excluded by the premise of the theorem. Contradiction.

Appendix C. Cells always split

Proposition: For herding with an incommensurate basis we have that any region, however small, will eventually split.

Proof: Consider a small region, R_0. Assume that it will never split. Then, because we know that any point will remain inside a compact region U, there is some finite time τ such that the region obtained after τ_2 iterations, R_{τ_2}, will overlap with some region generated at an earlier time R_{τ_1}. (The visited region will grow linearly in time with only a finite total volume available to move around in.) Call $R_{\tau_1} = R_1$ and $R_{\tau_2} = R_2$. Because the map is non-periodic (Appendix B), when the intersection happens, every element in the region R_2 will be translated by a vector $\mathbf{v} = \sum_i n_i \boldsymbol{\varrho}_i$ for some (finite) set of integers $\{n_i\}$. The conjoined region $R_{12} = R_1 \cup R_2$ must inherit the property that it will never split. However, after another $\tau_2 - \tau_1$ iterations, the regions R_1 has shifted again to R_2 and R_2 has shifted *over the same vector* $\mathbf{v}$ to R_3 (this follows because the region R_{12} as a whole cannot split and so R_1 and R_2 must follow the same sequence of translation vectors). Repeating this argument shows that the union of all these regions $R_{12..k}$ will grow *linearly*. Hence, since it cannot outgrow the compact region U, it will have to overlap with an edge at some finite time which causes the region to $R_{12..k}$ to split for some k. This constitutes a contradiction.

References

Aihara, K. (1994). Chaos in neural response and dynamical neural network models: toward a new generation of analog computing. In M. Yamaguti (Ed.), *Towards the harnessing of chaos*, 83–98. Elsevier, Science Publishers B.V., Amsterdam.

Bialek, W., Nemenman, I., & Tishby, N. (2001). Predictability, complexity and learning. *Neural Computation*, *13*, 2409–2463.

Chen, Y., & Welling, M. (2010). Parametric herding. *Proceedings of the Conference on Uncertainty in AI.*

Goetz, A. (2000). Dynamics of piecewise isometries. *Illinois Journal of Math*, *44:3*, 465–478.

Hinton, G., & Zemel, R. (1994). Autoencoders, minimum description length, and helmholtz free energy. *Neural Information Processing Systems.*

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*, 2554–2558.

Levina, A., Herrmann, J., & Geisel, T. (2007). Dynamical synapses causing self-organized criticality in neural networks. *Nature Physics*, *3*, 857 – 860.

Lu, K., & Wang, J. (2005). Construction of sturmian sequences. *J. Phys. A: Math. Gen.*, *38*, 2891–2897.

Maass, W., & Zador, A. M. (1998). Dynamic stochastic synapses as computational units. *Advances in Neural Information Processing Systems* (pp. 903–917). MIT Press.

Pantic, L., Torres, J., Kappen, H., & Gielen, C. (2002). Associative memory with dynamic synapses. *Neural Computation*, *14*, 2903–2923.

Pfister, J., Dayan, P., & Lengyel, M. (2009). Know thy neighbour: A normative theory of synaptic depression. *Advances in Neural Information Processing Systems – NIPS*.

Tieleman, T., & Hinton, G. (2009). Using Fast Weights to Improve Persistent Contrastive Divergence. *Proceedings of the International Conference on Machine Learning*.

Tsallis, C., Gell-Mann, M., & Sato, Y. (2006). Extensivity and entropy production. *Europhys. News*, *36*, 186–189.

Tsodyks, M., Pawelzik, K., & Markram, H. (1098). Neural networks with dynamic synapses. *Neural Computation*, *10*, 821–835.

van Beijeren, H. (2004). Generalized dynamical entropies in weakly chaotic systems. *Physica D*, *193*, 90–95.

Welling, M. (2009a). Herding dynamic weights for partially observed random field models. *Proc. of the Conf. on Uncertainty in Artificial Intelligence*. Montreal, Quebec, CAN.

Welling, M. (2009b). Herding dynamical weights to learn. *Proceedings of the 21st International Conference on Machine Learning*. Montreal, Quebec, CAN.

Welling, M., & Sutton, C. (2005). Learning markov random fields using contrastive free energies. *International Workshop on Artificial Intelligence and Statistics* (pp. 397–404).