Statistical Mechanics of Complex Systems

Application of Maximum Entropy and Network Analysis

Alberto Chimenti - 1210360

1 Dataset overview

The dataset reports a census (from 2005) of **S=299** alive plant species from Barro Colorado Forest in Panama.

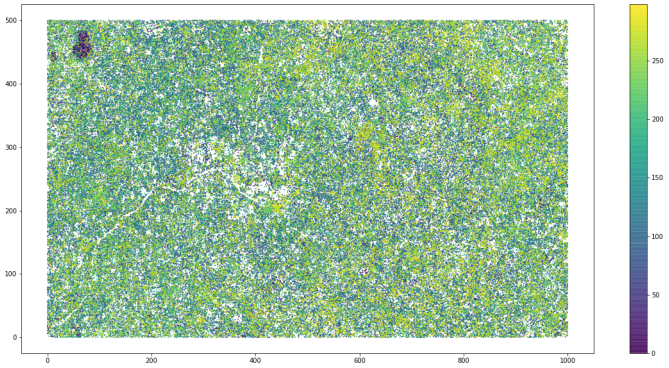

Figure 1: Dataset plot (the color grading is associated with species label)

The data was loaded in python using `pandas` library and only the alive species were selected among the full dataset. Information about species labels and their spatial location in the 50 hectare field were kept from the full dataset.

2 Data sub-sampling

The dataset was divided into **N=200** regions, each 0.25 hectares big, using boolean masks (see `code`). Assuming that the 200 subplots data are independent one from each other, the **abundances matrix** and the **average presence** p_i distribution were estimated over the samples.

The latter was defined as:

$$p_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij} \qquad where \quad p_{ij} = \begin{cases} 1 & if\ present \\ 0 & otherwise \end{cases}$$

3 Model: Maximum Entropy 1

The system was first modelled using the Maximum Entropy approach which lead us to the following expression of the PDF maximizing the entropy of the system:

$$P(\sigma|\lambda) = \frac{1}{\mathcal{Z}} e^{-\sum_{i=1}^{S}\lambda_i\sigma_i} \qquad with \quad \sigma_i = 2p_i - 1$$

By imposing the *normalization constraint*, we find the expression of the partition function:

$$\mathcal{Z} = \sum_{\{\sigma_i\}=\pm 1} e^{-\sum_{i=1}^{S}\lambda_i\sigma_i} = 2^S \prod_{i=1}^{S} cosh(\lambda_i)$$

The latter is equivalent to the partition function of a one-body interaction Ising Model.

This time by imposing $\langle\sigma_i\rangle_{emp} \simeq \langle\sigma_i\rangle_{model}$ as a constraint for our system, we obtain:

$$m_i = \langle\sigma_i\rangle_{emp} \simeq -\frac{1}{\mathcal{Z}}\frac{\partial\mathcal{Z}}{\partial\lambda_i} = tanh(\lambda_i)$$

one ends up with the relation for the *lagrange multipliers*

$$\lambda_i \simeq tanh^{-1}(m_i) \tag{1}$$

Since some of the species are present in every subplot ($m_i = +1$), a regularization term (10^{-16}) was introduced ad-hoc to avoid divergent values of the $\lambda_i s$.

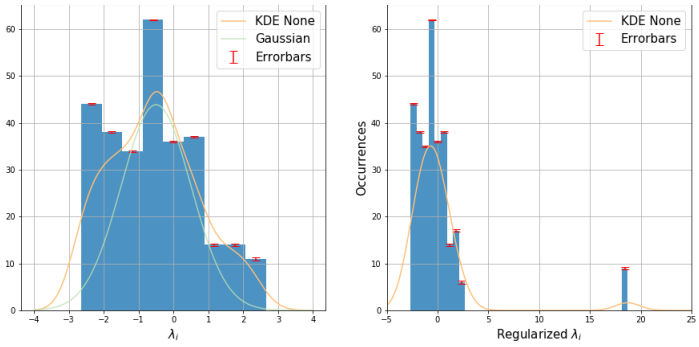

Figure 2: Lagrange multipliers distribution

The histograms are plotted along with their KDE. The Kernel Density Estimation was performed using Gaussian kernels. A Gaussian distribution was also plotted along with the unregularized parameters in order to underline possible similarities with their distribution.

4 Model: Maximum Entropy 2

The first model can be seen as equivalent (with $\beta = 1$) to the choice of an Ising Model with Hamiltonian equal to:

$$\mathcal{H}_1 = -\sum_{i=1}^{S}\lambda_i\sigma_i \tag{2}$$

In this section we extend the model using the *Mean-Field approximation* on the *Random Field Ising Model*:

$$\mathcal{H}_2 = -\frac{k}{S}\Big(\sum_{i=1}^{S}\sigma_i\Big)^2 - \sum_{i=1}^{S}\lambda_i\sigma_i \tag{3}$$

In this case, since there is no exact solution to the problem, the system was simulated using *Metropolis algorithm* and *stochastic gradient descent* (see `code`).

The 300 lagrange parameters have been initialized randomly, using a uniform distribution for k and a normal distribution for the $\lambda_i s$ following the results obtained from the previous model. Similarly a random distribution of the Ising spins was generated and one of them was randomly chosen and flipped at each step.

Then, for every step the transition probability was defined as:

$$P_t = \texttt{min}(1, e^{-(E'-E)})$$

In order to simplify the computation the ΔE estimation was calculated analitically (see Appendix):

$$E' - E = -2\lambda_k \sigma_k^{new} + 4\frac{k}{S}\sigma_k^{new}[\sigma_k^{new} - (s_+ + s_-)_{new}]$$

The index k represents the spin value which has been flipped.

After some tries, the number of "evolution steps" was set to 4000 and 400 more steps were performed for the mean computation. Those values were chosen as the best trade-off between energy stabilization and computational cost.

An example plot is shown in order to monitor the energy value of the system.

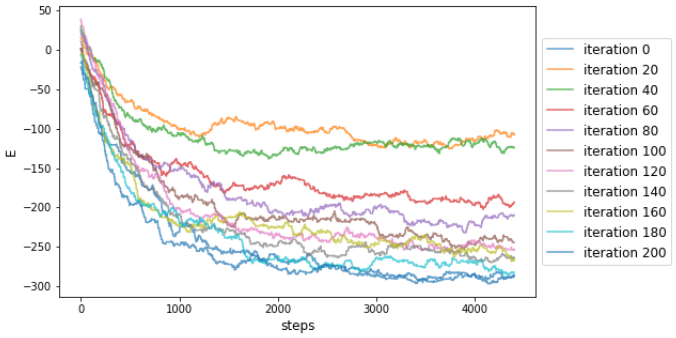

Figure 3: Energy evolution: each curve corresponds to a new update of the lagrangian parameters

Thereafter, using the simulation mean values obtained, the lagrange parameters were updated using *gradient descent* minimizing the following "gradients", taken as constrains for our new model:

$$\nabla\sigma_i = \langle\sigma_i\rangle_{model} - \langle\sigma_i\rangle_{emp}$$

$$\nabla\sigma_0 = \left\langle\left(\sum_{i=1}^{S}\sigma_i\right)^2\right\rangle_{model} - \left\langle\left(\sum_{i=1}^{S}\sigma_i\right)^2\right\rangle_{emp}$$

Several GD algorithms were implemented and among those the ADAM algorithm was chosen as the best one, leading to the following update rules:

$$(\lambda_i)_{t+1} = (\lambda_i)_t + \eta\frac{(\hat{\mathbf{m}}_i)_t}{\sqrt{(\hat{\mathbf{s}}_i)_t + \epsilon}}, \tag{4}$$

$$k_{t+1} = k_t + \eta\frac{(\hat{\mathbf{m}}_0)_t}{\sqrt{(\hat{\mathbf{s}}_0)_t + \epsilon}} \tag{5}$$

where, $(\hat{\mathbf{m}}_i, \hat{\mathbf{s}}_i, \hat{\mathbf{m}}_k, \hat{\mathbf{s}}_k)$ are defined in Appendix B along with the concerning parameters choice.

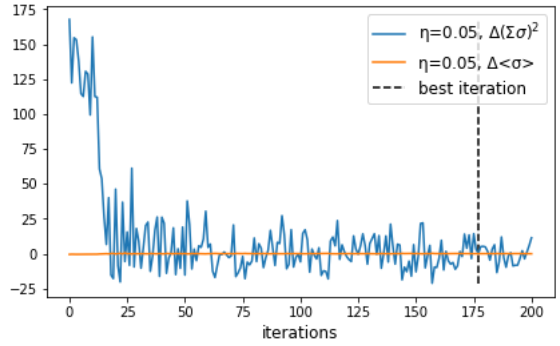

Figure 4: Stochastic gradients evolution

Several tries were performed for this section too, changing number of *update iterations* and *learning rate* η. The best model parameters were selected searching for the minimum of the two gradients:

$$t^* \rightarrow \mathbf{min}_{(over\ t)}\left\{|(\nabla\sigma_i)_t| + |(\nabla\sigma_0)_t|\right\}$$

By looking at Figure 4, it becomes evident that the quadratic gradient $\Delta\sigma_0$ revealed itself to be the dominant component in the choice of the best model.

The number of update iteration was set to 200 in order to let the gradients fluctuate and the best learning rate found was $\eta = 0.05$.

Subsequently we can see the evolution of the **k** parameter and the distribution of the λ_is reported.

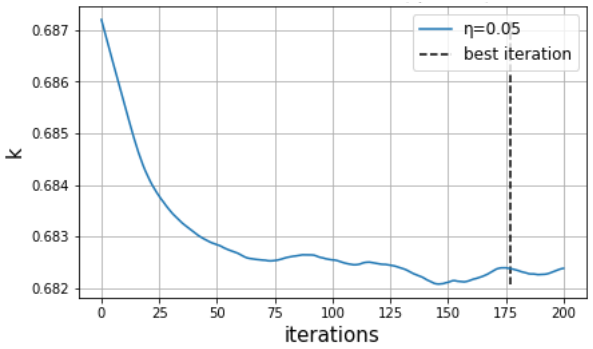

Figure 5: k parameter evolution (η=0.05)

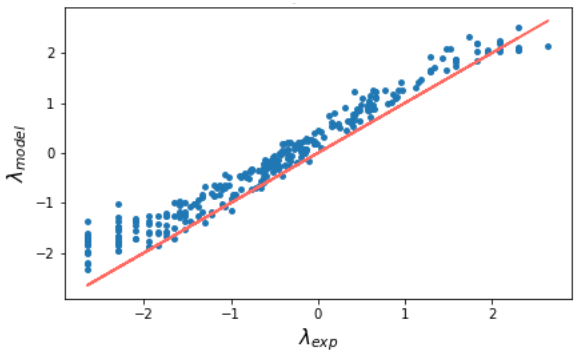

Figure 6: Distribution of the λ_is estimated in the best iteration

The distribution of the λ_is fits well the one from the first model but has mean shifted close to zero.

Finally another simulation was performed with the final parameters, this time with a smaller variance mean, which has been performed over 2000 samples after the "thermalization steps". The latter resulted in the following distribution of the magnetization values:

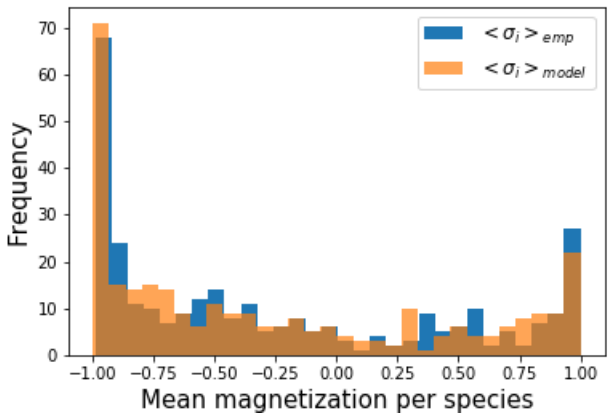

Figure 7: Distribution of the $\langle\sigma_i\rangle$

See the code for final parameters estimation values.

5 Phase Diagram

Assuming that the λ_is have a Gaussian distribution, we are able to find a relation between the parameter k and λ (See Schneider and E. Pytte).

Starting from the RFIM hamiltonian 3 we can calculate the mean free energy and the *self-consistent equation* for the magnetization.

$$\Big\langle F \Big\rangle_{\lambda} = S\Big[km^2 - \int d\lambda p(\lambda) ln2cosh\big[\beta(2km+\lambda)\big]\Big]$$

$$m = m_{SC}(m) := \int d\lambda p(\lambda) tanh\big[\beta(2km+\lambda)\big]$$

$$where \quad p(\lambda) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\Big(-\frac{\lambda^2}{2\sigma^2}\Big)$$

Starting from the trivial solution $m_{SC}(0) = 0$ the phase transition threshold is set to be $m'_{SC}(0) = 1$, which is equivalent to the following equation:

$$2k \int d\lambda \frac{p(\lambda)}{(\cosh(\lambda))^2} = 1$$

By substituting the PDF in the latter we find the useful relation for the phase transition threshold:

$$k = \frac{1}{2}\Big[\int \frac{d\lambda}{\sqrt{2\pi}\sigma} \frac{e^{\left(-\frac{\lambda^2}{2\sigma^2}\right)}}{(\cosh(\lambda))^2}\Big]^{-1} \tag{6}$$

Then using equation 6 the k value estimated in the previous section was plotted as a function of the standard deviation of the lambdas.

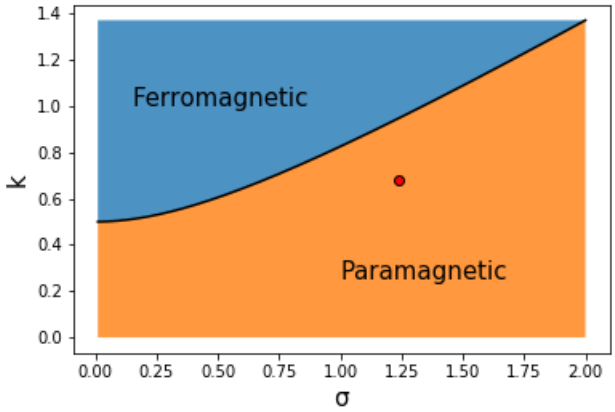

Figure 8: Phase diagram

We can see that the resulting model is in the paramagnetic phase.

6 Model: Maximum Entropy 3

Random pair interactions were finally considered leading to the following hamiltonian:

$$\mathcal{H} = \frac{1}{2}\sum_{i,j=1}^{S'} M_{ij} x_i x_j + \sum_{i=1}^{S'} \lambda_i x_i \tag{7}$$

In this case x_i indicates the abundances per species which assumes continuous integer values.

In order to be able to compute the partition function analitically we need to get rid of some mean abundances values by imposing the threashold $\langle x_i \rangle_{emp} > \sigma_{x,i}$, which leaves us with a population of $S' = 52$ species.

This time we impose $\Big(\langle x_i x_j \rangle_{emp} = \langle x_i x_j \rangle_{model}\,,\ \langle x_i \rangle_{emp} = \langle x_i \rangle_{model}\Big)$ as constraints to the model and find the following relations:

$$\lambda_i = -\sum_{j=1}^{S'} M_{ij} \langle x_i \rangle_{emp} \tag{8}$$

$$M_{ij}^{-1} = Cov(x_i, x_j) \tag{9}$$

By computing the values we found the following:

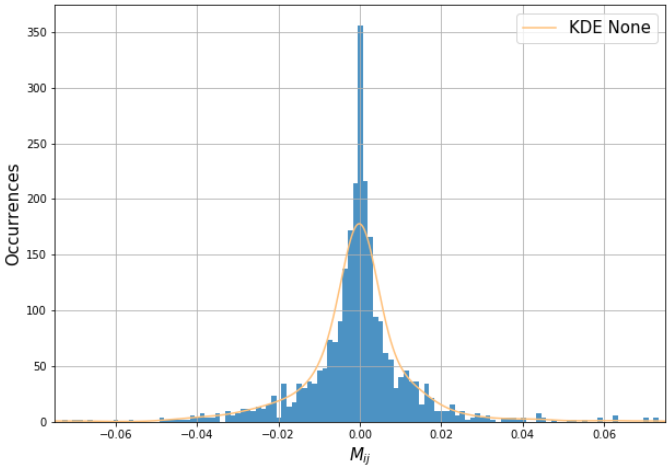

Figure 9: M_{ij} distribution

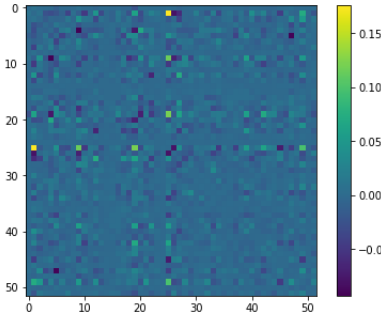

Figure 10: Interaction matrix

7 Network

The interaction matrix in Figure 10 was then used to build a network. Using M_{ij} as adjacency matrix and the Laplacian matrix eigenvalues properties, we found the threshold value for the existence of one single connected component.

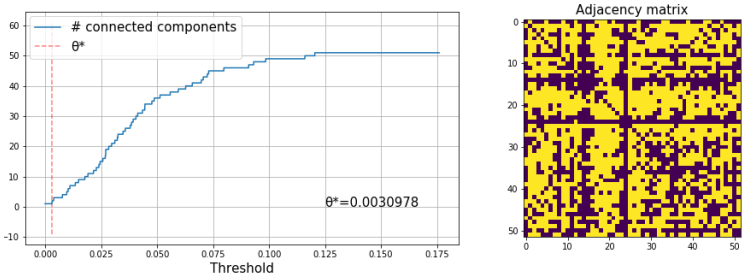

Figure 11: Left: Connected components threshold; Right: Adjacency matrix

The adjacency matrix in Figure 11 (Right) has been flattened in order to make the graph undirected and unweighted for a future comparison with Erdős–Rényi graphs.

Using the `networkx` package we calculated several graph properties and the clustering and degree distribution.

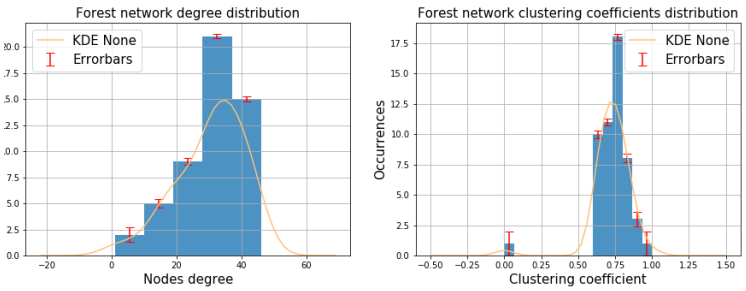

Figure 12: Forest network parameters distribution

Subsequently the same parameters were estimated for an Erdős–Rényi graph.

The graphs were created using the following probability:

$$p = \frac{\langle k \rangle}{S' - 1}$$

and the quantities were averaged over 10 random graphs with the same number of nodes

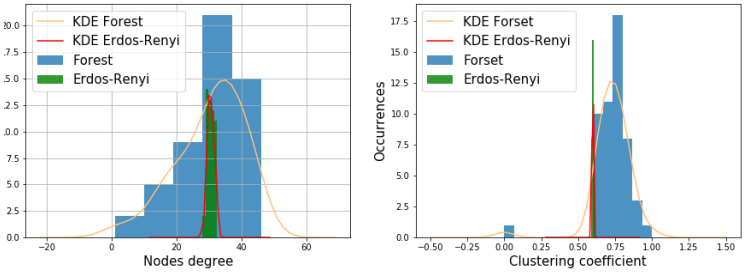

Figure 13: Forest - Erdős–Rényi network confrontation

The Erdős–Rényi graph showed much more narrower distribution but with compatible mean value in case of the degree distribution.

One can find listed in the `code` other parameters values for comparison.

A Metropolis: ΔE

$$E' - E = -\frac{k}{S}\Big(\sum_{i=1}^{S}\sigma_i^{new}\Big)^2 + \frac{k}{S}\Big(\sum_{i=1}^{S}\sigma_i^{old}\Big)^2 - \sum_{i=1}^{S}\lambda_i\sigma_i^{new} + \sum_{i=1}^{S}\lambda_i\sigma_i^{old}$$
$$= +\frac{k}{S}\Big[\Big(\sum_{i=1}^{S}\sigma_i^{old}\Big)^2 + \Big(\sum_{i=1}^{S}\sigma_i^{new}\Big)^2\Big] + \lambda_k\big(\sigma_k^{old} - \sigma_k^{new}\big)$$

We denote the quantity $\sum_{i=1}^{S}\sigma_i^{new} = \big(s_+ + s_-\big)_{new}$ where s_+, s_- are the number of the present and non present (with its sign) species, respectively.

Now we can distinguish the 2 cases:

$$\sigma_k^{old} - \sigma_k^{new} = \begin{cases} +2 & if\ +1 \to -1 \\ -2 & if\ +1 \to -1 \end{cases} \quad \Rightarrow \sigma_k^{old} - \sigma_k^{new} = -2\sigma_k^{new}$$

$$\sum_{i=1}^{S}\sigma_i^{old} = \begin{cases} \big(s_+ + s_-\big)_{new} + 2 & if\ +1 \to -1 \\ \big(s_+ + s_-\big)_{new} - 2 & if\ +1 \to -1 \end{cases} \quad \Rightarrow \sum_{i=1}^{S}\sigma_i^{old} = \big(s_+ + s_-\big)_{new} - 2\sigma_k^{new}$$

Therefore, after some calculations, we can write:

$$E' - E = -2\lambda_k\sigma_k^{new} + 4\frac{k}{S}\sigma_k^{new}[\sigma_k^{new} - (s_+ + s_-)_{new}]$$

B ADAM algorithm

The parameters have been set to $\beta_1 = 0.9$ $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$

$$\big(\mathbf{m}_i\big)_t = \beta_1\big(\mathbf{m}_i\big)_{t-1} + (1-\beta_1)\big(\nabla\sigma_i\big)_t, \qquad \big(\mathbf{s}_i\big)_t = \beta_2\big(\mathbf{s}_i\big)_{t-1} + (1-\beta_2)\big(\nabla\sigma_i\big)_t^2$$

$$\big(\mathbf{m}_0\big)_t = \beta_1\big(\mathbf{m}_0\big)_{t-1} + (1-\beta_1)\big(\nabla\sigma_0\big)_t, \qquad \big(\mathbf{s}_0\big)_t = \beta_2\big(\mathbf{s}_0\big)_{t-1} + (1-\beta_2)\big(\nabla\sigma_0\big)_t^2$$

$$\big(\hat{\mathbf{m}}_j\big)_t = \frac{\big(\mathbf{m}_j\big)_t}{1-\beta_1^{t+1}}, \qquad \big(\hat{\mathbf{s}_j}\big)_t = \frac{\big(\hat{\mathbf{s}_j}\big)_t}{1-\beta_2^{t+1}}$$

with $j = \{0, 1, ..., S\}$.

This algorithm renders the choice of the learning rate more flexible since it considers first and second order momentum to damp or accelerate the "update" each step.