Approximated Newton Algorithm for the Ising Model Inference Speeds Up Convergence, Performs Optimally and Avoids Over-fitting

Ulisse Ferrari

Institut de la Vision, Sorbonne Universités, UPMC

New Frontiers in Non-equilibrium Physics 2015
Outlook of the seminar

1. Introduction with an application of pairwise Ising Model to Neuroscience
2. Maximal Entropy model and the Vanilla (Standard) Learning Algorithm
3. Approximate Newton Method
4. The Long-Time Limit: Stochastic Dynamics
5. Properties of the Stationary Distribution
6. Conclusions and Perspectives
Model Inference:
Finding the probability distribution reproducing the data system statistics.
Model Inference:
Finding the probability distribution reproducing the data system statistics.
Useful for characterizing the behavior of systems of many, strongly correlated, units:
neurons, proteins, virus, species distribution, bird flocks but...
Model Inference:
Finding the probability distribution reproducing the data system statistics.
Useful for characterizing the behavior of systems of many, strongly correlated, units:
neurons, proteins, virus, species distribution, bird flocks
but...

which distribution?
Model Inference:
Finding the probability distribution reproducing the data system statistics.
Useful for characterizing the behavior of systems of many, strongly correlated, units: neurons, proteins, virus, species distribution, bird flocks but...

which distribution?

Maximum Entropy (MaxEnt) Inference:
Search for the largest entropy distribution satisfying a set of constraints.
Example: pairwise Ising Model

Given binary units data-set of $B$ configurations of $N$ units:

$$\left\{\{\sigma_i(b)\}^N_{i=1}\right\}^B_{b=1}$$

Find the MaxEnt model reproducing single and pairwise correlations:

$$\langle \sigma_i \rangle_{\text{MODEL}} = \langle \sigma_i \rangle_{\text{DATA}} \equiv \frac{1}{B} \sum_b \sigma_i(b)$$

$$\langle \sigma_i \sigma_j \rangle_{\text{MODEL}} = \langle \sigma_i \sigma_j \rangle_{\text{DATA}} \equiv \frac{1}{B} \sum_b \sigma_i(b)\sigma_j(b)$$
Example: pairwise Ising Model

Given binary units data-set of $B$ configurations of $N$ units:

$$\left\{ \{ \sigma_i(b) \}_{i=1}^N \right\}_{b=1}^B$$

Find the MaxEnt model reproducing single and pairwise correlations:

$$\langle \sigma_i \rangle_{\text{MODEL}} = \langle \sigma_i \rangle_{\text{DATA}} \equiv \frac{1}{B} \sum_b \sigma_i(b)$$

$$\langle \sigma_i \sigma_j \rangle_{\text{MODEL}} = \langle \sigma_i \sigma_j \rangle_{\text{DATA}} \equiv \frac{1}{B} \sum_b \sigma_i(b) \sigma_j(b)$$

Finely tune the parameters $\{h, J\}$ of the pairwise Ising model:

$$P_{h,j}(\sigma) = \exp \left\{ \sum_i h_i \sigma_i + \sum_{ij} J_{ij} \sigma_i \sigma_j \right\} / Z[h, J]$$
In vivo Pre-Frontal Cortex Recording:
In vivo Pre-Frontal Cortex Recording:
97 experimental sessions of:

Peyrache et al. Nat. Neurosci. (2009)
Ising Model Inference

\[ \sigma_i(b) = 1 \text{ if neuron } i \text{ spiked during time-bin } b \]

Ask to reproduce neurons firing rates and correlations.

Schneidman et al. Nature 2006; Cocco, Monasson, PRL (2011)
Ising Model Inference

\[ \sigma_i(b) = 1 \text{ if neuron } i \text{ spiked during time-bin } b \]

Ask to reproduce neurons firing rates and correlations.

Schneidman et al. Nature 2006; Cocco, Monasson, PRL (2011)
**Ising Model Inference**

\[ \sigma_i(b) = 1 \text{ if neuron } i \text{ spiked during time-bin } b \]

Ask to reproduce neurons firing rates and correlations.

97 \times 3 \text{ couplings network sets } (97 \times \{\text{PRE, TASK, POST}\})

*Schneidman et al. Nature 2006; Cocco, Monasson, PRL (2011)*
Learning related coupling Adjustment

\[ A = \sum_{i,j: J^{\text{TASK}}, J^{\text{POST}} \neq 0} \text{sign}(J_{ij}^{\text{TASK}} - J_{ij}^{\text{PRE}}) \cdot (J_{ij}^{\text{POST}} - J_{ij}^{\text{PRE}}) \]
Learning related coupling Adjustment

\[ A = \sum_{i,j: J_{\text{TASK}}, J_{\text{POST}} \neq 0} \text{sign}(J_{ij}^{\text{TASK}} - J_{ij}^{\text{PRE}}) \cdot (J_{ij}^{\text{POST}} - J_{ij}^{\text{PRE}}) \]
Learning related coupling Adjustment

\[ A = \sum_{i,j: J_{\text{TASK}}^{\text{POST}} \neq 0} \text{sign}(J_{ij}^{\text{TASK}} - J_{ij}^{\text{PRE}}) \cdot (J_{ij}^{\text{POST}} - J_{ij}^{\text{PRE}}) \]
1 Maximal Entropy Models and the Vanilla (standard) Learning Algorithm

2 Approximated Newton Method

3 The Long-Time Limit: Stochastic Dynamics

4 Properties of the Stationary Distribution
General MaxEnt

Given a list of $D$ observables to reproduce $\{\Sigma_a(\sigma)\}_{a=1}^D$
(generic functions of the system units)

Find the MaxEnt model parameters $\{X_a\}_{a=1}^D$

$$P_X(\sigma) = \exp \left\{ \sum_a X_a \Sigma_a(\sigma) \right\} / Z[X]$$

reproducing the observables averages:

$$\langle \Sigma_a \rangle_{\text{DATA}} \equiv P_a = Q_a[X] \equiv \langle \Sigma_a \rangle_X$$
Equivalent to log-likelihood maximization:

\[ X^* = \arg \max_X \left[ \log L[X] \right] \equiv \arg \max_X \left[ X \cdot P - \log Z[X] \right] \]
Approximated Newton Algorithm for the Ising Model Inference Speeds Up Convergence, Performs Optimally and Avoids Over-fitting

Maximal Entropy Models and the Vanilla (standard) Learning Algorithm

Equivalent to log-likelihood maximization:

\[ X^* = \arg \max_X \log L[X] \equiv \arg \max_X [X \cdot P - \log Z[X]] \]

in fact:

\[ \nabla_a \log L[X] = \frac{d}{dX_a} [X \cdot P - \log Z[X]] = P_a - Q_a[X] \]
Equivalent to log-likelihood maximization:

\[ X^* = \arg \max_X \left[ \log L[X] \right] \equiv \arg \max_X \left[ X \cdot P - \log Z[X] \right] \]

in fact:

\[ \nabla_a \log L[X] = \frac{d}{dX_a} \left[ X \cdot P - \log Z[X] \right] = P_a - Q_a[X] \]

Cannot be solved analytically. Ackley, Hinton and Sejnowski (Vanilla Gradient):

\[ X_{t+1} = X_t + \delta X_t^{VG}; \quad \delta X_t^{VG} = \alpha(P - Q[X_t]) \]
Equivalent to log-likelihood maximization:

\[ X^* = \arg \max_X \left[ \log L[X] \right] \equiv \arg \max_X \left[ X \cdot P - \log Z[X] \right] \]

in fact:

\[ \nabla a \log L[X] = \frac{d}{dX_a} \left[ X \cdot P - \log Z[X] \right] = P_a - Q_a[X] \]

Cannot be solved analytically. Ackley, Hinton and Sejnowski (Vanilla Gradient):

\[ X_{t+1} = X_t + \delta X_{t}^{VG}; \quad \delta X_{t}^{VG} = \alpha (P - Q[X_t]) \]

If \( 0 < P_a < 1 \) for all \( a = 1, \ldots D \), the problem is well posed:

\[ X^* \text{ exists and is unique and the dynamics converges} \]

(for infinitesimally small \( \alpha \))
A 2-dimensional example:

$$\log L[u, v] = -\frac{a}{2}(u - u_\infty)^2 - \frac{b}{2}(v - v_\infty)^2$$
A 2-dimensional example:

$$\log L[u, v] = -\frac{a}{2}(u - u_\infty)^2 - \frac{b}{2}(v - v_\infty)^2$$
A 2-dimensional example:

$$\log L[u, v] = -\frac{a}{2}(u - u_\infty)^2 - \frac{b}{2}(v - v_\infty)^2$$
A 2-dimensional example:

$$\log L[u, v] = -\frac{a}{2}(u - u_\infty)^2 - \frac{b}{2}(v - v_\infty)^2$$
A 2-dimensional example:
\[ \log L[u, v] = -\frac{a}{2}(u - u_\infty)^2 - \frac{b}{2}(v - v_\infty)^2 \]

**Vanilla Gradient:**
\[ \delta u_{t}^{\text{VG}} \sim (1 - \alpha a)^{-t} \Rightarrow \alpha < 2/a; \quad \delta v_{t}^{\text{VG}} \sim (1 - \alpha b)^{-t} \Rightarrow \alpha < 2/b \]
A 2-dimensional example:

$$\log L[u, v] = -\frac{a}{2}(u - u_\infty)^2 - \frac{b}{2}(v - v_\infty)^2$$

**Vanilla Gradient:**

$$\delta u_{t}^{\text{VG}} \sim (1 - \alpha a)^{-t} \Rightarrow \alpha < 2/a; \quad \delta v_{t}^{\text{VG}} \sim (1 - \alpha b)^{-t} \Rightarrow \alpha < 2/b$$

**Newton Method:**

$$\delta u_{t}^{\text{VG}} \sim (1 - \alpha)^{-t} \Rightarrow \alpha < 2; \quad \delta v_{t}^{\text{VG}} \sim (1 - \alpha)^{-t} \Rightarrow \alpha < 2$$
A 2-dimensional example:

\[ \log L[u, v] = -\frac{a}{2}(u - u_\infty)^2 - \frac{b}{2}(v - v_\infty)^2 \]

**Vanilla Gradient:**

\[ \delta u_t^{VG} \sim (1 - \alpha a)^{-t} \Rightarrow \alpha < 2/a; \quad \delta v_t^{VG} \sim (1 - \alpha b)^{-t} \Rightarrow \alpha < 2/b \]

**Newton Method:**

\[ \delta u_t^{VG} \sim (1 - \alpha)^{-t} \Rightarrow \alpha < 2; \quad \delta v_t^{VG} \sim (1 - \alpha)^{-t} \Rightarrow \alpha < 2 \]

\[ \alpha = 1 \quad \Rightarrow \quad \text{convergence in one step!} \]
1. Maximal Entropy Models and the Vanilla (standard) Learning Algorithm

2. Approximated Newton Method

3. The Long-Time Limit: Stochastic Dynamics

4. Properties of the Stationary Distribution
The same happens for the MaxEnt inference:

$$\log L[X \approx X^*] \approx \log L[X^*] - \frac{1}{2} \sum_{ab}(X_a - X_a^*) \chi[X^*]_{ab} (X_b - X_b^*)$$

$$\chi_{ab}[X] \equiv -\frac{\partial^2 \log L[X]}{\partial X_a \partial X_b} = \langle \Sigma_a \Sigma_b \rangle_X - \langle \Sigma_a \rangle_X \langle \Sigma_b \rangle_X$$
The same happens for the MaxEnt inference:

\[
\log L[X \approx X^*] \approx \log L[X^*] - \frac{1}{2} \sum_{ab} (X_a - X_a^*) \chi[X^*]_{ab} (X_b - X_b^*)
\]

\[
\chi_{ab}[X] \equiv -\frac{\partial^2 \log L[X]}{\partial X_a \partial X_b} = \langle \Sigma_a \Sigma_b \rangle_X - \langle \Sigma_a \rangle_X \langle \Sigma_b \rangle_X
\]

**Vanilla Gradient:** \( \delta X^\text{VG}_t = \alpha \nabla \log L[X_{t-1}] \)

\[
\delta X^\mu_t \equiv \sum_a V^\mu_a \delta X_{a,t} \sim (1 - \alpha \lambda^\mu)^{-t}
\]
The same happens for the MaxEnt inference:

\[
\log L[X \approx X^*] \approx \log L[X^*] - \frac{1}{2} \sum_{ab} (X_a - X^*_a) \chi[X^*]_{ab} (X_b - X^*_b)
\]

\[
\chi_{ab}[X] \equiv -\frac{\partial^2 \log L[X]}{\partial X_a \partial X_b} = \langle \Sigma_a \Sigma_b \rangle_X - \langle \Sigma_a \rangle_X \langle \Sigma_b \rangle_X
\]

**Vanilla Gradient:** \(\delta X^\text{VG}_t = \alpha \nabla \log L[X_{t-1}]\)

\[
\delta X^\mu_t \equiv \sum_a V^\mu_a \delta X_{a,t} \sim (1 - \alpha \lambda_\mu)^{-t}
\]

**Newton Method**\(^1\): \(\delta X^\text{NM}_t = \alpha \chi^{-1}[X_{t-1}] \nabla \log L[X_{t-1}]\)

\[
\delta X^\mu_t \equiv \sum_a V^\mu_a \delta X_{a,t} \sim (1 - \alpha)^{-t}
\]

\(^1\) (here equivalent to Amari98 Natural Gradient)
The same happens for the MaxEnt inference:

\[
\log L[X \approx X^*] \approx \log L[X^*] - \frac{1}{2} \sum_{ab} (X_a - X_a^*) \chi[X^*]_{ab} (X_b - X_b^*)
\]

\[
\chi_{ab}[X] \equiv -\frac{\partial^2 \log L[X]}{\partial X_a \partial X_b} = \langle \Sigma_a \Sigma_b \rangle_X - \langle \Sigma_a \rangle_X \langle \Sigma_b \rangle_X
\]

**Vanilla Gradient:** \( \delta X_t^{VG} = \alpha \nabla \log L[X_{t-1}] \)

\[
\delta X_t^{\mu} \equiv \sum_a V_a^{\mu} \delta X_{a,t} \sim (1 - \alpha \lambda_{\mu})^{-t}
\]

**Newton Method\(^1\):** \( \delta X_t^{NM} = \alpha \chi^{-1}[X_{t-1}] \nabla \log L[X_{t-1}] \)

\[
\delta X_t^{\mu} \equiv \sum_a V_a^{\mu} \delta X_{a,t} \sim (1 - \alpha)^{-t}
\]

**VERY SLOW:** expensive estimation & inversion of \( \chi[X] \)

\(^1\) (here equivalent to Amari98 Natural Gradient)
However, for the Ising model we can approximate:

\[ \chi_{ab}[X^*] \approx \chi_{ab} \equiv \langle \Sigma_a \Sigma_b \rangle_{\text{DATA}} - \langle \Sigma_a \rangle_{\text{DATA}} \langle \Sigma_b \rangle_{\text{DATA}} \]
However, for the Ising model we can approximate:

\[ \chi_{ab}[X^*] \approx \chi_{ab} \equiv \langle \Sigma_a \Sigma_b \rangle_{\text{DATA}} - \langle \Sigma_a \rangle_{\text{DATA}} \langle \Sigma_b \rangle_{\text{DATA}} \]

**Approximated Newton (AN) Method:**

\[ \delta X_t^{AN} = \alpha \chi^{-1} \nabla \log \mathcal{L}[X_{t-1}] \]
However, for the Ising model we can approximate:

\[ \chi_{ab}[\mathbf{X}^*] \approx \overline{\chi}_{ab} \equiv \langle \Sigma_a \Sigma_b \rangle_{\text{DATA}} - \langle \Sigma_a \rangle_{\text{DATA}} \langle \Sigma_b \rangle_{\text{DATA}} \]

Approximated Newton (AN) Method:

\[ \delta X_t^{\text{AN}} = \alpha \frac{1}{\chi} \nabla \log L[\mathbf{X}_{t-1}] \]

Remarks on \( \chi[\mathbf{X}^*] \approx \overline{\chi} \):

- equivalent to say that an Ising distribution properly describes data.
- states that the model Fisher is close to the observables co-variance.
As the algorithm works iteratively, it requires an early-stop condition.
As the algorithm works iteratively, it requires an early-stop condition.

Idea: stop the algorithm when $Q[X]$ is statistically compatible with $P$ using the $P$-covariance $\chi/B$. 
As the algorithm works iteratively, it requires an **early-stop condition**

idea: stop the algorithm when 

**Q[X]** is statistically compatible with **P**

using the **P**-covariance \( \frac{\chi}{B} \)

\[
e( \mathbf{P}, \mathbf{Q}[X] ) \equiv \frac{\mathcal{B}}{2D} \sum_{ab} (P_a - Q_a) \left( \frac{\chi^{-1}}{B} \right)_{ab} (P_b - Q_b)
\]

quantifies the distance between **Q[X]** and **P** in the \( \frac{\chi}{B} \) metric.
As the algorithm works iteratively, it requires an **early-stop condition**

idea: stop the algorithm when

\[ Q[X] \text{ is statistically compatible with } P \]

using the \( P \)-covariance \( \chi / B \)

\[
\epsilon\left( P , Q[X] \right) \equiv \frac{B}{2D} \sum_{ab} (P_a - Q_a) \left( \chi^{-1} \right)_{ab} (P_b - Q_b)
\]

quantifies the distance between \( Q[X] \) and \( P \) in the \( \chi / B \) metric.

For two \( i.i.d \) data-sets: \( \epsilon\left( P , P' \right) \approx 1 \)

\[ \Rightarrow \text{ we stop the algorithm as soon as } \epsilon < 1 \]
APPROXIMATED NEWTON ALGORITHM:

1 Initialization:
   (a) Choose $X_0$ and compute $Q[X_0]$ and $\epsilon_0 = \epsilon(P, Q[X_0])$
   (b) Then set $\alpha_0 = 1$ and $M = \min\left(\frac{2B}{\epsilon_0}, B\right)$ MCMC samplings
APPROXIMATED NEWTON ALGORITHM:

1 Initialization:
   (a) Chose $X_0$ and compute $Q[X_0]$ and $\epsilon_0 = \epsilon(P, Q[X_0])$
   (b) Then set $\alpha_0 = 1$ and $M = \min(\frac{2B}{\epsilon_0}, B)$ MCMC samplings

2 Iterate the following step:
   (a) update the $X_t$
   (b) estimate $Q[X_t]$ with $M = \min(\frac{2B}{\epsilon_{t-1}}, B)$ MCMC samplings
   (c) compute $\epsilon_t = \epsilon(P, Q[X_t])$
       (d1) $\epsilon_t < \epsilon_{t-1}$: accept the update and increase $\alpha$
       (d2) $\epsilon_t > \epsilon_{t-1}$: discard the update, lower $\alpha$ and re-estimate $Q[X_t]$. 
APPORXIMATED NEWTON ALGORITHM:

1 Initialization:
   (a) Chose $X_0$ and compute $Q[X_0]$ and $\epsilon_0 = \epsilon\left( P, Q[X_0] \right)$
   (b) Then set $\alpha_0 = 1$ and $M = \min\left(\frac{2B}{\epsilon_0}, B\right)$ MCMC samplings

2 Iterate the following step:
   (a) update the $X_t$
   (b) estimate $Q[X_t]$ with $M = \min\left(\frac{2B}{\epsilon_{t-1}}, B\right)$ MCMC samplings
   (c) compute $\epsilon_t = \epsilon\left( P, Q[X_t] \right)$,
      (d1) $\epsilon_t < \epsilon_{t-1}$: accept the update and increase $\alpha$
      (d2) $\epsilon_t > \epsilon_{t-1}$: discard the update, lower $\alpha$ and re-estimate $Q[X_t]$. 

3 stop the algorithm when $\epsilon_t < 1$. 
Rat retina ganglion cells

Two moving bars.

2.1h of MEA recording

\[ B = 4.8 \cdot 10^5 \text{ of } \Delta t = 16\text{ms} \]

\[ N = 95 \text{ cells} \]

\[ D = 4560 \text{ parameters to infer.} \]
Rat retina ganglion cells

Two moving bars.

2.1h of MEA recording

\[ B = 4.8 \cdot 10^5 \text{ of } \Delta t = 16ms \]

\[ N = 95 \text{ cells} \]

\[ D = 4560 \text{ parameters to infer.} \]

Convergence time from independent spins model with 8 × 3.4Ghz CPUs:

\[ T^{AN} = 144 \pm 4s \]

\[ T^{VG}(\alpha = 0.15) = 4.2 \cdot 10^4s \]
Approximated Newton Algorithm for the Ising Model Inference Speeds Up Convergence, Performs Optimally and Avoids Over-fitting

Approximated Newton Method

Rat retina ganglion cells

Two moving bars.

2.1h of MEA recording

\[ B = 4.8 \cdot 10^5 \text{ of } \Delta t = 16ms \]

\[ N = 95 \text{ cells} \]

\[ D = 4560 \text{ parameters to infer.} \]

Convergence time from independent spins model with 8 \times 3.4Ghz CPUs:

\[ T^{AN} = 144 \pm 4s \]

\[ T^{VG}(\alpha = 0.15) = 4.2 \cdot 10^4s \]

\[ c_{ij} = \langle \sigma_i \sigma_j \rangle - \langle \sigma_i \rangle \langle \sigma_j \rangle \]
Approximated Newton Algorithm for the Ising Model Inference Speeds Up Convergence, Performs Optimally and Avoids Over-fitting

Rat retina ganglion cells

Two moving bars.

2.1h of MEA recording

\[ B = 4.8 \cdot 10^5 \; \text{of} \; \Delta t = 16\text{ms} \]

\[ N = 95 \; \text{cells} \]

\[ D = 4560 \; \text{parameters to infer.} \]

Convergence time from independent spins model with \(8 \times 3.4\text{Ghz CPUs}:\)

\[ T^{\text{AN}} = 144 \pm 4\text{s} \]

\[ T^{\text{VG}}(\alpha = 0.15) = 4.2 \cdot 10^4\text{s} \]

\[ P(K) = \text{Prob} \left( \sum \sigma_i = K \right) \]
1. Maximal Entropy Models and the Vanilla (standard) Learning Algorithm

2. Approximated Newton Method

3. The Long-Time Limit: Stochastic Dynamics

4. Properties of the Stationary Distribution
\( Q[X] \) is estimated through \( M \) MCMC measurements.

\( Q[X] \Rightarrow Q[X]^{MC} \) is random variable!
$Q[X]$ is estimated through $M$ MCMC measurements.

$Q[X] \Rightarrow Q[X]^{MC}$ is random variable!

$\nabla \log L_{X}^{MC} = P - Q[X]^{MC} \to 0$ only on average,
Approximated Newton Algorithm for the Ising Model Inference Speeds Up Convergence, Performs Optimally and Avoids Over-fitting

The Long-Time Limit: Stochastic Dynamics

\( Q[X] \) is estimated through \( M \) MCMC measurements.

\[ Q[X] \Rightarrow Q[X]^{MC} \text{ is random variable!} \]

\( \nabla \log L_X^{MC} = P - Q[X]^{MC} \to 0 \text{ only on average,} \)

Change of Framework:

\[ X_t \to P_t(X) \]

\( X \), rather than converge to a fixed point, approaches a stationary \( P_\infty(X) \)
\( \mathbf{Q}[\mathbf{X}] \) is estimated through \( M \) MCMC measurements.

\[
\mathbf{Q}[\mathbf{X}] \Rightarrow \mathbf{Q}[\mathbf{X}]^\text{MC} \text{ is random variable!}
\]

\[
\nabla \log L^\text{MC}_x = \mathbf{P} - \mathbf{Q}[\mathbf{X}]^\text{MC} \to 0 \text{ only on average},
\]

\textbf{Change of Framework:}

\[\mathbf{X}_t \to \mathbf{P}_t(\mathbf{X})\]

\( \mathbf{X} \), rather than converge to a fixed point, approaches a stationary \( \mathbf{P}_\infty(\mathbf{X}) \)

\textbf{Master Equation:}

\[
\mathbf{P}_{t+1}(\mathbf{X}') = \int D\mathbf{X} \mathbf{P}_t(\mathbf{X}) \mathcal{W}_{\mathbf{X} \to \mathbf{X}'}(\alpha)
\]
For $M \gg 1$ and $X \approx X^*$:

$$\log L[X] \approx \log L[X^*] - \frac{1}{2} \sum_{ab} (X_a - X^*_a) \chi[X^*]_{ab} (X_b - X^*_b)$$
For $M \gg 1$ and $X \approx X^*$:

$$\log L[X] \simeq \log L[X^*] - \frac{1}{2} \sum_{ab} (X_a - X_a^*) \chi[X^*]_{ab} (X_b - X_b^*)$$

$$\langle \nabla_a \log L_{X}^{MC} \rangle = \sum_b \chi[X^*]_{ab} (X_b^* - X_b) \approx \sum_b \chi_{ab} (X_b^* - X_b)$$
For $M \gg 1$ and $X \approx X^*$:

$$\log L[X] \approx \log L[X^*] - \frac{1}{2} \sum_{ab} (X_a - X_a^*) \chi[X^*]_{ab} (X_b - X_b^*)$$

$\langle \nabla_a \log L^\text{MC}_X \rangle = \sum_b \chi[X^*]_{ab} (X_b^* - X_b) \approx \sum_b \bar{\chi}_{ab} (X_b^* - X_b)$

$\left\langle \nabla_a \log L^\text{MC}_X \nabla_b \log L^\text{MC}_X \right\rangle_c = \chi[X]_{ab}/M \approx \chi[X^*]_{ab}/M \approx \bar{\chi}_{ab}/M$
For $M \gg 1$ and $X \approx X^*$:

$$\log L[X] \simeq \log L[X^*] - \frac{1}{2} \sum_{ab} (X_a - X^*_a) \chi[X^*]_{ab} (X_b - X^*_b)$$

$$\langle \nabla_a \log L^\text{MC}_X \rangle = \sum_b \chi[X^*]_{ab} (X^*_b - X_b) \approx \sum_b \overline{\chi}_{ab} (X^*_b - X_b)$$

$$\left\langle \nabla_a \log L^\text{MC}_X \nabla_b \log L^\text{MC}_X \right\rangle_c = \chi[X]_{ab} / M \simeq \chi[X^*]_{ab} / M \approx \overline{\chi}_{ab} / M$$

A normal approximation gives:

$$P(\nabla \log L^\text{MC}_X) \simeq \mathcal{N} \left[ \overline{\chi} \cdot (X^* - X); \overline{\chi} / M \right] (\nabla \log L^\text{MC}_X)$$
For $M \gg 1$ and $X \approx X^*$:

\[
\log L[X] \simeq \log L[X^*] - \frac{1}{2} \sum_{ab} (X_a - X_a^*) \chi[X^*]_{ab} (X_b - X_b^*)
\]

1. $\langle \nabla_a \log L^\text{MC}_X \rangle = \sum_b \chi[X^*]_{ab} (X_b^* - X_b) \approx \sum_b \chi_{ab}(X_b^* - X_b)$

2. $\left\langle \nabla_a \log L^\text{MC}_X \nabla_b \log L^\text{MC}_X \right\rangle_c = \chi[X]_{ab}/M \approx \chi[X^*]_{ab}/M \approx \chi_{ab}/M$

A normal approximation gives:

\[
P(\nabla \log L^\text{MC}_X) \simeq N\left[ \chi \cdot (X^* - X) ; \chi/M \right] (\nabla \log L^\text{MC}_X)
\]

- $W^{\text{VG}}_{X \to X'}(\alpha) = \text{Prob}\left(\nabla \log L^\text{MC}_X = \frac{X' - X}{\alpha}\right)$

- $W^{\text{AN}}_{X \to X'}(\alpha) = \text{Prob}\left(\nabla \log L^\text{MC}_X = \frac{\chi}{\chi} \cdot \frac{X' - X}{\alpha}\right)$
Imposing $P_{t+1}(\mathbf{X}) = P_t(\mathbf{X})$

- $P_{\infty}^{VG}(\mathbf{X}) = \mathcal{N}[\mathbf{X}^*; \frac{\alpha}{M}(2\delta - \alpha \overline{\chi})^{-1}](\mathbf{X})$
- $P_{\infty}^{AN}(\mathbf{X}) = \mathcal{N}[\mathbf{X}^*; \frac{\alpha}{M(2-\alpha)} \overline{\chi}^{-1}](\mathbf{X})$
Imposing $P_{t+1}(X) = P_t(X)$

- $P^\text{VG}_\infty(X) = \mathcal{N}\left[X^*; \frac{\alpha}{M}(2\delta - \alpha \bar{\chi})^{-1}\right](X), \quad \alpha \lambda_\mu < 2$

- $P^\text{AN}_\infty(X) = \mathcal{N}\left[X^*; \frac{\alpha}{M(2-\alpha)} \bar{\chi}^{-1}\right](X), \quad \alpha < 2$
Imposing $P_{t+1}(X) = P_t(X)$

- $P_{\infty}^{VG}(X) = \mathcal{N}\left[X^*; \frac{\alpha}{M}(2\delta - \alpha \chi)^{-1}\right](X)$, $\alpha \lambda_\mu < 2$
- $P_{\infty}^{AN}(X) = \mathcal{N}\left[X^*; \frac{\alpha}{M(2-\alpha)} \chi^{-1}\right](X)$, $\alpha < 2$

Which self-consistently defines $X \approx X^*$
Approximated Newton Algorithm for the Ising Model Inference Speeds Up Convergence, Performs Optimally and Avoids Over-fitting

The Long-Time Limit: Stochastic Dynamics

Imposing $P_{t+1}(X) = P_t(X)$

- $P_{\infty}^{VG}(X) = \mathcal{N}\left[X^*; \frac{\alpha}{M}(2\delta - \alpha \overline{\chi})^{-1}\right](X), \quad \alpha \lambda_\mu < 2$
- $P_{\infty}^{AN}(X) = \mathcal{N}\left[X^*; \frac{\alpha}{M(2-\alpha)} \overline{\chi}^{-1}\right](X), \quad \alpha < 2$

Which self-consistently defines $X \approx X^*$

From $P(\nabla \log L_{X}^{MC}) = P(P - Q[X]^{MC})$

- $P_{\infty}^{VG}(Q^{MC}) = \mathcal{N}\left[P; \frac{2}{M} \overline{\chi} (2\delta - \alpha \overline{\chi})^{-1}\right](Q^{MC})$
- $P_{\infty}^{AN}(Q^{MC}) = \mathcal{N}\left[P; \frac{2}{M(2-\alpha)} \overline{\chi}\right](Q^{MC})$
Imposing $P_{t+1}(X) = P_t(X)$

- $P^\text{VG}_\infty(X) = \mathcal{N}[X^*; \frac{\alpha}{M} \left(2\delta - \alpha \overline{\chi}\right)^{-1}](X), \quad \alpha \lambda_\mu < 2$
- $P^\text{AN}_\infty(X) = \mathcal{N}[X^*; \frac{\alpha}{M(2-\alpha)} \overline{\chi}^{-1}](X), \quad \alpha < 2$

Which self-consistently defines $X \approx X^*$

From $P(\nabla \log L^\text{MC}_X) = P(P - Q[X]^\text{MC})$

- $P^\text{VG}_\infty(Q^\text{MC}) = \mathcal{N}[P; \frac{2}{M} \overline{\chi} \left(2\delta - \alpha \overline{\chi}\right)^{-1}](Q^\text{MC})$
- $P^\text{AN}_\infty(Q^\text{MC}) = \mathcal{N}[P; \frac{2}{M(2-\alpha)} \overline{\chi}](Q^\text{MC})$

Which is better? How to set the parameters?
1 Maximal Entropy Models and the Vanilla (standard) Learning Algorithm

2 Approximated Newton Method

3 The Long-Time Limit: Stochastic Dynamics

4 Properties of the Stationary Distribution
Algorithm Vs Empirical distributions

An experiment provides empirical estimates of $Q^{\text{EMP}}$:

$$P^{\text{EMP}}(Q^{\text{EMP}}) \approx \mathcal{N}[P^{\text{TRUE}}, \chi^{\text{EMP}}]$$

- $P^{\text{TRUE}}$: result from infinitely long experiment
- $\chi^{\text{EMP}}$: expected co-variance for $B$ measurements
Algorithm Vs Empirical distributions

An experiment provides empirical estimates of $\mathbf{Q}^{\text{EMP}}$:

$$P^{\text{EMP}}(\mathbf{Q}^{\text{EMP}}) \approx \mathcal{N}[P^{\text{TRUE}}, \chi^{\text{EMP}}]$$

- $P^{\text{TRUE}}$: result from infinitely long experiment
- $\chi^{\text{EMP}}$: expected co-variance for $B$ measurements
- $P$: one-shot sampling of $P^{\text{EMP}}$

An inference algorithm provides numerical estimates of $\mathbf{Q}^{\text{MC}}$:

$$P^{\text{ALG}}_P(\mathbf{Q}^{\text{MC}}) \approx \mathcal{N}[P, \chi^{\text{ALG}}]$$
Algorithm Vs Empirical distributions

An experiment provides empirical estimates of $Q^{\text{EMP}}$:

$$P^{\text{EMP}}(Q^{\text{EMP}}) \approx \mathcal{N}[P^{\text{TRUE}}, \chi^{\text{EMP}}]$$

An inference algorithm provides numerical estimates of $Q^{\text{MC}}$:

$$P^{\text{ALG}}_P(Q^{\text{MC}}) \approx \mathcal{N}[P, \chi^{\text{ALG}}]$$

- $P^{\text{TRUE}}$: result from infinitely long experiment
- $\chi^{\text{EMP}}$: expected co-variance for $B$ measurements
- $P$: one-shot sampling of $P^{\text{EMP}}$

An optimal inference algorithm should provide:

$P^{\text{ALG}}$ as close as possible to $P^{\text{EMP}}$.

What is the optimal $\chi^{\text{ALG}}$ value?
Kullback-Leibler distance between $P^{EMP}$ and $P^{ALG}_P$:

$$D_{KL} \left( P^{EMP}(\cdot) \| P^{ALG}_P(\cdot) \right)$$
Kullback-Leibler distance between $P_{\text{EMP}}$ and $P_{\text{ALG}}$:

$$D_{KL}(P_{\text{EMP}}(\cdot) \parallel P_{\text{ALG}}(\cdot))$$

$$\chi^{\text{OPT}} = \arg \min_{\chi^{\text{ALG}}} \int \mathbf{D} \mathbf{P} \; P_{\text{EMP}}(\mathbf{P}) \; D_{KL}(P_{\text{EMP}}(\cdot) \parallel P_{\text{ALG}}(\cdot))$$
Kullback-Leibler distance between $P^{EMP}$ and $P^{ALG}_P$:

$$D_{KL} \left( P^{EMP}(\cdot) \| P^{ALG}_P(\cdot) \right)$$

$$\chi^{OPT} = \arg \min_{\chi^{ALG}} \int \mathbf{D} P^{EMP}(\mathbf{P}) \ D_{KL} \left( P^{EMP}(\cdot) \| P^{ALG}_P(\cdot) \right)$$

The solution and its approximation are:

$$\chi^{OPT} = 2\chi^{EMP} \approx 2\chi / B$$
Kullback-Leibler distance between $P_{\text{EMP}}$ and $P_{\text{ALG}}$: 

$$D_{KL}(P_{\text{EMP}}(\cdot)||P_{\text{ALG}}(\cdot))$$

$$\chi^{\text{OPT}} = \arg \min_{\chi_{\text{ALG}}} \int D P P_{\text{EMP}}(P) D_{KL}(P_{\text{EMP}}(\cdot)||P_{\text{ALG}}(\cdot))$$

The solution and its approximation are:

$$\chi^{\text{OPT}} = 2\chi_{\text{EMP}} \approx 2\frac{\chi}{B}$$

to compare with:

$$\chi^{\text{VG}} = \frac{2}{M} \overline{\chi} (2\delta - \alpha \overline{\chi})^{-1}, \quad \chi^{\text{AN}} = \frac{2}{M(2-\alpha)} \overline{\chi}$$
Kullback-Leibler distance between $P_{\text{EMP}}$ and $P_{\text{ALG}}$:

$$D_{KL} \left( P_{\text{EMP}}(\cdot) \parallel P_{\text{ALG}}(\cdot) \right)$$

$$\chi^{\text{OPT}} = \arg \min \chi_{\text{ALG}} \int D P \ P_{\text{EMP}}(P) \ D_{KL} \left( P_{\text{EMP}}(\cdot) \parallel P_{\text{ALG}}(\cdot) \right)$$

The solution and its approximation are:

$$\chi^{\text{OPT}} = 2 \chi_{\text{EMP}} \approx 2 \frac{\chi}{B}$$

to compare with:

$$\chi^{\text{VG}} = \frac{2}{M} \chi \left( 2\delta - \alpha \frac{\chi}{M} \right)^{-1}, \quad \chi^{\text{AN}} = \frac{2}{M(2-\alpha)} \chi$$

AN with $M(2-\alpha) = B$ reaches the optimum!

VG underfits $\lambda_{\mu} \gg (2 - B/M)/\alpha$ and overfits $\lambda_{\mu} \ll (2 - B/M)/\alpha$
Synthetic data: Theory Vs Simulations

Bethe Lattice Ising Model

\[ N = 10, \ c = 4 \]
\[ J_{ij} = \pm 0.53, \]
\[ h_i = -0.14 - 2 \sum_j J_{ij} \]

100 independent estimations
of \( P \) and \( \overline{\chi} \)
through \( 2^{16} \) sampling of \( P^{EMP} \)

Inference with \( M = B \)
Synthetic data: Theory Vs Simulations

Bethe Lattice Ising Model

\[ N = 10, \ c = 4 \]
\[ J_{ij} = \pm 0.53, \]
\[ h_i = -0.14 - 2 \sum_j J_{ij} \]

100 independent estimations of \( P \) and \( \overline{\chi} \) through \( 2^{16} \) sampling of \( P^{EMP} \)

Inference with \( M = B \)
Conclusions:

- MaxEnt models are useful to describe multi-units systems
- The AN learning is faster than the VG algorithm.
- Within the large $B$ approximation is possible to completely characterize the long time behavior
- The AN with $\alpha = 1$ and $M = B$ is optimal against overfitting.

Perspectives:

- Improve the gaussian approximations
- Test the algorithm to non-pairwise models
- Generalize the class of model distributions beyond MaxEnt
- Include hidden variables and the RBM framework
THANKS

Collaborators for P.F. Cortex work:
- Francesco Battaglia
- Simona Cocco
- Remi Monasson
- Gaia Tavoni

Founding
- EU-FP7 FET OPEN project Enlightenment 284801
- Human Brain Project (HBP CLAP)

arXiv:1507.04254