Numerical study of the branching tree of states in spin glasses

G. Parisi, F. Ricci-Tersenghi, **D. Yllanes**

Dipartimento di Fisica
La Sapienza Università di Roma

CompPhys13, Leipzig University, November 2013

The Sherrington-Kirkpatrick model

- Spin glasses: Random, mixed-interacting magnetic systems that experience a random, yet cooperative, freezing of spins below some critical temperature T_c.

The Sherrington-Kirkpatrick model

- Spin glasses: Random, mixed-interacting magnetic systems that experience a random, yet cooperative, freezing of spins below some critical temperature T_c.

Edwards-Anderson model

$$\mathcal{H} = -\sum_{\langle i,j\rangle} J_{ij} s_i s_j, \quad s_i = \pm 1$$

- $J_{ij} = \pm 1$ with 50% probability.
- Disorder and frustration
- Order parameter from the overlap

$$q = \lim_{t\to\infty} \frac{1}{N} \sum_x \langle s_x(0) s_x(t)\rangle_t \quad \rightsquigarrow \quad q = \frac{1}{N}\sum_x \langle s_x\rangle^2 .$$

The Sherrington-Kirkpatrick model

- The EA model is too difficult to handle analytically.
- We consider its mean-field version:

$$\mathcal{H}_J = -\sum_{i,j} J_{ij} s_i s_j, \quad s_i = \pm 1 \qquad J_{ij} = \pm\frac{1}{\sqrt{N}}.$$

The Sherrington-Kirkpatrick model

The Sherrington-Kirkpatrick model

- The EA model is too difficult to handle analytically.
- We consider its mean-field version:

$$\mathcal{H}_J = -\sum_{i,j} J_{ij} s_i s_j, \quad s_i = \pm 1 \qquad J_{ij} = \pm\frac{1}{\sqrt{N}}.$$

- Solution by G. Parisi (Replica Symmetry Breaking, RSB).
- The overlap can take any value in $[0, q_\mathrm{M}]$ with non-zero probability density $p(q)$.
- $p(q)$ is smooth in $[0, q_\mathrm{M})$, but has a δ function at q_M

The Sherrington-Kirkpatrick model

- The EA model is too difficult to handle analytically.
- We consider its mean-field version:

$$\mathcal{H}_J = -\sum_{i,j} J_{ij} s_i s_j, \quad s_i = \pm 1 \qquad J_{ij} = \pm\frac{1}{\sqrt{N}}.$$

- Solution by G. Parisi (Replica Symmetry Breaking, RSB).
- The overlap can take any value in $[0, q_\mathrm{M}]$ with non-zero probability density $p(q)$.
- $p(q)$ is smooth in $[0, q_\mathrm{M})$, but has a δ function at q_M
- The order parameter is not a number, but a function $q(x)$.
- $x(q)$ is the cumulative probability of q: $x(q) = \int_0^q \mathrm{d}q' p(q')$.

Infinitely many states

- From the previous discussion, we know that there are infinitely many relevant pure states:

$$F_\alpha - F_\beta = \mathcal{O}(1), \qquad \text{even if } N \to \infty.$$

Infinitely many states

- From the previous discussion, we know that there are infinitely many relevant pure states:

$$F_\alpha - F_\beta = \mathcal{O}(1), \qquad \text{even if } N \to \infty.$$

- It makes sense to consider restricted averages: $\langle A \rangle_\alpha$, so that intensive quantities do not fluctuate in $\langle \cdots \rangle_\alpha$

Infinitely many states

- From the previous discussion, we know that there are infinitely many relevant pure states:

$$F_\alpha - F_\beta = \mathcal{O}(1), \qquad \text{even if } N \to \infty.$$

- It makes sense to consider restricted averages: $\langle A \rangle_\alpha$, so that intensive quantities do not fluctuate in $\langle \cdots \rangle_\alpha$
- Each state will have a weight w_α

$$\langle A \rangle = \sum_\alpha w_\alpha \langle A \rangle_\alpha .$$

(ferromagnet: consider $\langle A \rangle = \frac{1}{2} \langle A \rangle_+ + \frac{1}{2} \langle A \rangle_-$).

Infinitely many states

- From the previous discussion, we know that there are infinitely many relevant pure states:

$$F_\alpha - F_\beta = \mathcal{O}(1), \qquad \text{even if } N \to \infty.$$

- It makes sense to consider restricted averages: $\langle A \rangle_\alpha$, so that intensive quantities do not fluctuate in $\langle \cdots \rangle_\alpha$
- Each state will have a weight w_α

$$\langle A \rangle = \sum_\alpha w_\alpha \langle A \rangle_\alpha .$$

- Example: the overlap:

$$q_{\alpha\beta} = \frac{1}{N} \sum_i \langle s_i \rangle_\alpha \langle s_i \rangle_\beta \quad \Longrightarrow \quad p(q) = \overline{\sum_{\alpha\beta} w_\alpha w_\beta \delta(q - q_{\alpha\beta})}.$$

Infinitely many states

- From the previous discussion, we know that there are infinitely many relevant pure states:

$$F_\alpha - F_\beta = \mathcal{O}(1), \qquad \text{even if } N \to \infty.$$

- It makes sense to consider restricted averages: $\langle A \rangle_\alpha$, so that intensive quantities do not fluctuate in $\langle \cdots \rangle_\alpha$
- Each state will have a weight w_α

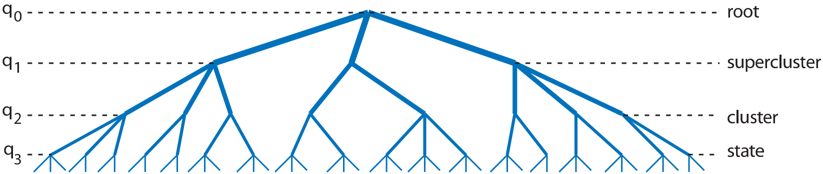
$$\langle A \rangle = \sum_\alpha w_\alpha \langle A \rangle_\alpha .$$

- Example: the overlap:

$$q_{\alpha\beta} = \frac{1}{N} \sum_i \langle s_i \rangle_\alpha \langle s_i \rangle_\beta \quad \Longrightarrow \quad p(q) = \overline{\sum_{\alpha\beta} w_\alpha w_\beta \delta(q - q_{\alpha\beta})}.$$

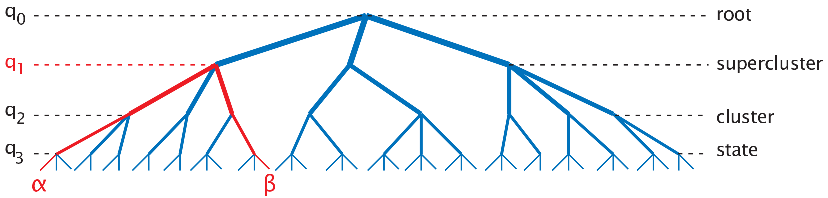
- It turns out that the states live in an ultrametric space, using $q_{\alpha\beta}$ to define a distance
- We can classify the states in a taxonomic tree, which branches out as we break the replica symmetry.

The tree of states (I)



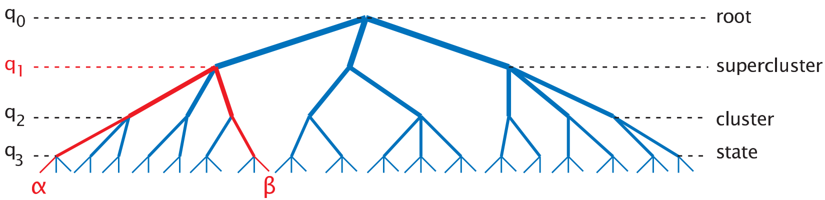
- Consider a simplified case where q can only take four discrete values (equivalently $K = 3$ steps of RSB).
- We can group the configurations (organisms) $\rightarrow$ states (species) $\rightarrow$ clusters (geni) $\rightarrow$ superclusters (families).

The tree of states (I)



- Consider a simplified case where q can only take four discrete values (equivalently $K = 3$ steps of RSB).
- We can group the configurations $\rightarrow$ states $\rightarrow$ clusters $\rightarrow$ superclusters .
- It makes sense, because the overlap (distance) depends only on the first common ancestor: $q_{\alpha\beta} = q_1$.

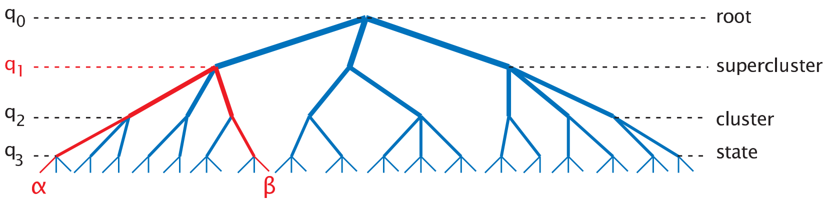
The tree of states (I)



- Consider a simplified case where q can only take four discrete values (equivalently $K = 3$ steps of RSB).
- We can group the configurations $\rightarrow$ states $\rightarrow$ clusters $\rightarrow$ superclusters .
- It makes sense, because the overlap (distance) depends only on the first common ancestor: $q_{\alpha\beta} = q_1$.
- Notice that the self-overlap is the same for all states:

$$q_{\alpha\alpha} = q_{\mathrm{M}}, \quad \forall\alpha$$

The tree of states (I)



- Consider a simplified case where q can only take four discrete values (equivalently $K = 3$ steps of RSB).
- We can group the configurations $\rightarrow$ states $\rightarrow$ clusters $\rightarrow$ superclusters .
- It makes sense, because the overlap (distance) depends only on the first common ancestor: $q_{\alpha\beta} = q_1$.
- Notice that the self-overlap is the same for all states:

$$q_{\alpha\alpha} = q_{\mathrm{M}}, \quad \forall\alpha$$

- We can consider a decomposition in clusters, instead of states:

$$\langle A \rangle = \sum_I W_I \langle A \rangle_I , \qquad W_I = \sum_{\alpha\in I} w_\alpha .$$

The tree of states (II)

- The real tree of states of a mean-field spin glass is more complicated:
  - Infinitely many levels: the tree branches out for any value of $q < q_{\mathrm{M}}$.
  - Infinite number of branches at any level.

The tree of states (II)

- The real tree of states of a mean-field spin glass is more complicated:
  - Infinitely many levels: the tree branches out for any value of $q < q_{\mathrm{M}}$.
  - Infinite number of branches at any level.
- Amusing analogy with QED: number of emitted photons diverges at low energy $\leftrightarrow$ infinite number of branches as we consider small values of w.

The tree of states (II)

- The real tree of states of a mean-field spin glass is more complicated:
  - Infinitely many levels: the tree branches out for any value of $q < q_{\mathrm{M}}$.
  - Infinite number of branches at any level.
- Amusing analogy with QED: number of emitted photons diverges at low energy $\leftrightarrow$ infinite number of branches as we consider small values of w.
- However, it is possible to compute analytically the distribution of the W_I, at any level in q:

$$P(W;q) = \frac{W^{x(q)-1}(1-W)^{x(q)-1}}{\Gamma(1-x(q))\Gamma(x(q))} .$$

The tree of states (II)

- The real tree of states of a mean-field spin glass is more complicated:
  - Infinitely many levels: the tree branches out for any value of $q < q_{\mathrm{M}}$.
  - Infinite number of branches at any level.
- Amusing analogy with QED: number of emitted photons diverges at low energy $\leftrightarrow$ infinite number of branches as we consider small values of w.
- However, it is possible to compute analytically the distribution of the W_I, at any level in q:

$$P(W;q) = \frac{W^{x(q)-1}(1-W)^{x(q)-1}}{\Gamma(1-x(q))\Gamma(x(q))} \,.$$

- $\sum_I W_I = 1 \implies$ the weights are not independent:

$$P(W,W';q) = x(q)\frac{\Theta(1-W-W')(WW')^{x(q)-1}(1-W-W')^{2x(q)-1}}{\Gamma(1-x(q))\Gamma(1-x(q))\Gamma(2x(q))} \,.$$

The tree of states (II)

- The real tree of states of a mean-field spin glass is more complicated:
  - Infinitely many levels: the tree branches out for any value of $q < q_{\mathrm{M}}$.
  - Infinite number of branches at any level.
- Amusing analogy with QED: number of emitted photons diverges at low energy $\leftrightarrow$ infinite number of branches as we consider small values of w.
- However, it is possible to compute analytically the distribution of the W_l, at any level in q:

$$P(W;q) = \frac{W^{x(q)-1}(1-W)^{x(q)-1}}{\Gamma(1-x(q))\Gamma(x(q))} .$$

- $\sum_l W_l = 1 \implies$ the weights are not independent:

$$P(W,W';q) = x(q)\frac{\Theta(1-W-W')(WW')^{x(q)-1}(1-W-W')^{2x(q)-1}}{\Gamma(1-x(q))\Gamma(1-x(q))\Gamma(2x(q))} .$$

- We want to generate explicit realisations of this tree.

Free-energy fluctuations

- The W_l are cumbersome to handle.
- We consider instead the free-energy fluctuations.

Free-energy fluctuations

- The W_l are cumbersome to handle.
- We consider instead the free-energy fluctuations.
- The free energy per spin of each state has fluctuations $f_\alpha/N = \mathcal{O}(1/N)$.

Free-energy fluctuations

- The W_l are cumbersome to handle.
- We consider instead the free-energy fluctuations.
- The free energy per spin of each state has fluctuations $f_\alpha/N = \mathcal{O}(1/N)$.
- The f_α are related to the w_α:

$$w_\alpha = \frac{\mathrm{e}^{-\beta f_\alpha}}{\sum_\gamma \mathrm{e}^{-\beta f_\gamma}} .$$

Free-energy fluctuations

- The W_l are cumbersome to handle.
- We consider instead the free-energy fluctuations.
- The free energy per spin of each state has fluctuations $f_\alpha / N = \mathcal{O}(1/N)$.
- The f_α are related to the w_α:

$$w_\alpha = \frac{e^{-\beta f_\alpha}}{\sum_\gamma e^{-\beta f_\gamma}} .$$

- However, it turns out that the f_α are independent:

$$\mathcal{P}(f_\alpha) \propto e^{-\beta x(q_M) f}$$

Free-energy fluctuations

- The W_I are cumbersome to handle.
- We consider instead the free-energy fluctuations.
- The free energy per spin of each state has fluctuations $f_\alpha/N = \mathcal{O}(1/N)$.
- The f_α are related to the w_α:

$$w_\alpha = \frac{e^{-\beta f_\alpha}}{\sum_\gamma e^{-\beta f_\gamma}} \ .$$

- However, it turns out that the f_α are independent:

$$\mathcal{P}(f_\alpha) \propto e^{-\beta x(q_\mathrm{M}) f}$$

- We can do this at any level of q:

$$W_I = \frac{e^{-\beta f_I}}{\sum_J e^{-\beta f_J}}, \qquad \mathcal{P}_q(f) \propto e^{-\beta x(q) f}.$$

Free-energy fluctuations

- The W_I are cumbersome to handle.
- We consider instead the free-energy fluctuations.
- The free energy per spin of each state has fluctuations $f_\alpha/N = \mathcal{O}(1/N)$.
- The f_α are related to the w_α:

$$w_\alpha = \frac{\mathrm{e}^{-\beta f_\alpha}}{\sum_\gamma \mathrm{e}^{-\beta f_\gamma}}\ .$$

- However, it turns out that the f_α are independent:

$$\mathcal{P}(f_\alpha) \propto \mathrm{e}^{-\beta x(q_\mathrm{M}) f}$$

- We can do this at any level of q:

$$W_I = \frac{\mathrm{e}^{-\beta f_I}}{\sum_J \mathrm{e}^{-\beta f_J}}, \qquad \mathcal{P}_q(f) \propto \mathrm{e}^{-\beta x(q) f}.$$

- Universality: everything is encoded in $x(q)$.

Generating the tree from the ground up (I)

- The formulation of the f_l considering a single q level in isolation is convenient for analytical computations.
- Here, we need a different approach.
- We will generate the tree step by step, beginning by the root and down to the states, at each step computing the weights.

Generating the tree from the ground up (I)

- The formulation of the f_I considering a single q level in isolation is convenient for analytical computations.
- Here, we need a different approach.
- We will generate the tree step by step, beginning by the root and down to the states, at each step computing the weights.
- We start considering a discretised $q(x)$ (equivalently, a finite number K of RSB steps).

Generating the tree from the ground up (I)

- The formulation of the f_l considering a single q level in isolation is convenient for analytical computations.
- Here, we need a different approach.
- We will generate the tree step by step, beginning by the root and down to the states, at each step computing the weights.
- We start considering a discretised $q(x)$ (equivalently, a finite number K of RSB steps).
- We have, therefore, $K + 1$ levels.

Generating the tree from the ground up (I)

- The formulation of the f_l considering a single q level in isolation is convenient for analytical computations.
- Here, we need a different approach.
- We will generate the tree step by step, beginning by the root and down to the states, at each step computing the weights.
- We start considering a discretised $q(x)$ (equivalently, a finite number K of RSB steps).
- We have, therefore, $K+1$ levels.
- In addition, since we cannot handle an infinite number of states, we will 'prune' the tree.

Generating the tree from the ground up (I)

- The formulation of the f_l considering a single q level in isolation is convenient for analytical computations.
- Here, we need a different approach.
- We will generate the tree step by step, beginning by the root and down to the states, at each step computing the weights.
- We start considering a discretised $q(x)$ (equivalently, a finite number K of RSB steps).
- We have, therefore, $K+1$ levels.
- In addition, since we cannot handle an infinite number of states, we will 'prune' the tree.
- We eliminate at each level all the clusters with $W_l < \epsilon$ (equivalent to neglecting all the states with $w_\alpha < \epsilon$).
- We are losing a total probability of $\sim \epsilon^{1-x(q_\mathrm{M})}$.

Generating the tree from the ground up (II)

- Given a cluster at level q_i, we want its subclusters at level q_{i+1}.

Generating the tree from the ground up (II)

- Given a cluster at level q_i, we want its subclusters at level q_{i+1}.
- We introduce M variables $f_1, \ldots, f_M$. They are not independent:

$$\mathcal{P}_{q_i \to q_{i+1}}(f_1, \ldots, f_M)) \propto \exp\left[-\beta x(q_{i+1}) \sum_{i=1}^{M} f_i\right] \left[\sum_{i=1}^{M} \exp(-\beta f_i)\right]^{x(q_i)}.$$

Generating the tree from the ground up (II)

- Given a cluster at level q_i, we want its subclusters at level q_{i+1}.
- We introduce M variables $f_1, \ldots, f_M$. They are not independent:

$$\mathcal{P}_{q_i \to q_{i+1}}(f_1, \ldots, f_M)) \propto \exp\left[-\beta x(q_{i+1}) \sum_{i=1}^{M} f_i\right] \left[\sum_{i=1}^{M} \exp(-\beta f_i)\right]^{x(q_i)} .$$

- Now the weights of the subclusters are (W = weight of the cluster at q_i)

$$w_i = W \frac{\exp(-\beta f_i)}{\sum_{i=1}^{M} \exp(-\beta f_i)} .$$

Generating the tree from the ground up (II)

- Given a cluster at level q_i, we want its subclusters at level q_{i+1}.
- We introduce M variables $f_1, \ldots, f_M$. They are not independent:

$$\mathcal{P}_{q_i \to q_{i+1}}(f_1, \ldots, f_M)) \propto \exp\left[-\beta x(q_{i+1}) \sum_{i=1}^{M} f_i\right] \left[\sum_{i=1}^{M} \exp(-\beta f_i)\right]^{x(q_i)} .$$

- Now the weights of the subclusters are (W = weight of the cluster at q_i)

$$w_i = W \frac{\exp(-\beta f_i)}{\sum_{i=1}^{M} \exp(-\beta f_i)} .$$

- It is not immediately obvious, but this method generates the same probability distributions for the f_α.
- This is because the correlation in the f_i of the subclusters at level q_{i+1} compensates the correlations of the weights of the clusters at q_i.

Generating the tree from the ground up (II)

- Given a cluster at level q_i, we want its subclusters at level q_{i+1}.
- We introduce M variables $f_1, \ldots, f_M$. They are not independent:

$$\mathcal{P}_{q_i \to q_{i+1}}(f_1, \ldots, f_M)) \propto \exp\left[-\beta x(q_{i+1}) \sum_{i=1}^{M} f_i\right] \left[\sum_{i=1}^{M} \exp(-\beta f_i)\right]^{x(q_i)} .$$

- Now the weights of the subclusters are (W = weight of the cluster at q_i)

$$w_i = W \frac{\exp(-\beta f_i)}{\sum_{i=1}^{M} \exp(-\beta f_i)} .$$

- It is not immediately obvious, but this method generates the same probability distributions for the f_α.
- This is because the correlation in the f_i of the subclusters at level q_{i+1} compensates the correlations of the weights of the clusters at q_i.
- Notice that the first step, the root, is going from $q = 0 \to q_0$. Since $x(q = 0) = 0$, in the first step we have independent f_i.

The cavity method

- We have seen how to generate the tree, we now want to compute physical quantities from it. We consider a cavity approach.
- We add an $N+1$ spin s_0 to a system with N spins s_i.
- The average properties of s_0 are the same as those of the s_i.
- Consider the N-spin system in equilibrium and compute the cavity field

$$h = \sum_{k=1}^{N} J_{0k} s_k$$

The cavity method

- We have seen how to generate the tree, we now want to compute physical quantities from it. We consider a cavity approach.
- We add an $N+1$ spin s_0 to a system with N spins s_i.
- The average properties of s_0 are the same as those of the s_i.
- Consider the N-spin system in equilibrium and compute the cavity field

$$h = \sum_{k=1}^{N} J_{0k} s_k$$

- The cavity field h_α for each state is the sum of $K+1$ Gaussian random variables:

$$h_\alpha = h_\alpha^0 + h_\alpha^1 + \ldots + h_\alpha^K,$$

  - h_α^0 has variance βq_0 and is common to the whole tree.
  - h_α^i has variance $\beta(q_i - q_{i-1})$ and is common to all the states down the same branch.

The cavity method

- We have seen how to generate the tree, we now want to compute physical quantities from it. We consider a cavity approach.
- We add an $N+1$ spin s_0 to a system with N spins s_i.
- The average properties of s_0 are the same as those of the s_i.
- Consider the N-spin system in equilibrium and compute the cavity field

$$h = \sum_{k=1}^{N} J_{0k} s_k$$

- The cavity field h_α for each state is the sum of $K+1$ Gaussian random variables:

$$h_\alpha = h_\alpha^0 + h_\alpha^1 + \ldots + h_\alpha^K,$$

- Example: $m_\alpha = \tanh(\beta h_\alpha)$.

The cavity method

- We have seen how to generate the tree, we now want to compute physical quantities from it. We consider a cavity approach.
- We add an $N+1$ spin s_0 to a system with N spins s_i.
- The average properties of s_0 are the same as those of the s_i.
- Consider the N-spin system in equilibrium and compute the cavity field

$$h = \sum_{k=1}^{N} J_{0k} s_k$$

- The cavity field h_α for each state is the sum of $K+1$ Gaussian random variables:

$$h_\alpha = h_\alpha^0 + h_\alpha^1 + \ldots + h_\alpha^K,$$

- Example: $m_\alpha = \tanh(\beta h_\alpha)$.
- The cavity step shifts the free energies:

$$w'_\alpha \sim w_\alpha \cosh(\beta h_\alpha)$$

$\Longrightarrow$ we have an iterative method to refine the tree

Testing the program

- We have described how to generate the tree given a $q(x)$.
- Also, how to refine the w_α with a cavity step
  (and with them get a new $q'(x)$).

Testing the program

- We have described how to generate the tree given a $q(x)$.
- Also, how to refine the w_α with a cavity step (and with them get a new $q'(x)$).
- As a consistency check of the program, we can try to find $q(x)$ at a given T, starting from an educated guess.

Testing the program

- We have described how to generate the tree given a $q(x)$.
- Also, how to refine the w_α with a cavity step
(and with them get a new $q'(x)$).
- As a consistency check of the program, we can try to find $q(x)$ at a given T, starting from an educated guess.
- We consider $T = 0.85$, close to the critical point ($T_\mathrm{c} = 1$),

Testing the program

- We have described how to generate the tree given a $q(x)$.
- Also, how to refine the w_α with a cavity step (and with them get a new $q'(x)$).
- As a consistency check of the program, we can try to find $q(x)$ at a given T, starting from an educated guess.
- We consider $T = 0.85$, close to the critical point ($T_c = 1$),
- In these conditions, $q(x)$ is linear with a very good approximation.

Testing the program

- We have described how to generate the tree given a $q(x)$.
- Also, how to refine the w_α with a cavity step (and with them get a new $q'(x)$).
- As a consistency check of the program, we can try to find $q(x)$ at a given T, starting from an educated guess.
- We consider $T = 0.85$, close to the critical point ($T_\mathrm{c} = 1$),
- In these conditions, $q(x)$ is linear with a very good approximation.
- We only need to find two parameters: q_M and x_M.

Testing the program

- We have described how to generate the tree given a $q(x)$.
- Also, how to refine the w_α with a cavity step (and with them get a new $q'(x)$).
- As a consistency check of the program, we can try to find $q(x)$ at a given T, starting from an educated guess.
- We consider $T = 0.85$, close to the critical point ($T_\mathrm{c} = 1$),
- In these conditions, $q(x)$ is linear with a very good approximation.
- We only need to find two parameters: q_M and x_M.
    1. Find the correct q_M for a fixed x_M and compute $F(x_\mathrm{M})$.
    2. Minimize $F(x_\mathrm{m})$ to find the correct x_M.

Testing the program

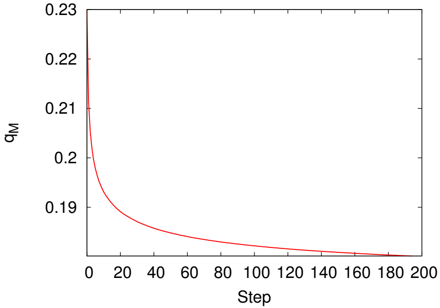
- We have described how to generate the tree given a $q(x)$.
- Also, how to refine the w_α with a cavity step
(and with them get a new $q'(x)$).
- As a consistency check of the program, we can try to find $q(x)$ at a given T, starting from an educated guess.
- We consider $T = 0.85$, close to the critical point ($T_\mathrm{c} = 1$),
- In these conditions, $q(x)$ is linear with a very good approximation.
- We only need to find two parameters: q_M and x_M.
  1. Find the correct q_M for a fixed x_M and compute $F(x_\mathrm{M})$.
  2. Minimize $F(x_\mathrm{m})$ to find the correct x_M.
- Let us study the solution at x_M fixed to the known correct value
$x_\mathrm{M} \approx 0.233122$.

$q(x)$ at $T = 0.85$

- We start with $q_{\mathrm{M}} = 0.23 \approx x_{\mathrm{M}}$ and iterate.
- We consider $\epsilon = 10^{-5} \Longrightarrow \sim 1 - \epsilon^{1-x_{\mathrm{M}}} = 99.99\%$ of the probability.
- $K = 20$ is enough, since $q(x)$ is so simple.
- We generate 10^6 samples (trees) and iterate
- Each iteration takes ≈ 2 min in a single CPU.

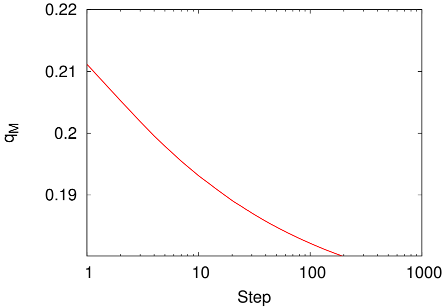
$q(x)$ at $T = 0.85$

- We start with $q_{\mathrm{M}} = 0.23 \approx x_{\mathrm{M}}$ and iterate.
- We consider $\epsilon = 10^{-5} \Longrightarrow \sim 1 - \epsilon^{1-x_{\mathrm{M}}} = 99.99\%$ of the probability.
- $K = 20$ is enough, since $q(x)$ is so simple.
- We generate 10^6 samples (trees) and iterate
- Each iteration takes ≈ 2 min in a single CPU.



$q(x)$ at $T = 0.85$

- We start with $q_M = 0.23 \approx x_M$ and iterate.
- We consider $\epsilon = 10^{-5} \Longrightarrow \sim 1 - \epsilon^{1-x_M} = 99.99\%$ of the probability.
- $K = 20$ is enough, since $q(x)$ is so simple.
- We generate 10^6 samples (trees) and iterate
- Each iteration takes ≈ 2 min in a single CPU.



The convergence is very slow (logarithmic).

$q(x)$ at $T = 0.85$

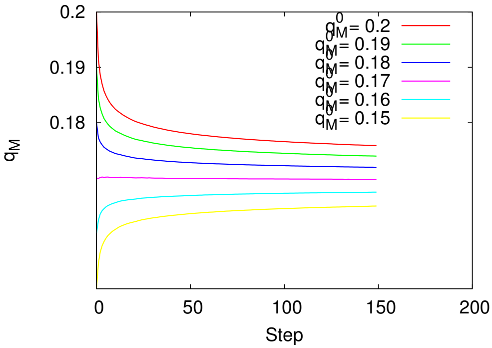
- We need a different approach.

$q(x)$ at $T = 0.85$

- We need a different approach.
- Start several computations, sweeping a wide range of starting q_{M}:

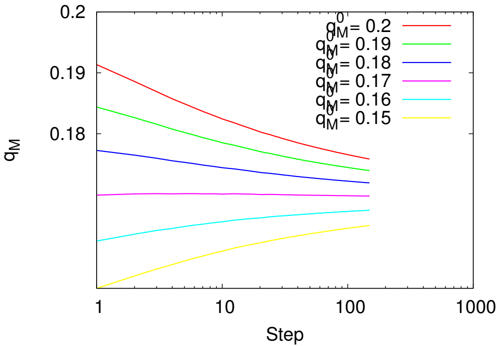
$q(x)$ at $T = 0.85$

- We need a different approach.
- Start several computations, sweeping a wide range of starting q_M:



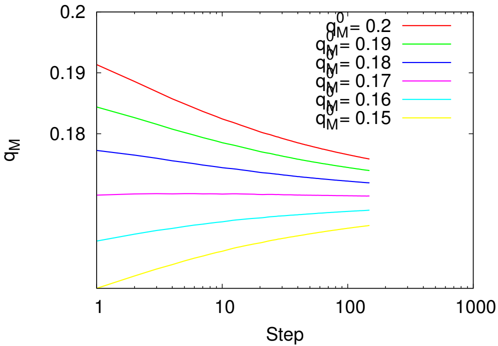
$q(x)$ at $T = 0.85$

- We need a different approach.
- Start several computations, sweeping a wide range of starting q_{M}:



- The evolution is monotonic $\Longrightarrow$ it is easy to find the stable solution.

- We need a different approach.
- Start several computations, sweeping a wide range of starting q_{M}:



- The evolution is monotonic $\Longrightarrow$ it is easy to find the stable solution.
- After 100 steps, starting with $q_{\mathrm{M}}^{(0)} = 0.17$, we find $q_M^{(100)} = 0.169687(3)$ (expected value $q_{\mathrm{M}} \approx 0.169691$).

Replicon

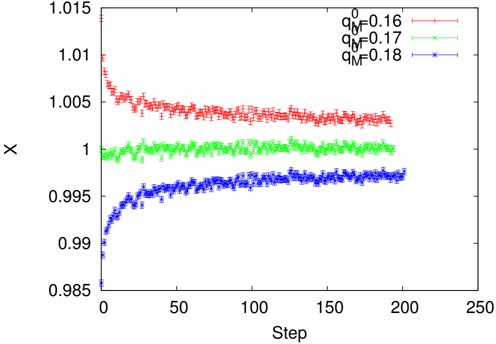
- The spin-glass susceptibility is $\chi_{\mathrm{SG}} = \overline{\left((1 - m_0^2)^2\right)^2} / \left[1 - \beta^2 \overline{(1 - m_0^2)^2}\right]$.
- It diverges for $T < T_{\mathrm{c}}$ so

$$X = \beta^2 \overline{(1 - m_0^2)^2} = \beta^2 \overline{\sum_\alpha w_\alpha (1 - m_\alpha^2)^2} = 1.$$

Replicon

- The spin-glass susceptibility is $\chi_{\mathrm{SG}} = \overline{((1-m_0^2)^2)^2}/\left[1-\beta^2\overline{(1-m_0^2)^2}\right]$.
- It diverges for $T < T_{\mathrm{c}}$ so

$$X = \beta^2\overline{(1-m_0^2)^2} = \beta^2\overline{\sum_\alpha w_\alpha(1-m_\alpha^2)^2} = 1.$$



- We obtain $X = 0.99972(33)$.

Conclusions

- We have reviewed the main properties of the tree of states in mean-field spin glasses.
- We have shown how to generate explicit realisations of this tree, in a self-consistent way.
- The states computed from the tree can be used to study physical quantities.
- Applications
    - Evaluate all the correlation functions of the model for fixed q.
    - Study other mean-field models (such as the full-RSB solution for the spin glass in a Bethe lattice).
    - Finite-size effects: generate the $P_{\mathcal{J}}(q)$ for single trees (single samples) and study the smoothing of the individual peaks.

Conclusions

- We have reviewed the main properties of the tree of states in mean-field spin glasses.
- We have shown how to generate explicit realisations of this tree, in a self-consistent way.
- The states computed from the tree can be used to study physical quantities.
- Applications
  - Evaluate all the correlation functions of the model for fixed q.
  - Study other mean-field models (such as the full-RSB solution for the spin glass in a Bethe lattice).
  - Finite-size effects: generate the $P_{\mathcal{J}}(q)$ for single trees (single samples) and study the smoothing of the individual peaks.

THANK YOU!