Graphical models and the Ising model

More on matchings: imposing constraints

Let's get back to one of our previous homework the matching problem. We remind: Given a (unweighted, undirected) graph $G(V, E)$ a matching $\mathsf{M} \in E$ is defined as a subset of edges such that if $(ij) \in \mathsf{M}$ then no other edge that contains node i or j can be in M. In other words a matching is a subset of edges such that no two edges of the set share a node. The size of a matching is simply $|\mathsf{M}|$.

(a) Last time you represented this problem via a graphical model. It is crucial to realize that usually there are several possible ways to do it.

This time the task is to write a graphical model for matchings such that if the original graph is a tree (i.e. has no loops) then also the graphical model is a tree (i.e. has no cycles when seen as a simple bipartite graph).

Hint: Factor nodes can have more than two neighbors.

Suggestion: Use variables $s_{(ij)} = 1$ is $(ij) \in \mathsf{M}$ and $s_{(ij)} = 0$ if $(ij) \notin \mathsf{M}$, express the "matching" constraint in terms of these variables, express also the size of the matching $e = |\mathsf{M}|/N$ in terms of these variables.

(b) Generalize the above to the problem of k-factor of a graph. A k-factor is a subset of edges such that every node has at most k edges in the k-factor. With this definition matching is a 1-factor.

(c) Consider a probability measure

$$P\left(\left\{s_{(ij)}\right\}_{(ij)\in E}\right) = \frac{1}{Z(\beta)} \prod_{(ij)\in E} \mathrm{e}^{\beta s_{(ij)}} \prod_{i=1}^{N} \mathbb{I}\left(\sum_{j\in\partial i} s_{(ij)} \leq k\right) \tag{1}$$

Let us introduce the entropy function $s(e)$ in such a way that if $\mathcal{N}(e)$ is the number of matchings of size Ne then $s(e) = \lim_{N\to\infty} \frac{1}{N} \log\left(\mathcal{N}(e)\right)$. For the purpose of the HW we will simply assume this limit exists. Assume you can compute $Z(\beta)$ and also the limit $f(\beta) \equiv \lim_{N\to\infty} \frac{1}{N} \log\left(Z(\beta)\right)$ (we will learn how very soon). How do you use $f(\beta)$ to compute $s(e)$? Additionally, use your intuition to sketch how will $s(e)$ look like for a random r-regular graph $r > k$.

Solution.

(a) Since we can use a length-$|E|$ configuration $\left\{s_{(ij)}\right\}_{(ij)\in E}$ to represent any set $\mathsf{M} \subseteq E$ by

$$s_{(ij)} = \begin{cases} 1, & \text{if edge } (ij) \text{ is in } \mathsf{M} \\ 0, & \text{if edge } (ij) \text{ is not in } \mathsf{M} \end{cases}.$$

Now we already know that each variable node represents, the next is to define the factor nodes. However, to ensure M is a valid matching, there are two valid ways to define the factor nodes:

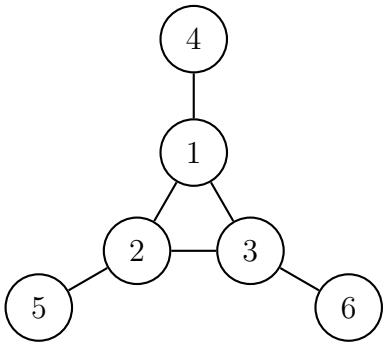
- For any node $i \in V$, at most one edge in M can be connected to it. For this approach, we have $|V|$ factor nodes f_i, each has function

$$\psi_i\left(\left\{s_{(ij)}\right\}_{j\in\partial i}\right) = \mathbb{I}\left(\sum_{j\in\partial i} s_{(ij)} \leq 1\right), \qquad \forall\, i \in V$$

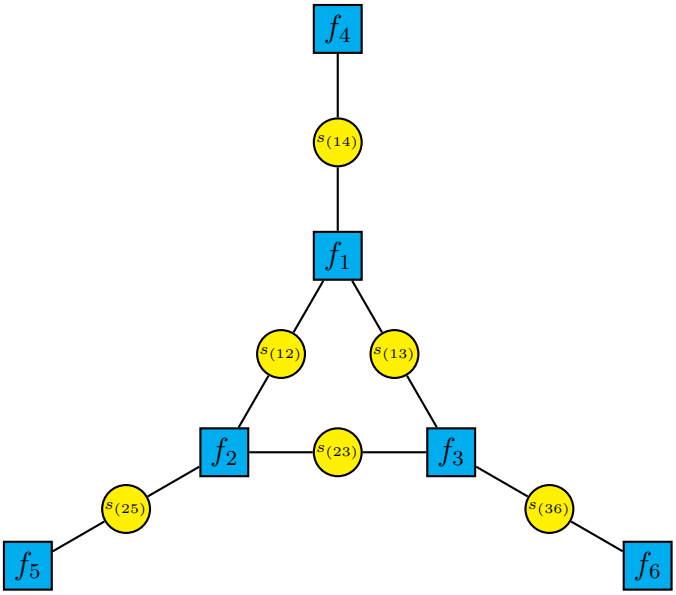
- For any two edges $e_1, e_2 \in \mathsf{M}$, they do not share end nodes. For this approach, we have $\sum_{i\in V}\binom{|\partial i|}{2}\mathbb{I}\left(|\partial i| \geq 2\right)$ factor nodes $f_{ij,ik}$, each has function

$$\psi_{ij,ik}\left(s_{(ij)}, s_{(ik)}\right) = \mathbb{I}\left(s_{(ij)} + s_{(ik)} < 2\right), \qquad \forall\, i \in V, \quad \forall\, j,k \in \partial i, \quad j \neq k.$$

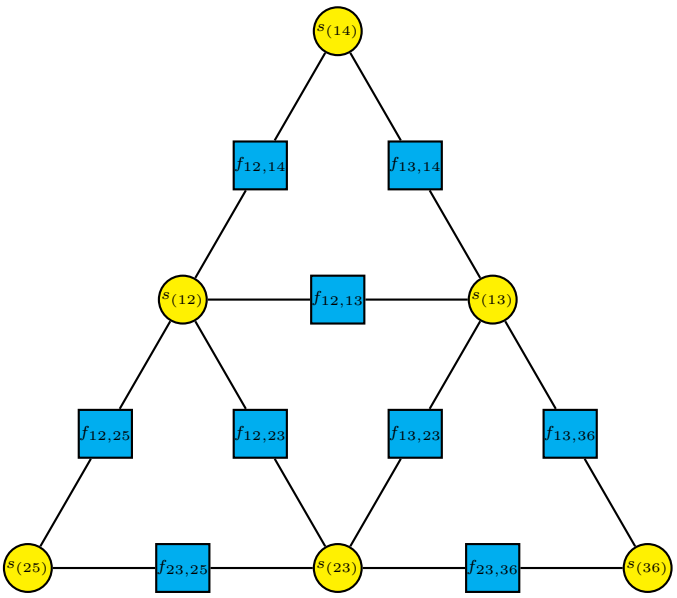
For example, if $G(V,E)$ looks like



The factor graph of the former way is



The factor graph of the latter way is



So which one is better? In the later lectures we will use belief-propagation (BP) algorithm to solve this problem. BP is exact on a tree structure but can only give an approximation if the graph is loopy. The second way has many short cycles so if we run BP on it, BP will give worse approximation.

(b) Using the first way to construct factor graph, the corresponding Boltzmann distribution is

$$P\left(\left\{s_{(ij)}\right\}_{(ij)\in E}\right) = \frac{1}{Z}\prod_{i\in V}\psi_i\left(\left\{s_{(ij)}\right\}_{j\in\partial i}\right) = \frac{1}{Z}\prod_{i\in V}\mathbb{I}\left(\sum_{j\in\partial i}s_{(ij)}\leq 1\right)$$

To generalize the problem into k-factor problem, we only need to replace the function in the factor nodes to be

$$\psi_i\left(\left\{s_{(ij)}\right\}_{j\in\partial i}\right) = \mathbb{I}\left(\sum_{j\in\partial i}s_{(ij)} = k\right), \qquad \forall\, i\in V$$

With this definition a *perfect* matching is a 1-factor.

(c) For notation simplicity, in this sub-question I will write $\vec{s}\triangleq\left\{s_{(ij)}\right\}_{(ij)\in E}$, notice that given a matching $\vec{s}$, from the definition of size in the suggestion its size is $\frac{1}{N}\sum_{(ij)\in E}s_{(ij)}$. Let $\mathcal{N}(e)$ denote the number of (generalized) matchings of size e , deriving from the definition of partition function yields

$$Z(\beta) = \sum_{\{\vec{s}\}}\prod_{(ij)\in E}\mathrm{e}^{\beta s_{(ij)}}\underbrace{\prod_{i=1}^{N}\mathbb{I}\left(\sum_{j\in\partial i}s_{(ij)}\leq k\right)}_{\text{condition to guarantee matching}}$$

$$\text{absorb delta into sum range} \quad = \sum_{\{\vec{s}\,|\,\vec{s}\text{ is matching}\}}\exp\left(\beta\sum_{(ij)\in E}s_{(ij)}\right)$$

$$\text{group matching by size} \quad = \int_0^{\frac{|E|}{N}}\mathrm{d}e\,\mathcal{N}(e)\mathrm{e}^{N\beta e} = \int_0^{\frac{|E|}{N}}\mathrm{d}e\,\exp\left\{N\left[\frac{1}{N}\log\left(\mathcal{N}(e)\right)+\beta e\right]\right\}$$

$$\text{Laplace's method} \quad \simeq \exp\left(N\left[s(e^*)+\beta e^*\right]\right)$$

where the maximizer $e^* = \arg\max_{e\in[0,|E|/N]}\left[s(e)+\beta e\right]$ is actually a function of β.

Suppose now we know how to compute the partition function $Z(\beta)$ and also the free entropy density $f(\beta)$, then

$$f(\beta) \equiv \lim_{N\to\infty} \frac{1}{N} \log\left(Z(\beta)\right) = s(e^*) + \beta e^* = \sup_{e\in[0,|E|/N]} [s(e) + \beta e]$$

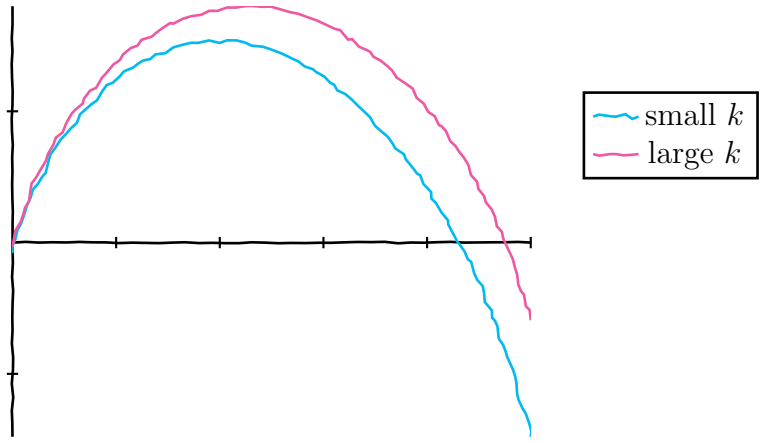
We can extend $s(e)$ to $e \in \mathbb{R}$ by setting $s(e) = -\infty$ for $e \notin [0, |E|/N]$. Therefore by Legendre transformation

$$\begin{aligned} & f(\beta) = \sup_{e\in\mathbb{R}} [s(e) + \beta e] = \sup_{e\in\mathbb{R}} [\beta e - (-s(e))] \\ \Rightarrow \quad & -s(e) = \sup_{\beta\in\mathbb{R}} [\beta e - f(\beta)] \\ \Rightarrow \quad & s(e) = -\sup_{\beta\in\mathbb{R}} [\beta e - f(\beta)] = \inf_{\beta\in\mathbb{R}} [f(\beta) - \beta e] \end{aligned}$$

For a random r-regular graph with $r > k$, there are $Nr/2$ edges.

- When e is small, if we pick Ne edges uniformly, it is very probable that none of the nodes is connected to larger than k edges, so $s(e)$ should increase from 0 because $\log \binom{Nr/2}{Ne}$ is increasing.
- When e is larger, the constraints start to play an important role to filter out many subset of edges, so $s(e)$ should decrease from somewhere between 0 and 1.
- If e keeps going larger, the constraints will be hard and hard to satisfy until impossible to satisfy, so $s(e)$ will keep decreasing and become negative from some point afterwards.
- The set of all k-factors is a strict subset of the set of all $(k+1)$-factors. Furthermore, at fixed e, the entropy function of k-factor should be smaller than $(k+1)$-factor since the constraints of $(k+1)$-factor are looser. There are some size=e subsets of edges is a $(k+1)$-factor but not a k-factor.

The sketch of $s(e)$ for small k (blue) and large k (red) are given below:



□

Curie-Weiss is Back

The Hamiltonian of the Curie-Weiss model reads

$$\mathcal{H}\left(\{s_i\}_{i=1}^N\right) = -\frac{1}{2N}\left(\sum s_i\right)^2 - h\sum_i s_i\,.$$

As we have seen in our first set of lectures, the free energy $f(\beta, h) = -\frac{1}{\beta} \lim_{N\to\infty} \log\left(Z_N(\beta, h)\right)$ of the model can be written (asymptotically) as $f(\beta, h) = \min_{m\in[-1,1]} \mathcal{F}_m(\beta, h, m)$, with the free energy at fixed magnetization $\mathcal{F}_m(\beta, h, m)$ being given by:

$$\mathcal{F}_m(\beta, h, m) = -\frac{1}{2}m^2 - hm + \frac{1}{\beta}\left[\frac{1+m}{2}\log\left(\frac{1+m}{2}\right) + \frac{1-m}{2}\log\left(\frac{1-m}{2}\right)\right]$$

and the equilibrium value of the magnetization is given by the minimizer m^*:

$$\left\langle \frac{\sum_i s_i}{N} \right\rangle = m^*$$

(a) Plot the function $\mathcal{F}_m(\beta, h, m)$ as a function of m in the two following cases: (a) at $h = 0$ for different values of β larger and lower than one and (b) at a value of β larger than 1 for different values (positive and negative) of h. Describe what you see in both cases.

(b) Show [1] that the minimizer m^* is also the solution of the so-called "self-consistent equation"

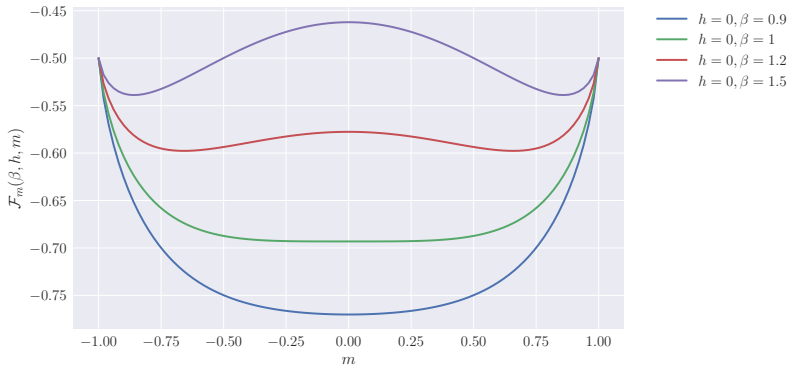
$$m = \tanh(\beta(m + h)) \,.$$

Compute the value of m^* in the three following cases: (a) $h = 10^{-6}$ and β^{-1} between 0 and 2, (b) $h = -10^{-6}$ and β^{-1} between 0 and 2, and (c) $\beta = 1.5$ and h between -1 and 1.

(c) We now focus on the behavior at $\beta = 1.5$ with h between 0.1 and 0.2: how many solution to the self-consistent equations are they? Which one is the correct one according to the Laplace method (plot the function $\mathcal{F}_m(\beta, h, m)$ to answer this question).

Solution.

(a) The plot of function $\mathcal{F}_m(\beta, h, m)$ under $h = 0$ and $\beta = 0.9, 1, 1.2, 1.5$



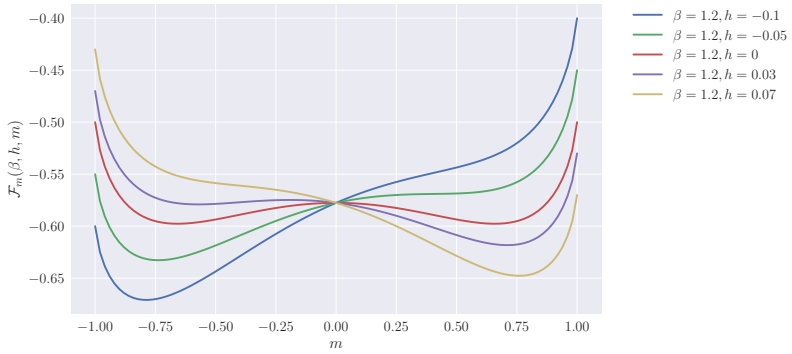
At high temperature ($\beta < 1$), the curve has only one minimum at zero. At low temperature ($\beta > 1$), the curve has two local minimums symmetric around zero. There is a phase transition at $\beta = 1$.

The plot of function $\mathcal{F}_m(\beta, h, m)$ under $\beta = 1.2$ and $h = -0.1, -0.05, 0, 0.03, 0.07$

[1] It may be useful to use one of the two following identities:

$$\operatorname{atanh}(x) = \frac{1}{2}\log\left(\frac{1+x}{1-x}\right) \text{ or}$$

$$H(m) = \log\left(2\cosh\left(\operatorname{atanh}(x)\right)\right) - x\operatorname{atanh}(x) = -\left[\frac{1+m}{2}\log\left(\frac{1+m}{2}\right) + \frac{1-m}{2}\log\left(\frac{1-m}{2}\right)\right]$$



Since $\beta > 1$, the minimum will drift away from zero. The number of local minimums can be one or two depending on $|h|$. If $h < 0$, the global minimum will be closer to -1, if $h > 0$, the global minimum will be closer to 1. When h moves pass zero, the global minimum changes sharply to another value with opposite sign.

(b) Since m^* is the minimizer of $\mathcal{F}_m(\beta, h, m)$ on interval $[-1, 1]$, we can take partial derivative of $\mathcal{F}_m(\beta, h, m)$ w.r.t. m and set it to zero

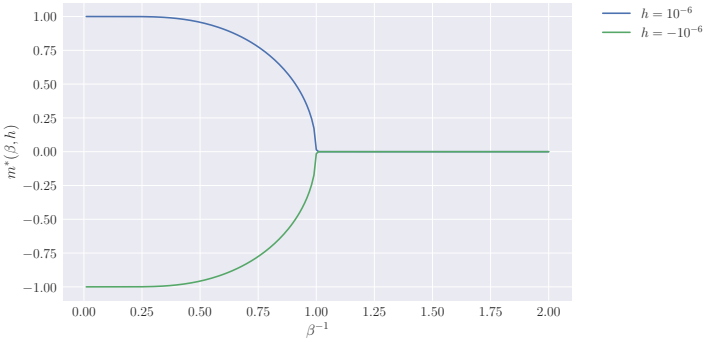
$$0 = \frac{\partial \mathcal{F}_m(\beta, h, m)}{\partial m}\bigg|_{m=m^*} = -m - h + \frac{1}{\beta}\frac{1}{2}\log\left(\frac{1+m}{1-m}\right)\bigg|_{m=m^*} = -m^* - h + \frac{1}{\beta}\operatorname{atanh}(m^*)\,.$$

By rearranging the equation yields

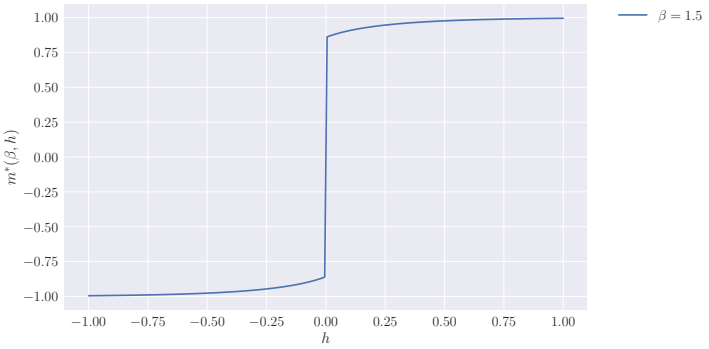
$$m^* = \tanh(\beta(m^* + h))\,,$$

which is exactly the self-consistent equation. Note that m^* is actually a function of inverse temperature β and magnetic field h.

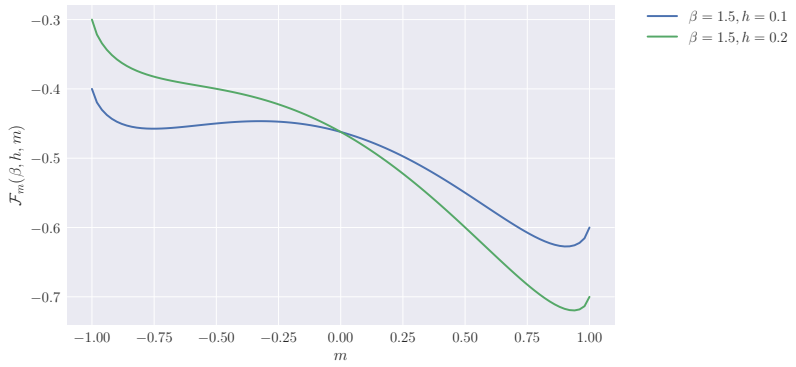
The plot of function $m^*(\beta, h)$ under $h = \pm 10^{-6}$ and $\beta^{-1} \in (0, 2]$



The plot of function $m^*(\beta, h)$ under $\beta = 1.5$ and $h \in [-1, 1]$



(c) The plot of function $\mathcal{F}_m(\beta, h, m)$ under $\beta = 1.5$ and $h = 0.1, 0.2$



For $h = 0.1$, there are two solutions, the right one has lower function value thus it is the correct one. For $h = 0.2$, there is only one solution, so it is the correct one.

□

Monte-Carlo-Markov-Chain

Consider again the Hamiltonian of the Curie-Weiss model. A very practical way to sample configurations of N spins from the Gibbs probability distribution

$$P\left(\{s_i\}_{i=1}^N ; \beta, h\right) = \frac{\exp\left(-\beta \mathcal{H}\left(\{s_i\}_{i=1}^N ; h\right)\right)}{Z_N(\beta, h)}$$

is the Monte-Carlo-Markov-Chain (MCMC) method, and in particular the Metropolis-Hastings algorithm. It works as follows:

1. Choose a starting configuration for the N spins values $s_i = \pm 1$ for $i = 1, \ldots, N$.
2. Choose a spin i at random. Compute the current value of the energy E_{now} and the value of the energy E_{flip} if the spins i is flipped (that is if $S_i^{\text{new}} = -S_i^{\text{old}}$).
3. Sample a number r uniformly in $[0, 1]$ and, if $r < \mathrm{e}^{\beta(E_{\text{now}} - E_{\text{flip}})}$ perform the flip (i.e. $S_i^{\text{new}} = -S_i^{\text{old}}$) otherwise leave it as it is.
4. Goto step 2.

If one is performing this program long enough, it is guarantied that the final configuration ($\{S\}$) will have been chosen with the correct probability.

(a) Write a code to perform the MCMC dynamics, and start by a configuration where all spins are equal to 1. Take $h = 0$, $\beta = 1.2$ and try your dynamics for a long enough time (say, with $t_{\max} = 100N$ attempts to flips spins) and monitor the value of the magnetization per spin $m = \sum_i s_i/N$ as a function of time. Make a plot for $N = 10, 50, 100, 200, 1000$ spins. Compare with the exact solution at $N = \infty$. Remarks? Conclusions?

(b) Start by a configuration where all spins are equal to 1 and take $h = -0.1$, $\beta = 1.2$. Monitor again the value of the magnetization per spin $m = \sum_i s_i/N$ as a function of time. Make a plot for $N = 10, 50, 100, 200, 1000$ spins. Compare with the exact solution at $N = \infty$. Remarks? Conclusions?

Solution. Before we write the code, let's first review the Curie-Weiss model. Since we may try different β, h, N, to avoid confusion it is better to write them explicitly. For any $k \in \{1, \dots, N\}$, we can split the Hamiltonian as

$$\begin{aligned}\mathcal{H}\left(\{s_i\}_{i=1}^N; h\right) &= -\frac{1}{N}\sum_{i<j} s_i s_j - h\sum_{i=1}^N s_i \\ &= \underbrace{-\frac{1}{N}\sum_{\substack{i>j \\ i\neq k, j\neq k}} s_i s_j - h\sum_{i\neq k} s_i}_{\text{do not contain } s_k} \underbrace{-\frac{1}{N}s_k\sum_{i\neq k} s_i - hs_k}_{\text{contains } s_k}\end{aligned}$$

Suppose we flip the k-th spin, then since the only difference is $S_k^{\text{new}} = -S_k^{\text{old}}$

$$\begin{aligned}E_{\text{now}} - E_{\text{flip}} &= \mathcal{H}\left(\{s_i^{\text{old}}\}_{i=1}^N; h\right) - \mathcal{H}\left(\{s_i^{\text{new}}\}_{i=1}^N; h\right) \\ &= \left[-\frac{1}{N}s_k^{\text{old}}\sum_{i\neq k} s_i^{\text{old}} - hs_k^{\text{old}}\right] - \left[-\frac{1}{N}s_k^{\text{new}}\sum_{i\neq k} s_i^{\text{new}} - hs_k^{\text{new}}\right] \\ &= \left[-\frac{1}{N}s_k^{\text{old}}\sum_{i\neq k} s_i^{\text{old}} - hs_k^{\text{old}}\right] - \left[\frac{1}{N}s_k^{\text{old}}\sum_{i\neq k} s_i^{\text{old}} + hs_k^{\text{old}}\right] \\ &= -2s_k^{\text{old}}\left[\frac{1}{N}\sum_{i\neq k} s_i^{\text{old}} + h\right]\end{aligned}$$

Secondly, when $N \to \infty$, the Boltzmann distribution is dominated by configurations whose magnetization equals m^*. So if we can run a simulation with infinite N, the trace plot will start with a warm-up stage to reach m^*, once m^* is reached, the trace plot will get stuck at there.

(a) Since $h = 0$, there the two local minimizers have same function value so either one can achieve the global minimum. However, we start our MCMC chain at all-one configuration, which is closer to the positive minimizer. It is possible that the random walk will reach the negative minimizer. Actually in the long run, the chain will first walk to the positive minimizer and stay for a while, then walk to the negative minimizer and stay for a while, and do this back and forth. The larger N is, the longer the chain stay at the minimizers, and the less fluctuation the chain will deviated from the minimizer.

(b) In this case we have similar behavior as part (a), the only difference is $h = -0.1$ breaks down the balance between two local minimizers. As discussed in Problem 2 (c), the negative minimizer is the correct one according to the Laplace method. So the trace plot will start with a warm-up stage to reach m^*, once m^* is reached, the trace plot will get stuck at there. The larger N is, the less fluctuation the chain will deviated from m^*.

□

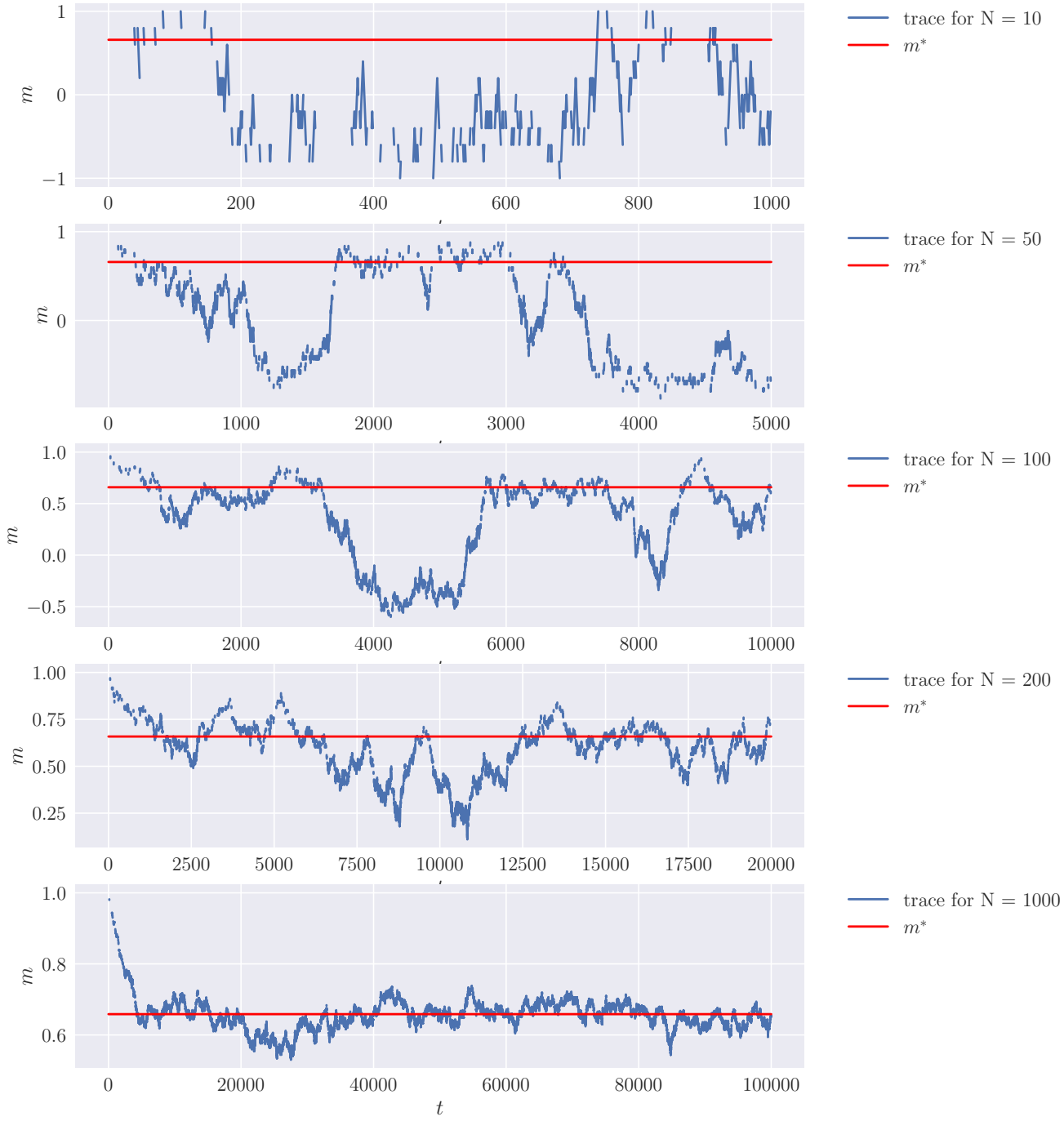


Figure 1: Trace plot for $\beta = 1.2, h = 0$ and $N = 10, 50, 100, 200, 1000$

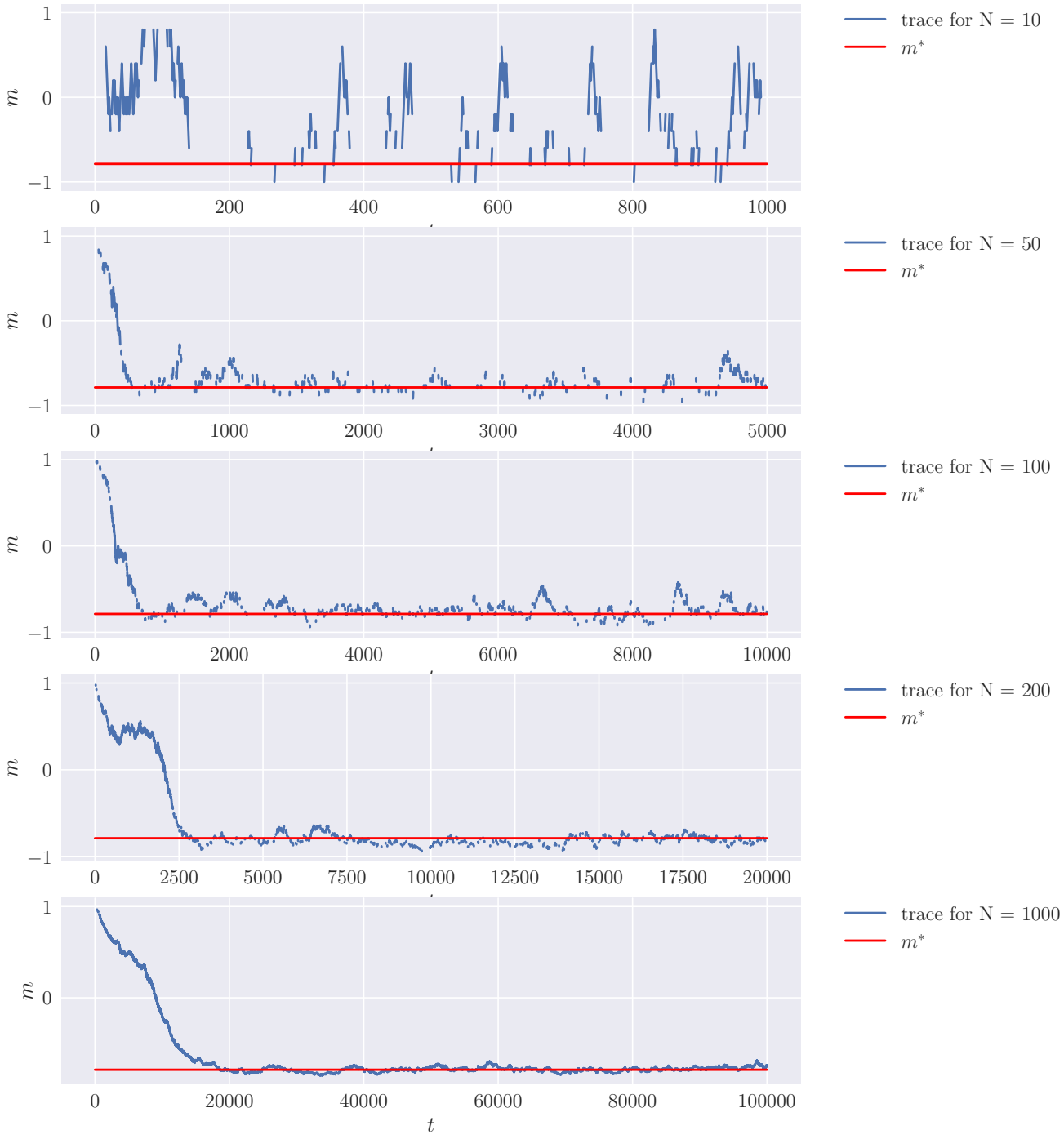


Figure 2: Trace plot for $\beta = 1.2, h = -0.1$ and $N = 10, 50, 100, 200, 1000$