Exactly solution of statistical mechanical models via Dynamic Programming

February 6, 2024

Dynamic Programming is a set of techniques for solving some class of mathematical problems in a computationally efficient way. This is typically achieved through a recursive strategy: the solution to a problem is given in the form of an algorithm, and it is found by solving smaller sub-problems similar in aspect to the original one. Here we see two examples where these techniques can be exploited to compute observables of (finite-size) statistical mechanical models. There will be no fancy math involved, just a wise use of the good old summations, multiplications, max/min, and sometimes derivatives.

As a measure of the running time of an algorithm we will be using the big O notation.

Problem 1. Chain-factorized probability distributions Given n discrete variables $x_1, x_2, \dots, x_n$ each taking values in $\mathcal{X} = \{1, 2, \dots, q\}$ and a function $f : \mathcal{X}^n \to \mathbb{R}$

$$f(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n-1} f_i(x_i, x_{i+1})$$

devise an algorithm running in polynomial time in n, q to compute:

1. $Z = \sum_{x_1, x_2, \dots, x_n} f(x_1, x_2, \dots, x_n)$
2. $p_i(\tilde{x}_i) = \sum_{x_1, x_2, \dots, x_n} \frac{1}{Z} f(x_1, x_2, \dots, x_n) \delta_{x_i, \tilde{x}_i}$
3. $\max_{x_1, x_2, \dots, x_n} f(x_1, x_2, \dots, x_n)$

Bonus:

1. Can you compute all marginals $p_i(\tilde{x}_i)$ $\forall i$ in time $\mathcal{O}(nq^2)$?
2. Find a polynomial strategy to draw samples from $p(x_1, x_2, \dots, x_n) = \frac{1}{Z} f(x_1, x_2, \dots, x_n)$ (Hint: any multivariate distribution can be decomposed as the product of conditional distributions via repeated application of Bayes' formula, then...)
3. Repeat all of the above for a system with periodic boundary conditions: $f(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} f_i(x_i, x_{i+1})$, with $x_{n+1} \equiv x_1$

Observation A one-dimensional Ising model is described by a distribution over binary spins $\sigma_1, \sigma_{2,} \dots, \sigma_n$, $\sigma_i \in \{1, -1\}$

$$p(\sigma_1, \sigma_{2,} \dots, \sigma_n) = \frac{1}{Z} \prod_{i=1}^{n} f_i(\sigma_i, \sigma_{i+1})$$

with $f_i(\sigma_i, \sigma_{i+1}) = e^{J\sigma_i\sigma_{i+1} + h\frac{\sigma_i + \sigma_{i+1}}{2}}$ for some $J, h \in \mathbb{R}$. In case you have met the transfer matrix method for solving the one-dimensional Ising model, convince yourself that it is equivalent to the strategy we see here.

Problem 2. Curie-Weiss model Given n binary variables $\sigma_1, \sigma_{2,} \dots, \sigma_n$ each taking values in $\mathcal{X} = \{1, -1\}$ and the probability distribution $p : \mathcal{X}^n \to \mathbb{R}$

$$p(\sigma_1, \sigma_{2,} \dots, \sigma_n) = \frac{1}{Z} e^{-\beta H(\sigma_1, \sigma_2, \dots, \sigma_n)}, \quad -H(\sigma_1, \sigma_{2,} \dots, \sigma_n) = \frac{J}{n} \left(\sum_{i=1}^{n} \sigma_i \right)^2 + \sum_{i=1}^{n} h_i \sigma_i$$

with $J, \beta, h_1, h_2, \dots, h_n \in \mathbb{R}$, devise an algorithm running in polynomial time in n to compute:

1. The partition function $Z = \sum_{\sigma_1, \sigma_2, \dots, \sigma_n} e^{-\beta H(\sigma_1, \sigma_2, \dots, \sigma_n)}$
2. Marginals $p_i(\tilde{\sigma}_i) = \sum_{\sigma_1, \sigma_2, \dots, \sigma_n} p(\sigma_1, \sigma_{2,} \dots, \sigma_n) \delta_{\sigma_i, \tilde{\sigma}_i}$
3. The average energy $\langle H \rangle = \sum_{\sigma_1, \sigma_2, \dots, \sigma_n} p(\sigma_1, \sigma_{2,} \dots, \sigma_n) H(\sigma_1, \sigma_{2,} \dots, \sigma_n)$ (Hint: use linearity of expectation values)
4. Can you do 3., but faster than $\mathcal{O}(n^3)$? (Hint: $\langle H \rangle$ is related to the derivative of a quantity which you can compute in $\mathcal{O}(n^2)$)

Bonus

1. The entropy $k = -\langle \log p \rangle = -\sum_{\sigma_1, \sigma_2, \dots, \sigma_n} p(\sigma_1, \sigma_{2,} \dots, \sigma_n) \log p(\sigma_1, \sigma_{2,} \dots, \sigma_n)$
2. A polynomial strategy do draw samples from p
3. Do the same for a Potts model: $\sigma_i \in \mathcal{X} = \{1, 2, \dots, q\}$ with energy

$$-H(\sigma_1, \sigma_{2,} \dots, \sigma_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} f(\sigma_i, \sigma_j) + \sum_{i=1}^{n} h_i(\sigma_i) \tag{1}$$

with $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $h_1, h_2, \dots, h_n : \mathcal{X} \to \mathbb{R}$. The procedure must be polynomial in n, q. (Hint: observe that the first term only depends on how many variables are in each of the q states)

Acknowledgement These exercises are heavily inspired by Alfredo Braunstein's notes for the course "Algorithms for optimization, inference, and learning".