MCV172, HW#2

Oren Freifeld

April 28, 2017

Dynamic Programming

The goal of this HW assignment is to build an exact sampler for the Ising model, on an 8×8 2D regular lattice, using dynamic programming. At each of three temperatures, ten random samples are displayed and two empirical expectations are computed.

The Ising model, at temperature Temp, is

$$p(x) = \frac{1}{Z} \exp\left(\frac{1}{\text{Temp}} \sum_{s \sim t} x_s x_t \right) \qquad \text{Temp} > 0 \tag{1}$$

$$Z = \sum_x \exp\left(\frac{1}{\text{Temp}} \sum_{s \sim t} x_s x_t \right) \tag{2}$$

where

$$S = \{(i,j)\}_{1 \le i \le 8, 1 \le j \le 8} \tag{3}$$

(the 8×8 lattice), $x_s \in \{-1, 1\}$, and $s \sim t$ means s and t form a pair of neighboring sites in S, as defined by a rectangular nearest-neighbor system (so that each interior site has four neighbors, to its North, South, East, and West).

Remark 1 *To be clear, in the summation above each unordered pair of sites appears only once (*i.e.*, no double counting). For example, in a 2×2 lattice, where the sites are given by $S = \{(1,1), (2,1), (2,1), (2,2)\}$, this would mean 4 (as opposed to 8) summands. For example, if* Temp $= 1$*, we get:*

$$p(x) = \frac{1}{Z} \exp\left(x_{(1,1)} x_{(1,2)} + x_{(1,1)} x_{(2,1)} + x_{(2,1)} x_{(2,2)} + x_{(1,2)} x_{(2,2)} \right) \tag{4}$$

$$\begin{aligned} Z &= \sum_x \exp\left(x_{(1,1)} x_{(1,2)} + x_{(1,1)} x_{(2,1)} + x_{(2,1)} x_{(2,2)} + x_{(1,2)} x_{(2,2)} \right) \\ &= \sum_{x_{(1,1)} \in \mathcal{R}} \sum_{x_{(1,2)} \in \mathcal{R}} \sum_{x_{(2,1)} \in \mathcal{R}} \sum_{x_{(2,2)} \in \mathcal{R}} \exp\left(x_{(1,1)} x_{(1,2)} + x_{(1,1)} x_{(2,1)} + x_{(2,1)} x_{(2,2)} + x_{(1,2)} x_{(2,2)} \right) . \end{aligned} \tag{5}$$

where $\mathcal{R} = \{-1, 1\}$. $\Diamond$

Remark 2 ***Before you start coding*** *a single line in this assignment, do yourself a favor and* ***read the entire document, including the programming notes at the end.*** ◊

Computer Exercise 1 *In the case of a* 2×2 *lattice and* $\mathrm{Temp} = 1$*, compute* Z *using brute force (you can do it using two loops, one nested inside the other). In effect:*

$$p(x) = \frac{1}{Z} \exp\left(\sum_{s \sim t} x_s x_t\right) \tag{6}$$

$$Z = \sum_x \exp\left(\sum_{s \sim t} x_s x_t\right) \tag{7}$$

where

$$S = \{(i,j)\}_{1 \le i \le 2, 1 \le j \le 2} \, . \tag{8}$$

Your result should be $Z = 121.23\ldots$. ◊

Computer Exercise 2 *In the case of a* 3×3 *lattice and* $\mathrm{Temp} = 1$*, compute* Z *using brute force (you can do it using three loops, one nested inside the other). In effect:*

$$p(x) = \frac{1}{Z} \exp\left(\sum_{s \sim t} x_s x_t\right) \tag{9}$$

$$Z = \sum_x \exp\left(\sum_{s \sim t} x_s x_t\right) \tag{10}$$

where

$$S = \{(i,j)\}_{1 \le i \le 3, 1 \le j \le 3} \, . \tag{11}$$

Your result should be $Z = 10^5 \cdot 3.65\ldots$. ◊

Computer Exercise 3 *Using dynamic programming and an* 8×8 *lattice, build an exact sampler for each of the three temperatures,* $\mathrm{Temp} \in \{1, 1.5, 2\}$. *Use the samplers to generate ten independent samples (to be clear, each such sample is an entire* 8×8 *image) at each of the three temperatures, and display these as thirty images all on a single plot (three rows, one for each temperature, and ten columns of* 8×8 *black-and-white images; the subplot command should be useful here).* ◊

Computer Exercise 4 *Using the three samplers you implemented above, at each of the three temperatures, draw 10,000 samples,* $x(1), \ldots, x(10000)$ *(each such sample is an* 8×8 *binary image) and compute two empirical expectations:*

$$\widehat{E}_{\mathrm{Temp}}(X_{(1,1)} X_{(2,2)}) \triangleq \frac{1}{10000} \sum_{n=1}^{10000} x_{(1,1)}(n) x_{(2,2)}(n) \tag{12}$$

$$\widehat{E}_{\mathrm{Temp}}(X_{(1,1)} X_{(8,8)}) \triangleq \frac{1}{10000} \sum_{n=1}^{10000} x_{(1,1)}(n) x_{(8,8)}(n) \tag{13}$$

where $\mathrm{Temp} = 1, 1.5,$ *and* 2 *and where* $x_{(i,j)}(n)$ *is the value at the* (i,j)*-th lattice site of sample* n. ◊

Exercise 1 *Using the results of the Computer Exercise above, explain the relative values of $\widehat{E}$, in terms of the spatial distance of the lattice sites and in terms of the temperature.* $\diamond$

Programming Notes

1. Sampling is fast; computing the conditional probabilities is slow. So you might want to save the conditional probabilities after they are computed (or, at least save the "T functions" – see below).

2. Debug your dynamic-programming implementation on 2×2 and 3×3 lattices, at temperature Temp $= 1$, by computing Z, the normalizing constant. You should get $Z = 121.23\ldots$ and $Z = 10^5 \cdot 3.65\ldots$, respectively.

3. The most computationally-efficient approach is to use a raster or boustrophedonic[1] site-visitation schedule and then compute conditional probabilities, site-by-site, in backward order. The number of computations is $O(64 \cdot 2^9) = O(2^{15})$. However, you are requested to do something simpler (and suboptimal): the programming burden becomes substantially easier if the 8×8 lattice is represented as a Markov chain with 8 sites, one for each row, and 8 corresponding variables, $y_1, \ldots, y_8$. Although computational complexity becomes $O(8 \cdot 2^{16}) = O(2^{19})$, the programming complexity is vastly reduced – so we will go with that. Each site variable, y_s (where $s = 1, \ldots, 8$) has 2^8 states, $y_s \in \{0, 1, \ldots, 255\}$, corresponding to the 2^8 possible configurations of row s in the Ising lattice. The correspondence is thus $y_s \leftrightarrow (x_{(s,1)}, \ldots, x_{(s,8)})$. A convenient mapping between y_s and $(x_{(s,i)})_{i=1}^8$ is to use the 8-bit binary representation of y_s, together with the conversion of 0's to -1's.

   Example 1 *Suppose $y_s = 136$. The 8-bit binary representation of 136 is '10001000'. Therefore, the corresponding row (*i.e., $(x_{(s,i)})_{i=1}^8$*) is [1 -1 -1 -1 1 -1 -1 -1].* $\diamond$

   The following helper function, whose usage is optional, computes the $y_s \to (x_{(s,i)})_{i=1}^8$ mapping.

```
import numpy as np
def y2row(y):
    """
    y: an integer in (0,1,...,255)
    """
    if not 0<=y<=255:
        raise ValueError(y,"y must be an integer in {0,...,255}")
    my_str=np.binary_repr(y,8)
    my_list = map(int,my_str)
    my_array = np.asarray(my_list)
    my_array[my_array==0]=-1
    row=my_array
    return row
```

[1] Namely, from right to left and from left to right in alternate lines. The word came from Greek and refers to how an ox turns when plowing a field.

Remark 3 *Note that in this HW assignment, the y's merely stand for groups of x's, used as a mechanism for simplifying the programming. These y's should not be confused with "observations". In this assignment we just use the prior, $p(x)$, and there are no noisy observations. However, letting z denote observations, the sampling mechanism above can be easily adjusted to sampling from, the posterior, $p(x|z)$, in the case that each data point is connected in the graph to a single site in the original graph of $p(x)$;* i.e.*, if the likelihood, $p(z|x)$, factorizes as $p(z|x) = \prod_{s \in S} p(z_s|x_s)$. As we saw in class, the only thing that will have to change is the functional form of the clique functions.* $\diamondsuit$

4. In this representation, there are two kinds of clique functions, singletons

$$G = G(y_s) = \exp\left(\frac{1}{\text{Temp}} \sum_{i=1}^{7} x_{(s,i)} x_{(s,i+1)}\right), \qquad s = 1, 2, \ldots, 8 \tag{14}$$

representing, for each of the eight rows, s, the product of the seven intra-row pair cliques from the original graph, and pair cliques

$$F = F(y_s, y_{s+1}) = \exp\left(\frac{1}{\text{Temp}} \sum_{i=1}^{8} x_{(s,i)} x_{(s+1,i)}\right) \qquad s = 1, 2, \ldots, 7 \tag{15}$$

representing, for each of the seven pairs of rows, $(s, s+1)$, the product of the eight inter-row pair cliques from the original graph. Thus,

$$p(y_1, \ldots, y_8) \propto \left(\prod_{s=1}^{8} G(y_s)\right)\left(\prod_{s=1}^{7} F(y_s, y_{s+1})\right) \tag{16}$$

The dynamic programming iterations, adopting the site-visitation schedule $1, 2, \ldots, 8$, is then

$$T_1(y_2) = \sum_{y_1=0}^{255} G(y_1) F(y_1, y_2) \qquad y_2 \in \{0, 1, \ldots, 255\} \tag{17}$$

$$T_k(y_{k+1}) = \sum_{y_k=0}^{255} T_{k-1}(y_k) G(y_k) F(y_k, y_{k+1}) \qquad 2 \le k \le 7 \quad y_{k+1} \in \{0, 1, \ldots, 255\} \tag{18}$$

$$\mathbb{R} \ni T_8 = \sum_{y_8=0}^{255} T_7(y_8) G(y_8) \quad (= Z) \tag{19}$$

Working backwards:

$$p_8(y_8) = \frac{T_7(y_8) G(y_8)}{Z} \qquad y_8 \in \{0, 1, \ldots, 255\} \tag{20}$$

$$p_{k|k+1}(y_k|y_{k+1}) = \frac{T_{k-1}(y_k) G(y_k) F(y_k, y_{k+1})}{T_k(y_{k+1})} \qquad k = 7, 6, \ldots, 2 \quad y_k, y_{k+1} \in \{0, 1, \ldots, 255\} \tag{21}$$

$$p_{1|2}(y_1|y_2) = \frac{G(y_1) F(y_1, y_2)}{T_1(y_2)} \tag{22}$$

Note that, on a computer, we store p_8 as a 1D array of length $2^8 = 256$, and $p_{k|k+1}$ as a $2^8 \times 2^8 = 256 \times 256$ array.

Remark 4 *As usual, when writing p, we could have also absorbed singletons into the pairwise functions,* e.g.*:*

$$\begin{aligned} p(y_1, \ldots, y_8) \propto & \underbrace{G(y_1)F(y_1, y_2)}_{H_{1,2}(y_1,y_2)} \underbrace{G(y_2)F(y_2, y_3)}_{H_{2,3}(y_2,y_3)} \underbrace{G(y_3)F(y_3, y_4)}_{H_{3,4}(y_3,y_4)} \\ & \times \underbrace{G(y_4)F(y_4, y_5)}_{H_{4,5}(y_4,y_5)} \underbrace{G(y_5)F(y_5, y_6)}_{H_{5,6}(y_5,y_6)} \underbrace{G(y_6)F(y_6, y_7)}_{H_{6,7}(y_6,y_7)} \\ & \times \underbrace{G(y_7)G(y_8)F(y_7, y_8)}_{H_{7,8}(y_7,y_8)} = \prod_{s=1}^{7} H_{s,s+1}(y_s, y_{s+1}) \end{aligned} \tag{23}$$

$\diamondsuit$