

Gibbs Sampling and Online Learning Introduction

Lecturer: Drew Bagnell

Scribes: {Shichao Yang}¹

1 Sampling

Samples are valuable because they can be used to compute expected values of functions over a probability distribution. They can also be used to compute marginals by counting. Most sampling techniques use the ratios taught in last class to minimize computation time. This section discusses two techniques: importance sampling and Gibbs' sampling. The latter (or more commonly some variant) is the more frequently used method.

1.1 Importance Sampling

This technique has been discussed earlier in class: We want $\mathbb{E}_p f(\mathbf{x})$, but we can't sample p , so we instead sample \mathbf{x}_i 's from a different distribution q and weight each $f(\mathbf{x}_i)$ by $\frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)}$. As far as possible, q should be similar to p in order to make the sampling more precise.

What distribution should we draw from? We can draw from $q = \prod q_i(\mathbf{x}_i)$, but this usually does not work since in most cases, it is too far from the actual distribution p . It is easier to sample from chains and trees, so often if a non-tree graph is almost a tree, we can convert it to a tree by breaking some links, sample from the new tree-based distribution, and use importance sampling to compute the expectations for the non-tree graph. Importance sampling usually only works for graphs that are nearly trees: in most examples that are of interest, this is never the case. We then need the following Gibbs sampling.

1.2 Gibbs Sampling

Consider the Gibbs field in Figure ???. Gibbs' sampling works as follows:

1. Start with some initial configuration (say $[0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1]$ in the order x_1 to x_{15}).
2. Pick a node at random (say x_7).
3. Resample the node's probability conditioned on its Markov blanket: generate $x_7^+ \sim p(x_7 | N(x_7))$, i.e. flip a coin in proportion to the likelihood of getting a 0 or a 1 given the neighbors.
4. Repeat steps 3 and 4 for T time steps.

For a large enough T , \mathbf{x}_t converges to a sample from $p(\mathbf{x})$. To get the next sample, it is necessary to wait another T time steps, as updates too close to each other may be correlated. If you are

¹Contact adapted from previous scribes: Jiayi Zhou in F12, Sankalp Arora in F11.

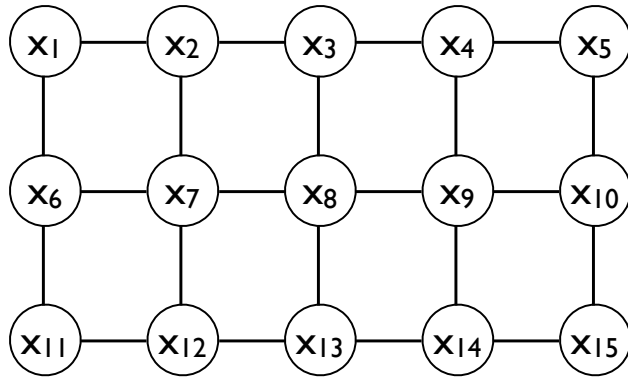


Figure 1: Gibbs' field with maximum clique size 2.

computing the expectation of a function under p , it is more advantageous to do that directly over the updates, than to wait for samples: for large enough T , $\frac{1}{T} \sum_t f(\mathbf{x}_t)$ converges to $\mathbb{E}_p f(\mathbf{x})$. Note that in generally x does not converge, it is the expectation that converges. There might be some other tricks. For example, it is better not to include x_t in beginning stages to compute $\frac{1}{T} \sum_t f(\mathbf{x}_t)$ as it is more randomly initially. We may also have many sequences of x_t running parallel and finally compute their mean.

1.3 Doubts and Answers

You may have the following doubts as I doubted before:

- How could the expectation converge? Think of the problem in 2D case ($x = [x_0, x_1]$). Pick out all those steps that we perform resampling on x_0 while fixing x_1 , look at all these x_0 and you will find out they will converge to as if sampled from the marginal probability $p(x_0)$, and so as x_1 .
- It is not iid sampled, why does it work? For adjacent steps it is not iid sampled, but if you look at every K steps (generally K cannot be too small), it is approximately iid sampled.

1.4 Derivation

To deeply understand why gibbs sampling could converge to the desired distribution, here we take a simple example. Think of robot in a chess board in Figure ???. Robot starts from one random cell and then could jump to its neighbours in four directions but will bounce back when reaching boundary. After running it for long steps, what's the final distribution of the robot place in the board? Will there be an equilibrium distribution or multiple? If multiple, will it change periodically?

For the equilibrium state, it means that belief of each cell $bel(x_i)$ won't change. In more detail, the probability of robot in its surroundings $N(x_i)$ and move to x_i is the same as robot in x_i and move to its surroundings $N(x_i)$. It is similar to the stable fluid. Fluid coming to this cell is the same as fluid going out.

It is easy to design the gibbs field similar as the chess board. If x represents the state of all cells at specific time, $x = x_1, x_2, \dots, x_i, \dots, x_{100}$, while \bar{x} represents another state: $\bar{x} = x_1, x_2, \dots, \bar{x}_i, \dots, x_{100}$. We want to know whether: $p(x)p(\bar{x}|x)$ equals $p(\bar{x})p(x|\bar{x})$ if we draw samples from the field using gibbs sampling. Actually,

$$\begin{aligned}
 p(x)p(\bar{x}|x) &= p(x) \frac{p(x_1, \dots, x_i = 1, \dots, x_n)}{\sum_{x_i=\pm 1} p(x_1, \dots, x_i, \dots, x_n)} \\
 &= \frac{p(x)p(\bar{x})}{\sum_{x_i=\pm 1} p(x_1, \dots, x_i, \dots, x_n)} \\
 p(\bar{x})p(x|\bar{x}) &= p(\bar{x}) \frac{p(x_1, \dots, x_i = 0, \dots, x_n)}{\sum_{x_i=\pm 1} p(x_1, \dots, x_i, \dots, x_n)} \\
 &= \frac{p(\bar{x})p(x)}{\sum_{x_i=\pm 1} p(x_1, \dots, x_i, \dots, x_n)}
 \end{aligned}$$

So the two expressions are the same value. This means $p(x)$ is actually the equilibrium state.

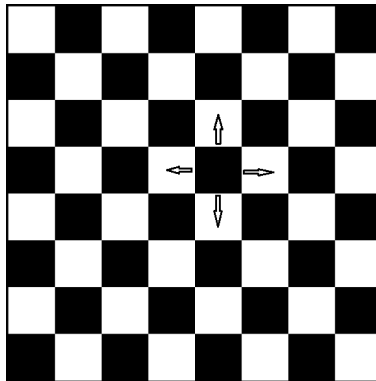


Figure 2: Chess board problem. Robot starts at one cell, and could move in four directions but will bounce back when reaching the boundary. We want to find the final distribution of each cell after long enough steps.

2 Online Learning Introduction

This section begins with a small discussion on logic and normative view of the world. Then we will question the nature of induction and its use as a tool for modeling and making decisions about the world. We will explore the limits of the inductive process, and in the process, prove how well different online induction algorithms (*e.g.* - weighted majority, randomized weighted majority etc.) perform in relation to each other through the new concepts of *loss* and *regret*.

2.1 Normative View of the World

Normative view is concerned with identifying the best decision to take or inference to make, assuming an ideal decision maker who is fully informed, is able to compute with perfect accuracy and is

fully rational. It assumes probability based inference and general decision making can and should only be defined by pure logic. The normative view suggests that any decision maker or inference engine that doesn't follow the aforementioned rules is wrong or invalid. Hence it does not state the amount of error induced in a system if approximation techniques (*like finite number of particles in a particle filter*) are used for the lack of adequate computation resources, time or knowledge. Succinctly put the normative view considers “known unknowns”, not the “unknown unknowns”. And this fallacy makes the normative view of little use in real life.

2.2 Induction

Our standard justification for the use of probability theory is that it seems like the “reasonable” thing to do. Logic is certainly reasonable, and probability naturally and uniquely extends our notion of logic to include degree-of-belief type statements. Thus, if one accepts logic, one must accept probability as a way of reasoning about the world.

In all the probabilistic systems discussed so far (*e.g.* localization and mapping), we can momentarily discard our normative view by looking at their common elements. At the very core, something like mapping, for example, can be seen simply as an induction problem. We take in data about world, and using prior knowledge, we make inferences, and then predictions using the inferences. In fact, Bayes' rule can be seen as the “canonical” way to do these operations. Induction is a method to predict the future based on observations in the past, and to make decisions from limited information. Thus, we can look generally at induction itself in order to arrive at answers to certain fundamental questions behind many robotic learning systems. Questions such as: “*Why does inductive reasoning work?*”, “*When does it fail?*”, and “*How good can it be?*” will tell us about the nature of learning and lead us to unique notions of optimality with regard to worst-case adversarial conditions.

2.2.1 Induction from a Chicken's Perspective

We begin with an illustrative example of how inductive reasoning can fail. Imagine you are a chicken. Every day of your life, the farmer comes to you at a certain time in the day and feeds you. On the 1001th day, the farmer walks up to you. What do you predict? What can you predict other than that he will feed you? Unfortunately for you, this day the farmer has decided to eat you. Thus, induction fails spectacularly. You may object to this example and say that the model of the world that the chicken was using was wrong, but this is precisely the point of the discussion: perhaps we can *never* have a model that perfectly aligns with the real world. Perhaps, you know vaguely of your impending doom, so you assign a low probability of being eaten everyday. You still will (probably) guess incorrectly on that 1001th day because your guess of being fed will dominate your feeling of doom. Furthermore, for any model you bear, an adversarial or random farmer can always thwart it to force false inductive predictions about your fate each day.

2.2.2 Three views of induction:

1. *The David Hume view.* No matter what, your model will never capture enough of the world to be correct all the time. Furthermore, induction relies heavily on priors made from assumptions, and potentially small changes in a prior can completely change a future belief. If your model is as good as your assumptions or priors, and most assumptions are not likely to be

valid at all, then induction is fundamentally broken. We find this view, although true, is not very satisfying or particularly useful.

2. *The Goldilocks/Panglossian/Einstein Position.* It just so happens that our world is set up in a way such that induction works. It is the nature of the universe that events can be effectively modeled through induction to a degree that allows us to make consistently useful predictions about it. In other words, we're lucky. Once again, we do not find this view very satisfying or particularly useful.
3. *The No Regret view.* It is true that you cannot say that induction will work or work well for any given situation, but as far as strategies go, it's provably near-optimal. This view comes from the outgrowth of game theory, computer science, and machine learning. This thinking is uniquely late 20th century.