Course on probabilistic graphical models
Master MVA 2015-2016
Review exercises- Part III

December 15, 2015

Chromatic Gibbs sampler

It is interesting to consider variants of Gibbs sampling in which several variables are resampled at the same time and that could be executed in parallel. A natural naive attempt is to try at time t to resample in parallel each variable X_i from a Bernoulli distribution with probability $\sigma\big(\eta_i + \sum_{j \in N_i} \eta_{ij} x_j^{(t-1)}\big)$, where σ is the logistic function.

- Find a counterexample with two nodes that shows that this parallel update does not converge to the Gibbs distribution.

Understanding this intuitively is relatively simple, but doing the actual calculations is a bit tedious.
Consider the following joint distribution on a graph with two nodes corresponding to the pair of variables (X, Y)

	0	*1*
0	$(1-\varepsilon)/2$	$\varepsilon/2$
1	$\varepsilon/2$	$(1-\varepsilon)/2$

For ε *small, this is a distribution that favors the configurations* $(0,0)$ *and* $(1,1)$*. For this model and if we apply Gibbs sampling, a simple calculations shows that* $\mathbb{P}(Y=1|X=1) = \mathbb{P}(Y=0|X=0) = 1-\varepsilon$*. and* $\mathbb{P}(Y=1|X=0) = \mathbb{P}(Y=1|X=0) = \varepsilon$*.. Now intuitively, if we happen to start with the configuration* $(0,1)$ *and we apply regular Gibbs sampling, we will with high probability move to the configuration* $(0,0)$ *or* $(1,1)$ *depending on which variable we update first. However, if we do updates in parallel there is a large probability that we will observe a sequence* $(0,1)$ *then* $(1,0)$ *then* $(0,1)$ *then* $(1,0)$ *then* $(0,1)$ *then* $(1,0)$*... If we take the limit when* ϵ *goes to zero it seems clear that the updates in parallel are not going to converge to the correct distribution.*

Now, if we want to make a rigorous proof (to be skipped in a first reading), we can for example consider the transition matrix from $\{(0,0),(1,0),(0,1),(1,1)\}$ *to itself induced by the "parallel Gibbs sampling". This matrix is*

$$\begin{bmatrix} (1-\varepsilon)^2 & \varepsilon(1-\varepsilon) & \varepsilon(1-\varepsilon) & \varepsilon^2 \\ \varepsilon(1-\varepsilon) & \varepsilon^2 & (1-\varepsilon)^2 & \varepsilon(1-\varepsilon) \\ \varepsilon(1-\varepsilon) & (1-\varepsilon)^2 & \varepsilon^2 & \varepsilon(1-\varepsilon) \\ \varepsilon^2 & \varepsilon(1-\varepsilon) & \varepsilon(1-\varepsilon) & (1-\varepsilon)^2 \end{bmatrix}$$

and if "parallel Gibbs sampling" was working its stationary distribution should be $\frac{1}{2}(1-\varepsilon,\varepsilon,\varepsilon,1-\varepsilon)^\top$ *and so it should be a a left eigenvector of the previous matrix. Computing any row of the product between this vector and this matrix shows that* $\frac{1}{2}(1-\varepsilon,\varepsilon,\varepsilon,1-\varepsilon)^\top$ *cannot be the stationary distribution unless* $\varepsilon = \frac{1}{2}$*. QED*

We will focus on the case where the graph is a two-dimensional grid of size $n \times m$ with each node not on the boundary connected to four neighbors. We partition the nodes according to a checkerboard pattern: let $A = \{(i,j) \in [n] \times [m] \mid i+j \text{ is even } \}$ and $B = ([n] \times [m]) \backslash A$ with the notation $[k] = \{1, \dots, k\}$. Consider a reduced graph composed of two nodes associated with the variables X_A and X_B, with $X_A = (X_{i,j})_{(i,j)\in A}$ and likewise for X_B. Consider, on this reduced graph, the standard Gibbs sampling scheme that samples X_A conditionally on X_B and then X_B conditionally on X_A.

- Characterize the conditional distribution of $X_A|X_B$ and show how to sample from $X_A|X_B$ easily. Propose an efficient algorithm to partially parallelize Gibbs sampling.

  Because of the configuration in checkerboard, B contains all the Markov blankets of all the elements in A. This implies that B separates any element i in A from $A\backslash\{i\}$. As a result

  $$\forall i \in A, \qquad X_i \perp\!\!\!\perp X_{A\backslash\{i\}} \mid X_B.$$

  Now this shows that

  $$p(x_A|x_B) = p(x_i|x_B)\, p(x_{A\backslash\{i\}}|x_B).$$

  But using the fact that B also separates all elements in $A\backslash\{i\}$, we have $X_j \perp\!\!\!\perp X_{A\backslash\{i,j\}} \mid X_B$ so that we have a full factorization:

  $$p(x_A|x_B) = \prod_{i\in A} p(x_i|x_B) = \prod_{i\in A} p(x_i|x_{N_i}),$$

  where N_i is the Markov blanket of i. This last equality is due to the fact that by construction for all $i \in A$, $N_i \subset B$ and to the fact that for any set C such that $N_i \subset C$ and $i \notin C$ we have $p(x_i|x_{N_i}) = p(x_i|x_C)$.

  But now, $p(x_i|x_{N_i})$ is exactly the conditional distribution that we usually sample from in the usual Gibbs sampling algorithm.

  So if we use the standard Gibbs sampling algorithm that samples X_A conditionally on X_B and then X_B conditionally on X_A, the obtained algorithm amounts to iteratively

  - *sample in parallel all variables X_i for $i \in A$ from the Gibbs transition given the values of their neighbors, so that all the $(X_i)_{i\in A}$ are resampled*
  - *then conditionally on the new values that these $(X_i)_{i\in A}$ now take, sample in parallel all variables X_j for $j \in B$ using their Gibbs transition.*

  Now, provided the Gibbs transition allow transitions to all possible values of X_i, it not difficult to see that after updating X_A then X_B there is a non-zero probability to have transitioned to any configuration of all variables, which means that a cycle "update X_A then X_B" corresponds to a regular transition. Since by construction, the Gibbs distribution is the stationary distribution of the chain, this proves that this algorithm creates a Markov chain which converges to the desired distribution. The algorithm is efficient because a number of updates are now done in parallel.

- How would you generalize this idea to a general graph?

  The key idea is that we were able to split the graph in two sets A and B such that all the neighbors of elements of A were in B and vice-versa. In fact what is important in the proof is that no element of A should have a neighbor in A itself. So if we paint the nodes of A in red, no red node should have a red neighbor...

  This leads to the classical coloring problem of the nodes of a graph: we can group together points that would share the same color, but no neighbors should have the same color. So the strategy is to first compute a coloring of the graph (`https://en.wikipedia.org/wiki/Graph_coloring`), which of course will require in general more than two sets. This coloring defines a collection of disjoint sets $A_1, \dots, A_K$ of different colors that partition the nodes of the graph. The idea now is to consider Gibbs sampling on the "macro" variables $X_{A_1}, \dots, X_{A_K}$. Using the very same argument as in the previous question, this will reduce to apply Gibbs updates in parallel to all the node of a given set A_j.