Boosting association rule mining in large datasets via Gibbs sampling

Guoqi Qian[a], Calyampudi Radhakrishna Rao[b,c,1], Xiaoying Sun[d], and Yuehua Wu[d]

[a]School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia; [b]Department of Biostatistics, University at Buffalo, The State University of New York, Buffalo, NY 14221-3000; [c]CRRAO Advanced Institute of Mathematics, Statistics and Computer Science, Hyderabad-500046, India; and [d]Department of Mathematics and Statistics, York University, Toronto, ON, Canada M3J1P3

Contributed by Calyampudi Radhakrishna Rao, March 19, 2016 (sent for review December 3, 2015; reviewed by Marepalli Rao and Talluri Jagannadha Rao)

Current algorithms for association rule mining from transaction data are mostly deterministic and enumerative. They can be computationally intractable even for mining a dataset containing just a few hundred transaction items, if no action is taken to constrain the search space. In this paper, we develop a Gibbs-sampling–induced stochastic search procedure to randomly sample association rules from the itemset space, and perform rule mining from the reduced transaction dataset generated by the sample. Also a general rule importance measure is proposed to direct the stochastic search so that, as a result of the randomly generated association rules constituting an ergodic Markov chain, the overall most important rules in the itemset space can be uncovered from the reduced dataset with probability 1 in the limit. In the simulation study and a real genomic data example, we show how to boost association rule mining by an integrated use of the stochastic search and the Apriori algorithm.

association rule | Gibbs sampling | transaction data | genomic data

Association rule mining (1, 2) in many research areas such as marketing, politics, and bioinformatics is an important task. One of its well-known applications is the market basket analysis. An example of association rule from the basket data might be that "90% of all customers who buy bread and butter also buy milk" (1), providing important information for the supermarket's management of stocking and shelving. Instead of mining all association rules from a database, an interesting and useful task is to discover the most important association rules for a given consequent. For instance, a store manager of Walmart might be interested in knowing which items most of the customers purchased given that they got automotive services done in the store. For a genomic dataset, one might be interested in finding which SNP (single nucleotide polymorphism at certain loci in a gene) variables and their values imply a certain disease with the highest probability. The focus of this paper is to identify the most important association rules in a transaction dataset.

Let us formally define the problem of association rule mining using the notations of ref. 3. Define $I=\{I_1,I_2,\ldots,I_m\}$ as a set of m items called the "item space" and $D=\{t_1,t_2,\ldots,t_n\}$ as a list of transactions, where each transaction in D is just a subset of items in I, i.e., $t_j\subset I$, $j=1,\ldots,n$. An association rule is defined as an implication of the form $X\Rightarrow Y$ where $X,Y\subset I$ and $X\cap Y=\emptyset$. The sets of items (for short, "itemsets") X and Y are called "antecedent" and "consequent" of the rule, respectively. The support of an itemset, X, $\mathrm{supp}(X)$ is defined as the proportion of transactions in D which contain X. The confidence of an association rule is defined as $\mathrm{conf}(X\Rightarrow Y)=[\mathrm{supp}(X\&Y)]/\mathrm{supp}(X)$, where $X\&Y$ is the itemset obtained by amalgamating X with Y. The support of an itemset measures its commonness and the confidence of an association rule measures its association strength. By the essential meaning of support, we can also define the support for a rule $X\Rightarrow Y$, which is just $\mathrm{supp}(X\Rightarrow Y)\equiv\mathrm{supp}(Y\Rightarrow X)\equiv\mathrm{supp}(X\&Y)$.

Constraint-based search is mostly used in current algorithms to mine association rules. For instance, the Apriori algorithm (1) mines all rules satisfying a user-specified minimum support or minimum confidence, and maximum length. It is difficult to use such an algorithm in a dense dataset because it either searches through too many rules being computationally infeasible if the constraint is low, or misses the important ones otherwise. Some rule-mining algorithms use well-defined metrics to identify the most important association rules (4). But, they also use deterministic and exhaustive search, consequently becoming computationally intractable when applied to a dense dataset with, say, a few hundred items in the item space.

In this paper, we present a stochastic search algorithm to mine the most important, or optimal, association rules from a transaction dataset without information loss. The motivation comes from a genomic dataset of a disease and hundreds of SNP variables, and from the desire to mine the most important association rules for the disease outcome. Because the deterministic search algorithms are not able to cope with the computing intensity and immensity for this dataset, we have developed the stochastic algorithm to overcome the difficulty.

A New Algorithm Based on Gibbs Sampling

Motivation. Consider a dataset for supervised learning which contains observations of a response variable and a number of predictor variables from a sample of individual subjects. Such a dataset can be converted into a transaction dataset for association rule mining if both the response and the predictors are of categorical type. For example, datasets used in genome-wide association studies often consist of observations of categorical response and predictors on subjects. Here the response is a disease outcome having two categories, case (C) and noncase (NC), and each predictor is the so-called SNP variable having 3 categories corresponding to 0, 1, and 2 copy numbers of the minor allele at the loci. In this case, the response variable can be

Significance

Informative association rule mining is fundamental for knowledge discovery from transaction data, for which brute-force search algorithms, e.g., the well-known Apriori algorithm, were developed. However, operating these algorithms becomes computationally intractable in searching large rule space. The stochastic search algorithm developed here tackles this challenge by using the idea of annealing Gibbs sampling. Large rule space of exponential order can still be randomly searched by this algorithm to generate an ergodic Markov chain of viable length. The ergodic chain contains the most informative rules with probability 1, creating a much reduced rule space for subsequent mining without information loss. Such capacity of the algorithm is demonstrated using a carefully designed simulation study and a real genomic dataset containing 1,067 items.

Author contributions: G.Q., C.R.R., and Y.W. designed research; G.Q., C.R.R., X.S., and Y.W. performed research; G.Q., X.S., and Y.W. analyzed data; and G.Q., C.R.R., X.S., and Y.W. wrote the paper.

Reviewers: M.R., University of Cincinnati; and T.J.R., Indian Statistical Institute.

The authors declare no conflict of interest.

[1]To whom correspondence should be addressed. Email: crr1@psu.edu.

www.pnas.org/cgi/doi/10.1073/pnas.1604553113

Table 1. Association rules and their measurements

				Frequencies		
Rules	Supp.	Conf.	$g_{-}(\cdot)$	$\xi=3$	6	10
$I_1 \Rightarrow I_C$	0.47	0.890	0.420	0.242	0.382	0.595
$I_1, I_3 \Rightarrow I_C$	0.28	1.000	0.280	0.190	0.194	0.166
$I_3 \Rightarrow I_C$	0.33	0.650	0.210	0.171	0.155	0.095
$I_1, I_2 \Rightarrow I_C$	0.21	0.910	0.190	0.113	0.093	0.064
$I_1, I_2, I_3 \Rightarrow I_C$	0.11	1.000	0.110	0.101	0.063	0.021
$I_2 \Rightarrow I_C$	0.22	0.470	0.100	0.094	0.057	0.035
$I_2, I_3 \Rightarrow I_C$	0.12	0.570	0.070	0.089	0.056	0.024
$I_2 \Rightarrow I_{NC}$	0.25	0.53	0.13	0.182	0.240	0.345
$I_3 \Rightarrow I_{NC}$	0.18	0.35	0.06	0.154	0.173	0.161
$I_2, I_3 \Rightarrow I_{NC}$	0.09	0.43	0.04	0.149	0.146	0.136
$I_1 \Rightarrow I_{NC}$	0.06	0.11	0.007	0.125	0.115	0.091
$I_1, I_2 \Rightarrow I_{NC}$	0.02	0.09	0.002	0.138	0.101	0.097

"-" in g-(·) represents the association rule $\boldsymbol{J} \Rightarrow I_C$ (upper part) or $\boldsymbol{J} \Rightarrow I_{NC}$ (lower part).

represented by 2 response items, and each predictor variable can be represented by 3 predictor items.

To present the above discussion more explicitly, let n be the total number of transactions and k be the total number of predictor items. Denote the two response items as I_C and I_{NC}. Then the association rules of our interest have the antecedent being a subset of $\{I_1, \ldots, I_k\}$ and the consequent being either I_C or I_{NC}. These rules represent the associations between values of various predictor variables and the response variable which are different from those revealed by a supervising learning model such as the logistic or log-linear regression model.

An alternative representation of the transactions is binary vectors. Let $J_s = 1$ or 0 indicate the presence or absence of item s for $s = 1, \ldots, k, C, NC$. Denote $\boldsymbol{J} = (J_1, \ldots, J_k)$. The components of this binary vector are not necessarily independent of each other and the involved dependence provides a probabilistic interpretation to the associations among all of the items. Each transaction is an observation of the binary vector. Therefore, the collection of association rules of our interest can be divided into two families as $\mathcal{R}_C = \{\boldsymbol{J} \Rightarrow I_C,\ (J_1, \ldots, J_k) \in \{0,1\}^k\}$ and $\mathcal{R}_{NC} = \{\boldsymbol{J} \Rightarrow I_{NC},\ (J_1, \ldots, J_k) \in \{0,1\}^k\}$. Let $I = \{I_1, I_2, \ldots, I_k\}$. Denote the power set of I by 2^I that is the itemset space consisting of all possible itemsets of I. Given the consequent being either I_C or I_{NC}, one has to search through 2^I for all possible association rules. The following two properties clearly hold for this transaction dataset:

Property 1: $0 \le \mathrm{supp}(\boldsymbol{J} \Rightarrow I_{-}) \le \mathrm{conf}(\boldsymbol{J} \Rightarrow I_{-}) \le 1$, where I_{-} represents I_C or I_{NC}.

Property 2: Because $J_C + J_{NC} = 1$, $\mathrm{conf}(\boldsymbol{J} \Rightarrow I_C) + \mathrm{conf}(\boldsymbol{J} \Rightarrow I_{NC}) = 1$.

Our interest is to find association rules with high confidence. A constraint-based algorithm like the Apriori is computationally challenging when the item space is too large. It is even more difficult when the rules with high confidence have very low support. An example given in ref. 5 is that the forestry society FallAll conducted association rules mining to a dataset of 1,000 observations on marsh sides for providing advice on draining swamps to grow new forests. The Apriori algorithm was applied to this dataset by specifying the minimum support and confidence as 0.05 and 0.80, respectively. But, a strong association rule of confidence 1.0 and support 0.04 was missed with this set of constraints. In general, mining association rules in a dense dataset can miss important rules and get misinformed by noninformative rules produced due to improper constraints. This mishap motivates us to propose a random sampling and search procedure by defining a probability distribution for all possible association rules from the power set of I, i.e., 2^I, to find out those rules having high-level combined importance of confidence and support.

New Algorithm. The probability distribution for sampling and searching important association rules entails incorporating both support and confidence of the rules into the procedure. For this, we first define a new measure for association rules in $\mathcal{R}_C \cup \mathcal{R}_{NC}$ and call it the "importance," which is of the form $g(\boldsymbol{J} \Rightarrow I_{-}) = g_{-}(\boldsymbol{J}) = f(\mathrm{supp}(\boldsymbol{J} \Rightarrow I_{-}), \mathrm{conf}(\boldsymbol{J} \Rightarrow I_{-}))$, for a given association rule $\boldsymbol{J} \Rightarrow I_{-}$ with I_{-} being I_C or I_{NC}. Here f is a user-specified positive increasing function reflecting certain combined importance of the support and confidence of the rule. Plausible choices of f are the minimum, summation, or product of the support and confidence. Once f is specified, our aim becomes finding the most important association rules in $\mathcal{R}_C$ and $\mathcal{R}_{NC}$ which can be achieved by the following random-sampling-based search procedure.

We illustrate the idea of this procedure by focusing on rules in $\mathcal{R}_C$. The same applies for finding the most important rules in $\mathcal{R}_{NC}$. In light of the non-Bayesian optimization idea of ref. 6, we propose a probability distribution defined on $\mathcal{R}_C$ as

$$p_C(\boldsymbol{J}) = P(\boldsymbol{J} \Rightarrow I_C) = \frac{e^{\xi g(\boldsymbol{J} \Rightarrow I_C)}}{\sum_{\text{all } \boldsymbol{J}} e^{\xi g(\boldsymbol{J} \Rightarrow I_C)}}, \qquad [1]$$

where $\xi > 0$ is a tuning parameter. The most important rule in $\mathcal{R}_C$, denoted as $\boldsymbol{J}_{opt} \Rightarrow I_C$, is also the one maximizing $p_C(\boldsymbol{J})$ over $\mathcal{R}_C$, i.e., $\boldsymbol{J}_{opt} = \arg\max_{\boldsymbol{J}} p_C(\boldsymbol{J})$. This implies that $\boldsymbol{J}_{opt}$ can be found (with probability 1) from a random sample of $\boldsymbol{J}$ s generated from $p_C(\boldsymbol{J})$ if the sample size is sufficiently large. It can be proved that $\boldsymbol{J}_{opt}$ appears most frequently and has the largest value of $g(\boldsymbol{J} \Rightarrow I_C)$ in the sample with probability 1. However, generating a random sample from $p_C(\boldsymbol{J})$ is not trivial when k is not small, because the rule space $\mathcal{R}_C$ becomes huge and the normalizing denominator in $p_C(\boldsymbol{J})$ becomes intractable in evaluation. It turns out that the method of Gibbs sampling can be used to generate random samples from $p_C(\boldsymbol{J})$, where we need all conditional probability distributions of J_s given $\boldsymbol{J}_{-s}$:

$$\begin{aligned} p_C(J_s = 1 | \boldsymbol{J}_{-s}) &= \frac{p_C(J_s = 1, \boldsymbol{J}_{-s})}{p_C(\boldsymbol{J}_{-s})} \\ &= \frac{p_C(J_s = 1, \boldsymbol{J}_{-s})}{p_C(J_s = 1, \boldsymbol{J}_{-s}) + p_C(J_s = 0, \boldsymbol{J}_{-s})}, \\ p_C(J_s = 0 | \boldsymbol{J}_{-s}) &= 1 - p_C(J_s = 1 | \boldsymbol{J}_{-s}) \end{aligned}$$

for $s = 1, 2, \ldots, k$. Here $\boldsymbol{J}_{-s}$ is the subvector of $\boldsymbol{J}$ with J_s removed and $(J_s, \boldsymbol{J}_{-s})$ is the vector with J_s being put back into its original position in $\boldsymbol{J}$.

Then the Gibbs sampling algorithm for generating $\boldsymbol{J}$ s from $p_C(\boldsymbol{J})$ is given as the following:

i) Arbitrarily choose an initial vector $\boldsymbol{J}^{(0)} = (J_1^{(0)}, \ldots, J_k^{(0)})$;

ii) Repeating for $j = 1, 2, \ldots, M$, the antecedent $\boldsymbol{J}^{(j)}$ of the rule $(\boldsymbol{J}^{(j)} \Rightarrow I_C)$ is obtained by generating $J_s^{(j)}, s = 1, 2, \ldots, k$ sequentially from the Bernoulli distribution $p_C(J_s | J_1^{(j)}, \ldots, J_{s-1}^{(j)}, J_{s+1}^{(j-1)}, \ldots, J_k^{(j-1)})$;

iii) Return $(\boldsymbol{J}^{(1)}, \ldots, \boldsymbol{J}^{(M)})$ for the association rules sample $\{\boldsymbol{J}^{(j)} \Rightarrow I_C; j = 1, \ldots, M\}$.

Table 2. Items appearing in the random sample

T_1	Item	I_{390}	I_3	I_2	I_1
	Proportion	0.01	0.43	0.51	0.55
T_2	Item	I_{390}	I_3	I_2	I_1
	Proportion	0.01	0.43	0.51	0.55
T_3	Item	I_{390}	I_2	I_1	I_3
	Proportion	0.01	0.55	0.60	0.85

Table 3. Top 10 frequent items appearing in the rules identified by the Apriori algorithm for each dataset T_1, T_2, or T_3

T_1	Item	I_{44}	I_{292}	I_{135}	I_{97}	I_{286}	I_{184}	I_{187}	I_3	I_1	I_2
	Proportion	0.019	0.021	0.023	0.024	0.025	0.025	0.027	0.493	0.496	0.500
T_2	Item	I_{14}	I_7	I_4	I_{15}	I_8	I_6	I_{13}	I_3	I_1	I_2
	Proportion	0.087	0.090	0.091	0.093	0.105	0.130	0.136	0.496	0.499	0.500
T_3	Item	I_9	I_4	I_6	I_{10}	I_7	I_5	I_8	I_1	I_2	I_3
	Proportion	0.434	0.436	0.438	0.444	0.445	0.445	0.447	0.498	0.499	0.500

The generated sequence $\{\boldsymbol{J}^{(1)}, \cdots, \boldsymbol{J}^{(M)}\}$ is actually a Markov chain with its stationary distribution being $p_C(\boldsymbol{J})$ and it can be shown that the most frequent rule occurring in the generated sample converges to $\boldsymbol{J}_{opt}$ with probability 1 as $M \to \infty$. Moreover, those most important association rules in $\mathcal{R}_C$ are more likely to appear the most frequently in the generated sample than other less important ones, provided that the sample size M is sufficiently large. In the cases that the importance values of many important association rules are large but very close to each other, choosing a larger value for the tuning parameter ξ increases the probability ratio of every two rules, $[p_C(\boldsymbol{J}_1)]/p_C(\boldsymbol{J}_2) = e^{\xi(g(J_1 \Rightarrow I_C) - g(J_2 \Rightarrow I_C))}$, which helps differentiate the more important rules from the less important ones.

This is the framework of our random search procedure. We remark that the function g in [**1**] can be replaced by another interesting measure of association rules such as lift and leverage (4). Thus, a random sample can also be easily generated according to that interesting measure.

Once $\{\boldsymbol{J}^{(1)}, \cdots, \boldsymbol{J}^{(M)}\}$ is generated, the optimal association rules in $\mathcal{R}_C$, which have the highest importance, can be approximated by the association rules with the near-highest frequencies in the sample. The approximation precision can be achieved as high as one wants provided that the sample size is sufficiently large. Note that if the item space is very large, the generation of a long sample is computationally expensive. However, it is possible that in the random sample of a relatively small size M, the association rules could all be different from each other and each has the same frequency $1/M$. In this case, it is possible that none of the rules is optimal. Instead, we can compute the frequency for each item that ever appeared in the antecedents of the sampled rules. The frequency for item I_i is $\sum_{j=1}^{M} J_i^{(j)}/M$ for $i = 1, 2, \ldots, k$. We would obtain a subset of items that appear most frequently. Then we can apply the Apriori algorithm on the itemset space generated by the selected items to mine the optimal rules. Our simulation study shows that the random sample obtained by the Gibbs sampling method can largely reduce the itemset space for search and retain the most frequent predictor items from the optimal association rules simultaneously. In the next section we will elaborate how to use the generated sample of rules.

Simulation Study and Real Data Application

In this section, we present several numerical examples, first based on simulated data, and then on real data to demonstrate the performance of the random-sampling-based search procedure in different scenarios.

Simulation Studies. A transaction dataset containing strong association rules can be obtained by using the R package MultiOrd (7) to generate a list of binary vectors from a multivariate Bernoulli distribution of correlated binary random variables with a compatible pair of mean vector $\boldsymbol{p}$ and correlation matrix R (8). We start with a small dataset to show that our method is able to find the optimal association rules which are the same as the ones found by using the Apriori algorithm.

Example 1. Suppose a small transaction dataset has $k = 3$ predictor items I_1, I_2, I_3 and two response items I_C, I_{NC}. Also suppose that the marginal probability of vector (J_1, J_2, J_3, J_C) is $\boldsymbol{p} = (0.5, 0.5, 0.5, 0.5)$ and the correlation matrix for (J_1, J_2, J_3, J_C) is

$$R = \begin{pmatrix} 1 & 0 & 0 & 0.8 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.2 \\ 0.8 & 0 & 0.2 & 1 \end{pmatrix}.$$

Then we generate $n = 100$ binary vectors of (J_1, J_2, J_3, J_C) according to $(\boldsymbol{p}, R)$. We compute $J_{NC} = 1 - J_C$. Then we obtain a transaction dataset containing 100 transactions on 5 items $I_1, I_2, I_3, I_C, I_{NC}$. For each response item, there is in total $2^k - 1 = 7$ possible association rules. We first use the Apriori algorithm (3) to mine all association rules of the form $(\boldsymbol{J} \Rightarrow I_C)$ or $(\boldsymbol{J} \Rightarrow I_{NC})$ with support and confidence greater than 0 and summarize the results in Table 1. We choose $g_-(\boldsymbol{J}) = \mathrm{supp}(\boldsymbol{J} \Rightarrow I_-) \times \mathrm{conf}(\boldsymbol{J} \Rightarrow I_-)$ with $I_- = I_C$ or I_{NC} depending on mining $\mathcal{R}_C$ or $\mathcal{R}_{NC}$, for illustration purposes. Then we use the proposed Gibbs sampling algorithm to generate three random samples of size $M = 1{,}000$ of association rules from the transaction dataset by choosing $\xi = 3, 6, 10$, respectively. The frequency of each association rule appearing in each sample is shown in Table 1. The rank of the frequency conforms to that of the importance $g_-(\boldsymbol{J})$, showing

Table 4. Top 10 important association rules from T_1 (left), T_2 (middle), and T_3 (right) and their frequencies in the relevant sample

Association rules	Supp	Conf	$g(\cdot)$	Frequency	Association rules	Supp	Conf	$g(\cdot)$	Frequency	Association rules	Supp	Conf	$g(\cdot)$	Frequency
$I_2 \Rightarrow I_C$	0.787	1.000	0.787	0.20	$I_2 \Rightarrow I_C$	0.787	1.000	0.787	0.20	$I_1, I_3 \Rightarrow I_C$	0.783	0.996	0.780	0.26
$I_3 \Rightarrow I_C$	0.783	1.000	0.783	0.12	$I_3 \Rightarrow I_C$	0.783	1.000	0.783	0.12	$I_1, I_2, I_3 \Rightarrow I_C$	0.783	0.996	0.780	0.23
$I_1 \Rightarrow I_C$	0.783	1.000	0.783	0.26	$I_1 \Rightarrow I_C$	0.783	1.000	0.783	0.26	$I_3 \Rightarrow I_C$	0.793	0.979	0.777	0.15
$I_2, I_3 \Rightarrow I_C$	0.783	1.000	0.783	0.12	$I_2, I_3 \Rightarrow I_C$	0.783	1.000	0.783	0.12	$I_2, I_3 \Rightarrow I_C$	0.787	0.987	0.777	0.21
$I_1, I_2 \Rightarrow I_C$	0.783	1.000	0.783	0.10	$I_1, I_2 \Rightarrow I_C$	0.783	1.000	0.783	0.10	$I_1, I_2 \Rightarrow I_C$	0.783	0.983	0.770	0.08
$I_1, I_3 \Rightarrow I_C$	0.780	1.000	0.780	0.10	$I_1, I_3 \Rightarrow I_C$	0.780	1.000	0.780	0.10	$I_1 \Rightarrow I_C$	0.783	0.975	0.764	0.03
$I_1, I_2, I_3 \Rightarrow I_C$	0.780	1.000	0.780	0.09	$I_1, I_2, I_3 \Rightarrow I_C$	0.780	1.000	0.780	0.09	$I_2 \Rightarrow I_C$	0.787	0.963	0.758	0.03
$I_3, I_{286} \Rightarrow I_C$	0.213	1.000	0.213	0.00	$I_1, I_{13} \Rightarrow I_C$	0.450	1.000	0.450	0.00	$I_3, I_8 \Rightarrow I_C$	0.610	0.995	0.607	0.00
$I_1, I_{286} \Rightarrow I_C$	0.213	1.000	0.213	0.00	$I_2, I_{13} \Rightarrow I_C$	0.450	1.000	0.450	0.00	$I_3, I_5 \Rightarrow I_C$	0.607	1.000	0.607	0.00
$I_2, I_{286} \Rightarrow I_C$	0.213	1.000	0.213	0.00	$I_1, I_2, I_{13} \Rightarrow I_C$	0.450	1.000	0.450	0.00	$I_1, I_3, I_8 \Rightarrow I_C$	0.607	1.000	0.607	0.00

"·" in $g(\cdot)$ represents the association rule $\boldsymbol{J} \Rightarrow I_C$.

the good performance of our method. It is easy to see that the frequencies have more power to differentiate the most important rules from the less important ones, as the value of ξ increases.

Next we illustrate how to use the random search procedure and how well it performs on three more complex datasets.

Example 2. Consider an item space $I=(I_1,I_2,\ldots,I_{398},I_C,I_{NC})$ with $k=398$ predictor items and two response items. Set each marginal probability of (J_1,J_2,J_3,J_C) to 0.8, and the marginal probability of the other predictor items to 0.2, i.e.,

$$\begin{aligned}\boldsymbol{p}&=\{p_1,p_2,p_3,p_4,\ldots,p_{398},p_C\}\\&=\{0.8,0.8,0.8,0.2,\ldots,0.2,0.8\}.\end{aligned}$$

The correlation matrix R between items is set to be an identity matrix except that $R(J_{s_1},J_{s_2})=0.99$ where $s_1,s_2\in\{1,2,3,C\}$. Then we generate $n=300$ binary vectors from $(J_1,J_2,\ldots,J_{398},J_C)$ according to $(\boldsymbol{p},R)$. The transaction dataset T_1 is accordingly formed to contain 400 items and 300 transactions knowing that the status of I_{NC} in each transaction is completely determined by $J_{NC}=1-J_C$.

Example 3. The transaction dataset T_2 has the same item space, the same number of transactions, and the same correlation matrix as T_1 but a different marginal probability vector

$$\begin{aligned}\boldsymbol{p}&=\{p_1,p_2,p_3,p_4,\ldots,p_{20},p_{21},\ldots,p_{398},p_C\}\\&=\{0.8,0.8,0.8,0.5\ldots,0.5,0.2,\ldots,0.2,0.8\}.\end{aligned}$$

Example 4. The transaction dataset T_3 also has $l=400$ items and $n=300$ transactions. The marginal probability vector is

$$\begin{aligned}\boldsymbol{p}&=\{p_1,p_2,p_3,p_4,\ldots,p_{10},p_{11},\ldots,p_{398},p_C\}\\&=\{0.8,0.8,0.8,0.6\ldots,0.6,0.2,\ldots,0.2,0.8\}.\end{aligned}$$

The correlation matrix R is an identity matrix except that

$$\begin{aligned}&R(J_{s_1},J_{s_2})=0.9,\ \text{for } s_1\neq s_2;\ s_1,\ s_2\in\{1,2,3,C\},\\&R(J_{s_1},J_{s_2})=0.5,\ \text{for } s_1\neq s_2;\ s_1,\ s_2\in\{4,\ldots,10,C\},\\&R(J_{s_1},J_{s_2})=0.5,\ \text{for } s_1\in\{1,2,3\},\ s_2\in\{4,5,\ldots,10\}.\end{aligned}$$

From the settings of T_1, T_2, and T_3, we see that items I_1, I_2, and I_3 have high support and the antecedents of the important association rules in these datasets most likely contain some of I_1, I_2, and I_3. We now use the Apriori algorithm and the new Gibbs-sampling-based search procedure to see whether we can unveil these attributes in T_1, T_2, and T_3.

To mine the association rules in $\mathcal{R}_C$ of each transaction dataset, a random sample of size $M=100$ association rules is generated from each $\mathcal{R}_C$ using the new algorithm. We find that the larger ξ is, the more frequently the three items I_1,I_2,I_3 appear in the generated sample. When $\xi=100$, all items ever appearing in the sample are I_1,I_2,I_3, and I_{390}. Proportions of the sampled association rules containing each of (I_1,I_2,I_3,I_{390}) from T_1, T_2, and T_3 are shown in Table 2. The item I_{390} appears only once in each sample, thus seeming not to have high support in the datasets.

We then apply the Apriori algorithm with the constraint of minimum support 0.05 and minimum confidence 0.6 on the search. This identifies 31,525, 170,600, and 442,191 association rules from T_1,T_2,T_3 respectively. The 10 most frequent items appearing in these rules for each dataset and their respective proportions of appearance are shown in Table 3. For each dataset the top 10 of the identified rules according to the importance $g(\cdot)$ are also calculated and presented in Table 4, together with their respective frequencies of appearance in the corresponding random sample generated. Ranks of the top 10 rules in terms of the frequencies in Table 4 more or less conform to their ranks in terms of the importance measure. We find that as the dependence structure of the transaction dataset becomes more complicated, our algorithm can generate a random sample containing the most important association rules that are confirmed by the Apriori algorithm.

To mine the association rules in $\mathcal{R}_{NC}$, we first use the Apriori algorithm with the minimum support and confidence setting (0.05, 0.6) for each dataset, and find no rules. But, a conclusion of having no important association rules in $\mathcal{R}_{NC}$ cannot be drawn yet because it is computationally infeasible to use the Apriori algorithm to search a larger collection of itemsets by weakening the minimum support and confidence constraint. Then, in the hope of making a difference we use the proposed algorithm with various values for ξ. The number of items ever appearing in the generated samples decreases from 398 to around 100 when $\xi=1{,}000$. But, it could not be further reduced by increasing ξ except for T_1. Also we are still unable to find any important rules from the generated samples. Hence we tend to conclude that there are no important association rules in $\mathcal{R}_{NC}$ for the simulated datasets T_1, T_2, and T_3. Now we apply the Apriori algorithm with minimum support and confidence (0.05, 0.6) on the subset of each transaction dataset that includes only I_{NC} and other items ever appearing in each generated random sample using $\xi=1{,}000$, and again are unable to find any important rules of the form $(\boldsymbol{J}\Rightarrow I_{NC})$. These results conform to the setting used to simulate the transaction datasets T_1, T_2, and T_3, where $p_{NC}=0.2$ is small.

From Examples 2–4, we see that our method is capable of finding the most important association rules that also appear most frequently in the random sample generated by properly choosing a large value for ξ. In cases where the item space is large and the support of rules is very low, our proposed algorithm can be combined with the Apriori algorithm to more efficiently tackle the association rule mining task.

Table 5. Top 10 important association rules in $\mathcal{R}_{NC}$ for the SNPs data and their sampling frequencies of appearance

				Frequency in the random sample		
Association rules	Supp($\boldsymbol{J}\Rightarrow I_{NC}$)	Conf($\boldsymbol{J}\Rightarrow I_{NC}$)	$g(\boldsymbol{J}\Rightarrow I_{NC})$	$\xi=100$	$\xi=200$	$\xi=300$
$I_{136}\Rightarrow I_{NC}$	0.803	0.848	0.681	0.220	0.727	0.957
$I_{906}\Rightarrow I_{NC}$	0.812	0.830	0.674	0.117	0.123	0.030
$I_{110}\Rightarrow I_{NC}$	0.795	0.839	0.667	0.037	0.050	0
$I_{10}\Rightarrow I_{NC}$	0.790	0.842	0.665	0.027	0.030	0
$I_{136}, I_{906}\Rightarrow I_{NC}$	0.786	0.845	0.664	0.023	0.023	0.003
$I_{874}\Rightarrow I_{NC}$	0.795	0.831	0.660	0.023	0	0
$I_{136}, I_{874}\Rightarrow I_{NC}$	0.773	0.851	0.658	0.023	0	0
$I_{110}, I_{136}\Rightarrow I_{NC}$	0.769	0.854	0.657	0.023	0.003	0
$I_{10}, I_{136}\Rightarrow I_{NC}$	0.764	0.858	0.656	0.020	0.003	0
$I_{191}\Rightarrow I_{NC}$	0.795	0.824	0.655	0.017	0.020	0

Table 6. Top 10 frequent items appearing in the random samples of association rules for I_C (lower section, i.e., the bottom six rows) and I_{NC} (upper section)

$\xi = 100$	Item	I_{1004}	I_{1061}	I_{589}	I_{1066}	I_{874}	I_{191}	I_{10}	I_{110}	I_{906}	I_{136}
	Proportion	0.043	0.063	0.070	0.070	0.077	0.080	0.137	0.163	0.257	0.470
$\xi = 200$	Item				I_{1061}	I_{1066}	I_{191}	I_{10}	I_{110}	I_{906}	I_{136}
	Proportion				0.007	0.007	0.027	0.033	0.057	0.150	0.767
$\xi = 300$	Item							I_{1061}	I_{1066}	I_{906}	I_{136}
	Proportion							0.007	0.007	0.033	0.963
$\xi = 2{,}700$	Item	I_{750}	I_{45}	I_{1004}	I_{42}	I_{389}	I_{804}	I_{191}	I_{193}	I_{214}	I_{711}
	Proportion	0.60	0.63	0.70	0.72	0.86	0.92	0.98	0.997	0.997	0.997
$\xi = 3{,}500$	Item	I_{914}	I_{750}	I_{42}	I_{389}	I_{1004}	I_{191}	I_{193}	I_{214}	I_{711}	I_{804}
	Proportion	0.64	0.71	0.74	0.95	0.97	0.99	0.99	0.99	0.99	0.99
$\xi = 6{,}000$	Item	I_{937}	I_{45}	I_{750}	I_{1004}	I_{389}	I_{214}	I_{711}	I_{191}	I_{193}	I_{804}
	Proportion	0.65	0.67	0.67	0.84	0.90	0.93	0.96	0.99	0.99	0.99

Real Data Application. We apply the proposed Gibbs sampling method to analyze a case-control dataset that contains genomic observations for $n = 229$ women, 39 of which are breast cancer cases obtained from the Australian Breast Cancer Family Study (ABCFS) (9) and 190 of which are controls from the Australian Mammographic Density Twins and Sisters Study (AMDTSS) (10). The dataset is formed by sampling from a much larger data source from ABCFS and AMDTSS. Each woman in the dataset has 366 genetic observations being the genotype outcomes (from a Human610-Quad beadchip array) of the 366 SNPs on a specific gene pathway suspected to be susceptible to breast cancer. An SNP variable typically takes a value from 0, 1, and 2, representing the number of the minor alleles at the SNP loci. But, in the current dataset there are 31 SNPs, with only 2 of the 3 possible values being observed. Our task is to find out whether there are any SNPs having significant associations with the risk of breast cancer and what these SNPs are. One could use a logistic model to tackle this task. But, it is difficult due to that the number of predictor variables (i.e., SNPs) in the data is much larger than the number of observations, and the SNPs are highly associated with each other due to linkage disequilibrium. Because this dataset can be easily turned into a transaction one, we are able to use an association rule-mining method to undertake the task. The binary transaction dataset converted from our case-control dataset contains 1,067 predictor (SNP) items (denoted as $I_1, \ldots, I_{1067}$) and 2 response items I_C (breast cancer) and I_{NC} (no breast cancer). It is easy to see that $0 \le \mathrm{supp}(\boldsymbol{J} \Rightarrow I_C) \le 0.17$ and $0 \le \mathrm{supp}(\boldsymbol{J} \Rightarrow I_{NC}) \le 0.83$. We choose the importance measure of association rules as $g_-(\boldsymbol{J}) = \mathrm{supp}(\boldsymbol{J} \Rightarrow I_-) \times \mathrm{conf}(\boldsymbol{J} \Rightarrow I_-)$, I_- being either I_C or I_{NC} for illustration purposes. Now our aim is to find the most important association rules for I_C and I_{NC}.

To find an estimate of the most important rule in $\mathcal{R}_{\mathcal{NC}}$, we first generate a random sample of $M = 300$ association rules from the converted transaction dataset with $\xi = 100$, 200, or 300, respectively. The frequency of each association rule appearing in each sample is computed and shown in Table 5. We can see that the difference in the frequencies of association rules becomes larger as the value of ξ increases, which makes the most important association rules stand out. The top 10 frequent items ever appearing in each of the three samples and their proportions of appearance in the respective sample are shown in the upper section of Table 6, where we see the item I_{136} is the most frequent item ever appearing in each generated sample. Moreover, the most important association rule $I_{136} \Rightarrow I_{NC}$ appears the most frequently in each random sample (see the top part in Table 5).

We then try to use the Apriori algorithm to find the most important association rules in $\mathcal{R}_{\mathcal{NC}}$ with different specifications of the minimum support and confidence setting. The top 10 important association rules can be found by the Apriori algorithm with the minimum support 0.6 and minimum confidence 0.8 setting, and their various measures are shown in Table 5. It can be shown that the association rule $I_{136} \Rightarrow I_{NC}$ is indeed the most important rule in R_{NC} which is the same rule found by the stochastic search and confirms the good performance of our proposed method.

For the association rules in $\mathcal{R}_{\mathcal{C}}$, the support of any of them is not greater than 0.17. Because the support of rules is too low and the item space is very large, the Apriori algorithm cannot cope with the computing intensity and immensity involved, even with the setting of minimum support 0.2 and minimum confidence 1. So, we try to use our proposed method to find the most important rule with consequent I_C or reduce the size of the item space. The number of items appearing in the generated samples decreases from 1,067 to about 35 by increasing ξ from 10 to 6,000. But, it cannot be further reduced by larger value of ξ. The top 10 frequent items ever appearing in the generated samples are reported in the lower portion of Table 6. For illustration purposes we choose $\xi = 6{,}000$, with which the number of distinct items appearing in the random sample is 35. We apply the Apriori algorithm on the subset of transaction dataset including only these 35 items by specifying the minimum support and

Table 7. Top 10 association rules for I_C after reducing the item space

Association rules	Supp($\boldsymbol{J} \Rightarrow I_C$)	Conf($\boldsymbol{J} \Rightarrow I_C$)	$g(\boldsymbol{J} \Rightarrow I_C)$
$I_7, I_{42}, I_{750}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_{645}, I_{914}, I_{42}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_{645}, I_{42}, I_{937}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_{636}, I_{914}, I_{42}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_{636}, I_{42}, I_{937}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_7, I_{45}, I_{750}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_{645}, I_{914}, I_{45}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_{645}, I_{937}, I_{45}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_{636}, I_{914}, I_{45}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061
$I_{636}, I_{937}, I_{45}, I_{1004}, I_{389}, I_{214}, I_{711}, I_{191}, I_{193}, I_{804} \Rightarrow I_C$	0.066	0.938	0.061

confidence as 0.2 and 1, respectively. The Apriori algorithm is still not implementable. So, we then single out a subset of 22 items from the 35 items which appeared in at least three-fourths of the sampled association rules and cut out a new subset of the original transaction dataset by including only these 22 items in the transactions. By specifying the minimum support and confidence as 0.05 and 0.6, a total number of 286,188 association rules have been found in the new subset transaction data. The top 10 important association rules among them are reported in Table 7. From Table 7, we see that the measurements of importance of these association rules are very low and close to each other. It is not possible to find out these rules by applying the Apriori algorithm alone. Our proposed Gibbs-sampling-based algorithm can be used to reduce the number of items for mining; the reduced data subset is exactly where the Apriori algorithm can be applied to find the most important association rules subject to negligible information loss. One could look into these rules or the frequent items in Tables 6 and 7 to find out the biological meaning behind them.

ACKNOWLEDGMENTS. This work is partially supported by the Natural Sciences and Engineering Research Council of Canada. G.Q.'s research is also partly supported by the Australian National Health and Medical Research Council Project Grant APP1033452.

1. Agrawal R, Imielinski T, Swami A (1993) Mining association rules between sets of items in large databases. *ACM SIGMOD Rec* 22(2):207–216.
2. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Data Bases* (Morgan Kaufmann, San Francisco), pp 487–499.
3. Hahsler M, Grün B, Hornik K (2005) arules – A computational environment for mining association rules and frequent item sets. *J Stat Softw* 14(15):1–25.
4. Bayardo RJ, Agrawal R (1999) Mining the most interesting rules. *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York), pp 145–154.
5. Hämäläinen W (2009) Statapriori: An efficient algorithm for searching statistically significant association rules. *Knowl Inf Syst* 23(3):373–399.
6. Qian G, Field C (2002) Using MCMC for logistic regression model selection involving large number of candidate models. *Monte Carlo and Quasi-Monte Carlo Methods 2000,* eds Fang K-T, et al. (Springer, Berlin), pp 460–474.
7. Amatya A, Demirtas H (2015) MultiOrd: An R package for generating correlated ordinal data. *Commun Stat Simul Comput* 44(7):1683–1691.
8. Chaganty NR, Joe H (2006) Range of correlation matrices for dependent Bernoulli random variables. *Biometrika* 93(1):197–206.
9. Dite GS, et al. (2003) Familial risks, early-onset breast cancer, and BRCA1 and BRCA2 germline mutations. *J Natl Cancer Inst* 95(6):448–457.
10. Odefrey F, et al.; Australian Twins and Sisters Mammographic Density Study (2010) Common genetic variants associated with breast cancer and mammographic density measures that predict disease. *Cancer Res* 70(4):1449–1458.