Association Rule Mining for Genome-Wide Association Studies through Gibbs Sampling

Guoqi Qian (qguoqi@unimelb.edu.au)
University of Melbourne
Pei-Yun Sun
University of Melbourne

Research Article

Keywords: Gibbs sampling, association rule mining, genome-wide association study, genotype-phenotype association, epistatic interaction

Posted Date: June 24th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1768333/v1

Association Rule Mining for Genome-Wide Association Studies through Gibbs Sampling

Guoqi Qian · Pei-Yun Sun

Received: date / Accepted: date

Abstract Finding associations between genetic markers and a phenotypic trait such as coronary artery disease (CAD) is of primary interest in genome-wide association studies (GWAS). A major challenge in GWAS is the involved genomic data often contain large number of genetic markers and the underlying genotype-phenotype relationship is mostly complex. Current statistical and machine learning methods lack the power to tackle this challenge with effectiveness and efficiency. In this paper we develop a stochastic search method to mine the genotype-phenotype associations from GWAS data. The new method generalizes the well-established association rule mining (ARM) framework for searching for the most important genotype-phenotype association rules, where we develop a multinomial Gibbs sampling algorithm and use it together with the Apriori algorithm to overcome the overwhelming computing complexity in ARM in GWAS. Three simulation studies based on synthetic data are used to assess the performance of our developed method, delivering the anticipated results. Finally, we illustrate the use of the developed method through a case study of CAD GWAS.

Statements and Declarations: The authors have no relevant financial or non-financial interests to disclose. The authors have no competing interests to declare that are relevant to the content of this article.

Guoqi Qian, corresponding and first author
School of Mathematics and Statistics, The University of Melbourne, Parkville VIC3010, Australia
E-mail: qguoqi@unimelb.edu.au

Pei-Yun Sun, first author
School of Mathematics and Statistics, The University of Melbourne, Parkville VIC3010, Australia
E-mail: pssun@unimelb.edu.au

Keywords Gibbs sampling · association rule mining · genome-wide association study · genotype-phenotype association · epistatic interaction

1 Introduction

Genome-wide association study (GWAS) is an observational study for discovering associations between genetic markers and certain phenotypic trait such as coronary artery disease (CAD). Findings from a GWAS can help in reliably predicting an individual's risk of having this disease and in developing effective ways for prevention or treatment [24]. Different forms of genetic markers may be considered in GWAS. A simple and widely used one of them, called single nucleotide polymorphism (SNP), refers to the specific genetic variants (i.e. two alleles) occurred in two corresponding base positions (loci) in a pair of chromosomes in at least 1% of the population. Numerically a SNP is characterized as the number of minor allele at the loci, thus takes 3 possible values 0, 1 and 2, representing the three states of the alleles pair: homozygous recessive, heterozygous, and homozygous dominant. Typically, observations are available on hundreds of thousands or even millions of SNPs in a GWAS with a sample of hundreds of cases and controls, thanks to the use of modern high-throughput DNA sequencing techniques. Also the SNPs are likely to be highly correlated in-between knowing they dwell in the tiny chromosome space. These complications present significant computational challenges on analyzing the intrinsic SNPs versus phenotype associations by the current statistical and machine learning methods.

Many statistical methods for GWAS formulate the SNPs versus phenotype associations by various regres-

sion models for categorical data, and then assess the significance of each individual SNP-phenotype association by single-locus tests [25]. The regression models that have been used in GWAS include ANOVA [13], generalized linear models (GLMs) [16], and (generalized) linear mixed models (LMMs) [27]. Principal components of the SNPs are sometimes used in GLMs to reduce the effects of false positives attributed to the population substructure. Approaches based on LMMs, e.g. CMLM [28] and ECMLM [15], have been shown to be successful in dealing with both population substructure and relatedness in genomic data by treating them as fixed effects and random effects respectively.

Yet it is well recognized that some phenotypes or diseases have complex genetic etiologies. In such cases, each individual SNP may have a weak marginal effect or no effect on the disease, but a combination of some SNPs can synergistically contribute to the risk of the disease. This has brought forth the need for including epistatic (aka. gene-gene) interactions into the statistical regression models. However, in the presence of hundreds of thousands of SNPs in GWAS, the number of interaction terms grows exponentially when the interaction order increases, leading to "large p, small n" scenarios with computational difficulties [12]. In these situations, exhaustive testing methods are not able to be scaled to interactions of higher orders than the pairwise ones.

To overcome this exponential explosion challenge, two-stage approaches have been developed to search for epistasis through dimension reduction and variable selection. Several penalized regression methods such as LASSO regression [22] and Elastic-Net [29] have been used in GWA studies by e.g. [26] and [6] respectively in a two-stage manner: screen the whole genome to get a small set of SNPs potentially having significant main effects on the disease, then apply variable selection to identify all significant SNPs' main and interaction effects from the small SNPs set obtained from screening. Some other two-stage approaches focus more on the screening stage. For example, Fan and Lv [7] developed a dimension reduction method via correlation learning, which is called Iterative Sure Independence Screening (ISIS). This method has been extended to GWAS, resulting in several enhanced versions of ISIS, such as GWASelect [10], EPISIS [23], and TS-SIS [14] etc. Nevertheless, most of these methods are only capable of analyzing those effects involving a small number of selected SNPs that have strong marginal effects but weak interaction ones.

In recent decades, machine learning has been widely used to tackle high-dimensional data analysis problems, which inspires its applications to GWAS. Among all machine learning methods, random forest and association rule mining seem to be the two typical methods for identifying important SNP-phenotype associations in GWAS. Random forest, which is a supervised learning method, can rank the importance of each SNP in terms of its association with the phenotype in an ensemble of classification trees. An importance score, e.g. Gini index or cross-entropy, for each SNP is defined based on certain loss function expressing the error in predicting the phenotype and can be easily computed from growing trees [4]. Since this importance score is calculated in the presence of other variables in the model, it is particularly suitable to be used at the filtering stage (stage 1) in the two-stage GWAS approaches.

As for association rule mining (ARM), it is an unsupervised machine learning method designed to search for important associations rules made up from an arbitrary number of items [3]. Unlike the SNP-phenotype association terms identified from a statistical regression or supervised learning model, the SNP-phenotype association rules identified from ARM are formulated based on various concepts used in set theory. Thus, the abovementioned association rules and the association terms explain different types of SNP-phenotype associations which may not be the same to each other. This would be welcome since they provide complementary information to GWAS.

Early ARM methods are computationally intensive for mining even a moderate-sized basket of items. Recently, several parallel and distributed computing techniques such as GARMS [1] and BPARES [2] have been applied to boost ARM but it could still be hard to mine large-scale GWAS data due to the underlying combinatory explosion in constructing rules. Qian et al. [17] proposed a Gibbs-sampling induced stochastic search approach to mine the rules efficiently. However, this approach treats each item as a binary variant, which does not apply to ARM in GWAS because the SNPs there are multinomial variants. In this paper, we will develop a multinomial Gibbs-sampling induced new stochastic search method for ARM in GWAS, which scales well to large-scale GWAS data.

The rest of the paper is structured as follows. In Section 2 we introduce the format of GWAS data and the framework of ARM first before developing the random (stochastic) search algorithm `MultinomRSA`. The core of this algorithm is a Gibbs sampler having multinomial marginal conditional distributions. The `MultinomRSA` algorithm is particularly suitable for mining SNPs versus phenotype association rules because each SNP item is better modelled by a multinomial random variable. In Section 3, we assess the performance of our method using simulation studies and a real-world CAD dataset.

Finally, in Section 4, we provide conclusions and an overview of potential areas for further research.

2 Methodology

2.1 Data, Conceptions and Notations

2.1.1 GWAS data

According to its genetic definition, a SNP can be numerically coded as the number of minor alleles at the corresponding pair of loci. The process of obtaining coded values for all observed SNPs is called genotyping. Normally there are two types of alleles in human genome: major allele and minor allele. Also there are two alleles dwelling at each locus. Therefore, each genotyped SNP variable has 3 possible values: 0, 1, and 2, representing the three states of the alleles pair: homozygous recessive, heterozygous, and homozygous dominant. For example, if the minor allele is denoted as T and the major is A, then SNP $= 0$ if AA is observed; $= 1$ if AT or TA is observed; and $= 2$ if TT is observed at the loci.

A real-world GWAS example to be presented in Section 3 uses a large portion of the PennCATH dataset that is collected from a case-control observational study of CAD [20]. The PennCATH data consist of the genotyped values of 861,473 SNPs across 3850 individuals whose coronary artery disease (CAD) information and information on cardiovascular risk factors are also available. If the phenotype CAD and the genotyped SNPs can be regarded as a set of items, relations between CAD and the SNPs can be mined by ARM. In order to see this, we provide a brief review of ARM in the following.

2.1.2 Association rule mining framework

Association rule mining (ARM) was developed in [3] for mining interesting relations between items from transactions data recorded by point-of-sale systems in supermarket. For example, the rule {ham, cheese}$\rightarrow${bread} found in a supermarket's sales data would suggest that a customer buying ham and cheese is likely to buy bread as well. Information on how frequent this rule is observed in the sales data can be used as the basis for certain marketing decisions made by the supermarket.

A common formulation of ARM is given in [9], being summarized here. Define $\mathcal{I} = \{I_1, I_2, \cdots, I_m\}$ as a set of m items called the *item space* and $\mathcal{D} = \{\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_L\}$ as a list of transactions, where each transaction in $\mathcal{D}$ is just a subset of items in $\mathcal{I}$, i.e. $\mathbf{t}_l \subset \mathcal{I}$, $l = 1, \cdots, L$. An *association rule* is defined as an implication of the form $\mathbf{X} \rightarrow \mathbf{Y}$ where $\mathbf{X}, \mathbf{Y} \subset \mathcal{I}$ and $\mathbf{X} \cap \mathbf{Y} = \emptyset$. The sets of items (for short *itemsets*) $\mathbf{X}$ and $\mathbf{Y}$ are called *antecedent* and *consequent* of the rule, respectively. The *support* of an itemset, $\mathbf{X}$, supp($\mathbf{X}$) is defined as the proportion of transactions in $\mathcal{D}$ which contain $\mathbf{X}$. The *confidence* of an association rule is defined as

$$\text{conf}(\mathbf{X} \rightarrow \mathbf{Y}) = \frac{\text{supp}(\mathbf{X}\&\mathbf{Y})}{\text{supp}(\mathbf{X})}$$

where $\mathbf{X}\&\mathbf{Y}$ is the itemset obtained by amalgamating $\mathbf{X}$ with $\mathbf{Y}$. Define $\text{conf}(\mathbf{X} \rightarrow \mathbf{Y}) = -\infty$ if $\text{supp}(\mathbf{X}) = 0$, to indicate no need to measure conf($\mathbf{X} \rightarrow \mathbf{Y}$) if the itemset $\mathbf{X}$ is not observed in the transactions dataset $\mathcal{D}$. 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 $\mathbf{X} \rightarrow \mathbf{Y}$, which is just

$$\text{supp}(\mathbf{X} \rightarrow \mathbf{Y}) \equiv \text{supp}(\mathbf{Y} \rightarrow \mathbf{X}) \equiv \text{supp}(\mathbf{X}\&\mathbf{Y}).$$

By ARM we aim to find out the rules that have high support or high confidence or some other properly defined metrics.

Note that for transactions data generated from the item space $\mathcal{I}$, any itemset $\mathbf{X}$ containing k items in $\mathcal{I}$ can be equivalently expressed as a binary indicator vector $\mathbf{V}(\mathbf{x}) = (J_1, \cdots, J_m)$, where $J_\ell = 1$ if $I_\ell \in \mathbf{X}$ and $J_\ell = 0$ if $I_\ell \notin \mathbf{X}$, $\ell = 1, \cdots, m$. It is easy to see that the number of distinct transactions that can be generated from $\mathcal{I}$ is $2^m - 1$.

From an initial look, GWAS data do not resemble transactions data. Thus, it seems ARM is not applicable to GWAS data mining. However, a SNP variable having 3 levels can be represented by 3 indicator variables: $J_0^{(\text{SNP})}$, $J_1^{(\text{SNP})}$ and $J_2^{(\text{SNP})}$, where $J_\ell^{(\text{SNP})} = 1$ if the SNP is observed at level ℓ; $J_\ell^{(\text{SNP})} = 0$ otherwise; $\ell = 0, 1, 2$. This implies that a SNP variable can be regarded as a set of 3 items $I_0^{(\text{SNP})}$, $I_1^{(\text{SNP})}$ and $I_2^{(\text{SNP})}$, corresponding to their respective indicators $J_0^{(\text{SNP})}$, $J_1^{(\text{SNP})}$ and $J_2^{(\text{SNP})}$. Also the phenotype variable naturally specifies 2 items I_{D} and I_{ND} corresponding to disease and no disease, respectively. Hence, observations of each individual in a GWAS dataset containing m SNP variables can be converted as a specific transaction that consists of m items from the $3m$ SNP items and one item from I_{D} and I_{ND}, such that each SNP variable contributes one and only one item to the transaction. Such a transaction can be equivalently represented by the binary indicators determined by all items in the transaction.

In order to represent not only a transaction but also any itemset of the m SNPs, we introduce an additional indicator $J_{\text{no}}^{(\text{SNP})}$ which equals 1 or 0 depending

on whether or not the SNP is inside the itemset. Now write

$$\mathbf{J}^{(\text{SNP})} = (J_{\text{no}}^{(\text{SNP})}, J_0^{(\text{SNP})}, J_1^{(\text{SNP})}, J_2^{(\text{SNP})})$$

as a set of 4 binary indicators for a given SNP. Then an itemset $\mathbf{I}(\text{SNPs}1:k)$ containing k observations from k SNPs, denoted as $\{\text{SNP}1, \cdots, \text{SNP}k\}$, can be represented as

$$\mathbf{I}^{(\text{SNPs}1:k)} = (I_{\ell(1)}^{(\text{SNP}1)}, \cdots, I_{\ell(k)}^{(\text{SNP}k)}),$$

where $\ell(j)$ is the observed value of SNPj, $j = 1, \cdots, k$. The corresponding indicator vector for $\mathbf{I}^{(\text{SNPs}1:k)}$ is

$$\mathbf{J}^{(\text{SNPs}1:k)} = (\mathbf{J}^{(\text{SNP}1)}, \cdots, \mathbf{J}^{(\text{SNP}m)}),$$

in which there are m 1's, $3m$ 0's, and $m-k$ indicators of form $J_{\text{no}}^{(\text{SNP})}$ equal 1. It is easy to see that, for an item space of m SNPs and a phenotype, there are up to 2×3^m distinct transactions that can be observed in a GWAS data set, whereas $4^m - 1$ non-empty distinct itemsets can be generated from an item space of m SNPs. In this paper, we are interested in mining the SNPs-phenotype induced association rules of the following forms

$$(\mathbf{J}^{(\text{SNP}_1)}, \cdots, \mathbf{J}^{(\text{SNP}_m)}) \to I_{\text{D}} \quad (1)$$

$$(\mathbf{J}^{(\text{SNP}_1)}, \cdots, \mathbf{J}^{(\text{SNP}_m)}) \to I_{\text{ND}} \quad (2)$$

2.2 ARM Processing and Metrics

Support and *confidence* are the key metrics for evaluating how "interesting" or "informative" an association rule $\mathbf{X} \to \mathbf{Y}$ is. But it is computationally infeasible to search for the most interesting rules based on $\text{supp}(\mathbf{X} \to \mathbf{Y})$ and/or $\text{conf}(\mathbf{X} \to \mathbf{Y})$ by a brute-force approach even if the associated item space has moderate number of items, because the search space has a cardinality of exponential order. Current approaches for tackling this difficulty are to use constrained search. A typical such method is the Apriori algorithm [3] in which one sets thresholds t_{supp}, t_{conf} and t_{len}, respectively for *support*, *confidence* and *length* of each of the rules to be searched. Namely, one either searches for the rules of the highest support(s) subject to

$$\text{length}(\mathbf{X} \to \mathbf{Y}) \le t_{\text{len}} \quad \text{and} \quad \text{conf}(\mathbf{X} \to \mathbf{Y}) \ge t_{\text{conf}};$$

or searches for the rules of the highest confidence(s) subject to

$$\text{length}(\mathbf{X} \to \mathbf{Y}) \le t_{\text{len}} \quad \text{and} \quad \text{supp}(\mathbf{X} \to \mathbf{Y}) \ge t_{\text{supp}}.$$

Effectiveness and efficiency of such a constrained search method critically depend on the selection of t_{supp}, t_{conf} and t_{len}. And it is very difficult to be scaled to ARM on large item space.

In this paper, we propose to use a stochastic search approach instead for ARM to find the most "interesting" or "informative" rules. For this we need a different metric to measure the interestingness of an association rule. In Qian et al. [17] an *importance* measure is proposed for each association rule $\mathbf{X} \to \mathbf{Y}$, which is defined as

$$\text{imp}(\mathbf{X} \to \mathbf{Y}) = f(\text{supp}(\mathbf{X} \to \mathbf{Y}), \text{conf}(\mathbf{X} \to \mathbf{Y})) \quad (3)$$

where $f(\cdot,\cdot)$ can be any positive and increasing function with respect to $\text{supp}(\mathbf{X} \to \mathbf{Y})$ and $\text{conf}(\mathbf{X} \to \mathbf{Y})$. Here we choose

$$\text{imp}(\mathbf{X} \to \mathbf{Y}) = \text{supp}(\mathbf{X} \to \mathbf{Y}) \times \text{conf}(\mathbf{X} \to \mathbf{Y})$$

as the *importance* of $\mathbf{X} \to \mathbf{Y}$. This keeps the same virtue of favouring the rules of high support and confidence in ARM. By our stochastic search method to be developed, we aim to find out those association rules that have the highest importance values.

2.3 Stochastic Bernoulli Gibbs sampling ARM

For an item space $\mathcal{I} = \{I_1, \cdots, I_{m+1}\}$ comprising itemsets of the form $\mathbf{J} = (J_1, \cdots, J_m)$, consider the following collection of association rules

$$\mathcal{R} = \{\mathbf{J}_m \to I_{m+1} \colon \mathbf{J}_m \in \{0,1\}^m, \mathbf{J} \ne \mathbf{0}\}.$$

All rules in $\mathcal{R}$ can be ranked according to their importance values. Denote $\mathbf{J}_{m(k)}$ as the kth order antecedent such that $\text{imp}(\mathbf{J}_{m(k)} \to I_{m+1})$ is the kth highest. When m is moderate or large, we know it is computationally infeasible to find the rules of the highest importance values in $\mathcal{R}$ by a brute-force search method because $2^m - 1$, the cardinality of $\mathcal{R}$, is of exponential order. On the other hand, consider a probability distribution on $\mathcal{R}$ defined by a softmax function, i.e.

$$P_\xi(\mathbf{J}_m) \equiv \Pr(\mathbf{J}_m \to I_{m+1}) = \frac{e^{\xi \cdot \text{imp}(\mathbf{J}_m \to I_{m+1})}}{\sum_{(\mathbf{J}'_m \to I_{m+1}) \in \mathcal{R}} e^{\xi \cdot \text{imp}(\mathbf{J}'_m \to I_{m+1})}} \quad (4)$$

where $\xi > 0$ is a tuning parameter for adjusting the probability ratio between every two rules in $\mathcal{R}$. It is easy to see that those rules in $\mathcal{R}$ having the kth highest importance value also have the kth largest probability defined in (4) for any ξ value. This implies that, if one can generate a random sample of rules from $P_\xi(\mathbf{J}_m)$, those rules having the highest importance values are more likely to appear in the sample and appear earlier in the sample than those rules not having high importance values. Moreover, when ξ is larger, the frequency of a high-importance rule appearing in the sample is even higher than that of a not-high-importance

rule appearing in the sample. Therefore, the problem of finding high-importance rules in $\mathcal{R}$, which is computationally not scalable, can be solved by finding high-importance rules from the random samples generated from (4), which we will see to be computationally feasible. By the probability law of large numbers for binomial and multinomial distributions, we can see the highest-importance rules in the generated samples converge to those highest-importance rules in $\mathcal{R}$ with probability 1, with the relevant approximation error being smaller than $1/\sqrt{\text{sample size}}$ with at least 95% probability. This suggests a polynomial order for the size of the generated samples is sufficient to ensure the convergence. This also means the computational complexity is of the same polynomial order, while that for the brute-force search is of exponential order.

Now the question is how to generate a random sample of rules from the probability distribution $P_\xi(\mathbf{J}_m)$ given by (4) which is a multivariate discrete distribution. This question is not trivial because the denominator in (4) involves $2^m - 1$ terms and is intractable to compute even when m equals upper 10's. Such difficulty can be bypassed by applying Gibbs sampling which is a Markov chain Monte Carlo (MCMC) algorithm aiming to generate a Markov chain from a feasible transition probability matrix such that the stationary distribution of this Markov chain equals the target multivariate discrete distribution.

The involved transition probability matrix in Gibbs sampling is determined by the product of all univariate conditional probability functions of $P_\xi(\mathbf{J}_m)$. It is easy to see that the conditional probability function of each J_s given $\mathbf{J}_{-s} = (\mathbf{J}_{1:(s-1)}, \mathbf{J}_{(s+1):m})$, $s = 1, \cdots, m$, is

$$P_\xi(J_s|\mathbf{J}_{-s}) = \frac{e^{\xi \cdot \text{imp}((\mathbf{J}_{1:(s-1)}, J_s, \mathbf{J}_{(s+1):m}) \to I_{m+1})}}{\sum_{J_s'=0}^{1} e^{\xi \cdot \text{imp}((\mathbf{J}_{1:(s-1)}, J_s', \mathbf{J}_{(s+1):m}) \to I_{m+1})}} \tag{5}$$

which is a Bernoulli probability distribution not involving the intractable denominator of (4). Here $\mathbf{J}_{a:b} = (J_a, J_{a+1}, \cdots, J_b)$ if integers a and b satisfy $a < b$; $\mathbf{J}_{a:b} = J_a$ if $a = b$; and $\mathbf{J}_{a:b} = \emptyset$ if $a > b$.

To implement Gibbs sampling for generating rules from (4) we start from a randomly selected transaction $(\mathbf{J}_m^{(0)}, I_{m+1}) = (J_1^{(0)}, \cdots, J_m^{(0)}, I_{m+1})$ from the transactions dataset, and construct the initial rule $\mathbf{J}_m^{(0)} \to I_{m+1}$. Then generate $J_1^{(1)}$ from $P_\xi(J_1|J_2^{(0)}, \cdots, J_m^{(0)})$ in (5) to substitute $J_1^{(0)}$. Continue to generate $J_s^{(1)}$ from $P_\xi(J_s|J_1^{(1)}, \cdots, J_{s-1}^{(1)}, J_{s+1}^{(0)}, \cdots, J_m^{(0)})$ in (5) to substitute $J_s^{(0)}$, with $s = 2, \cdots, m$. This ends up with the rule $\mathbf{J}_m^{(1)} \to I_{m+1}$, an update of $\mathbf{J}_m^{(0)} \to I_{m+1}$. This procedure is repeated sequentially for N times to get the N rules $\{\mathbf{J}_m^{(n)} \equiv (J_1^{(n)}, \cdots, J_m^{(n)}) \to I_{m+1},\ n = 1, \cdots, N\}$. We name the method just described as the *stochastic Bernoulli Gibbs ARM algorithm.*

By the properties of Gibbs sampling, the corresponding N itemsets $\mathbf{J}_m^{(1)}, \cdots, \mathbf{J}_m^{(n)}$ constitute a Markov chain having the probability distribution (4) as its unique stationary distribution. Therefore, the most important rules in $\mathcal{R}$ can be determined (with probability 1) from the generated Markov chain, and there are at least three ways to identify them.

Firstly, the most important rules can be determined by the most frequent itemsets among $\mathbf{J}_m^{(1)}, \cdots, \mathbf{J}_m^{(n)}$. However, this could be ineffective if the frequency of each distinct itemset in $\mathbf{J}_m^{(1)}, \cdots, \mathbf{J}_m^{(n)}$ is very small, e.g. 1 or 2, which is very likely the case if N and ξ are not large enough.

The second method is to identify the most important rules among the N generated rules and use them to estimate the most important rules in $\mathcal{R}$. By the ergodicity theorem for Markov chain, the most important sampled rules among the N generated converge to the most important population rules in $\mathcal{R}$ with probability 1 if the underlying Markov chain has a finite state space and satisfies the detailed balance condition. The most important sampled rules can be easily identified by computing the importance values for all the N generated rules. The second method is mostly effective if m is not very large and the generated Markov chain is sufficiently ergodic.

By the third method, we apply a two-step approach which first identifies the most frequent items from $\mathbf{J}_m^{(1)}$, $\cdots$, $\mathbf{J}_m^{(n)}$, and use these items to constitute a new item space, then apply Gibbs sampling again to determine the most important rules from the collection of association rules given by the new item space. Since the cardinality of the new item space will mostly be smaller than that of the original item space, its most important rules can be more efficiently identified by the second step Gibbs sampling or Apriori algorithm. And these most important rules will converge to the most important rules in $\mathcal{R}$ with probability 1 by the probability law of large numbers and the ergodicity theorem for Markov chain.

Rationales behind the aforementioned three ways of mining important rules are explained in detail in Qian and Zhao [18]. Finally, note that $P_\xi(\mathbf{J}_m)$ defined by (4) equals 0 if an itemset $(\mathbf{J}_m, I_{m+1})$ is not observed in the transaction dataset $\mathcal{D}$. Thus, $P_\xi(\mathbf{J}_m)$ is in effect defined on the collection of all itemsets of $(I_1, \cdots, I_m)$ observed in $\mathcal{D}$. Moreover, the Markov chain generated by Gibbs sampling via conditional probability distribution (5) is still ergodic in the association rule space induced from $\mathcal{D}$. In section 3 we will use simulated data and a real

data case study to demonstrate the effectiveness and efficiency of Gibbs sampling for ARM.

2.4 Stochastic multinomial Gibbs ARM for GWAS

Now we focus on using Gibbs sampling to mine the SNPs-phenotype association induced rules of the form $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}$ that is given in (1). ARM for rules of the form $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{ND}}$ that is given in (2) can be performed in the same way, thus is skipped.

Corresponding to (4) the target probability distribution for GWAS-ARM using Gibbs sampling is

$$P_{D\xi}(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})=\frac{e^{\xi\cdot\text{imp}\left((\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}\right)}}{\sum_{\{\mathbf{J}'^{(\text{SNP}_v)}\in I^{(4)},v=1,\cdots,m\}}e^{\xi\cdot\text{imp}\left((\mathbf{J}'^{(\text{SNP}_1)},\cdots,\mathbf{J}'^{(\text{SNP}_m)})\to I_{\text{D}}\right)}}\quad(6)$$

where $I^{(4)}$ is a collection of 4 row vectors of a 4 matrix giving the four possible row vector values that each itemset $\mathbf{J}'^{(\text{SNP}_v)}$ can take.

Since $\text{imp}\left((\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}\right)=0$ if the itemset $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})$ is not observed in the underlying GWAS dataset, it follows that the domain $\mathcal{D}_{\text{SNP}}$ of the $4m$-variate distribution function $P_{D\xi}$ of (6) is the collection of all possible itemsets of the antecedents of the rules $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}$ in the GWAS dataset. In other words, $\mathcal{D}_{\text{SNP}}\subseteq\{I^{(4)}\}^m$.

To generate a Markov chain having (6) as its stationary distribution, we need the conditional distribution of each $\mathbf{J}^{(\text{SNP}_v)}$ given $\mathbf{J}_{-v}^{(\text{SNPs}1:m)}$ (i.e. all elements of the itemset $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})$ except the vth element, $v=1,\cdots,m$):

$$P_{D\xi}\left(\mathbf{J}^{(\text{SNP}_v)}|\mathbf{J}_{-v}^{(\text{SNPs}1:m)}\right)\equiv P_{D\xi}\left(\mathbf{J}^{(\text{SNP}_v)}|\mathbf{J}^{(\text{SNPs}1:(v-1))},\mathbf{J}^{(\text{SNPs}(v+1):m)}\right)=\frac{e^{\xi\cdot\text{imp}\left((\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}\right)}}{\sum\limits_{\mathbf{J}'^{(\text{SNP}_v)}\in I^{(4)}}e^{\xi\cdot\text{imp}\left((\mathbf{J}^{(\text{SNPs}1:(v-1))},\mathbf{J}'^{(\text{SNP}_v)},\mathbf{J}^{(\text{SNPs}(v+1):m)})\to I_{\text{D}}\right)}}\quad(7)$$

which is a 4-category size-1 multinomial distribution.

Recall that each $\mathbf{J}^{(\text{SNP}_v)}$ has 4 possible values given by $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ and $(0,0,0,1)$, indicating the following 4 situations

1. SNP_v is not inside the rule $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}$ if $\mathbf{J}^{(\text{SNP}_v)}=(1,0,0,0)$.
2. $\text{SNP}_v=0$ is inside the rule $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}$ if $\mathbf{J}^{(\text{SNP}_v)}=(0,1,0,0)$.
3. $\text{SNP}_v=1$ is inside the rule $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}$ if $\mathbf{J}^{(\text{SNP}_v)}=(0,0,1,0)$.
4. $\text{SNP}_v=2$ is inside the rule $(\mathbf{J}^{(\text{SNP}_1)},\cdots,\mathbf{J}^{(\text{SNP}_m)})\to I_{\text{D}}$ if $\mathbf{J}^{(\text{SNP}_v)}=(0,0,0,1)$.

By applying Gibbs sampling to generate N itemsets $\{\mathbf{J}_{(n)}^{(\text{SNPs}1:m)},n=1,\cdots,N\}$ from (6) in the same way as is proceeded in section 2.3, we summarize the procedure by the following algorithm.

Algorithm 1 Stochastic multinomial Gibbs-ARM

Input: N, $\xi>0$, and data (e.g. GWAS data with m genotyped SNPs and a phenotype)
Output: N rules $\{\mathbf{J}_{(n)}^{(\text{SNPs}1:m)}\to I_{\text{D}},n=1,\cdots,N\}$
Initialise a rule $\mathbf{J}_{(0)}^{(\text{SNPs}1:m)}\equiv(\mathbf{J}_{(0)}^{(\text{SNP}_1)},\cdots,\mathbf{J}_{(0)}^{(\text{SNP}_m)})\to I_{\text{D}}$ from the GWAS data.
rules ← [ ]
for `n in` $1,...,N$ **do**
  for v `in` $1,...,m$ **do**
    Generate $\mathbf{J}_{(n)}^{(\text{SNP}_v)}$ from the multinomial distribution $P_{D\xi}\left(\mathbf{J}^{(\text{SNP}_v)}|\mathbf{J}_{(n)}^{(\text{SNPs}1:(v-1))},\mathbf{J}_{(n-1)}^{(\text{SNPs}(v+1):m)}\right)$ given in (7) to replace (update) $\mathbf{J}_{(n-1)}^{(\text{SNP}_v)}$.
  end for
  Append $(\mathbf{J}_{(n)}^{(\text{SNP}_1)},\cdots,\mathbf{J}_{(n)}^{(\text{SNP}_m)})\to I_{\text{D}}$ to *rules*.
end for
return *rules* $\{\mathbf{J}_{(n)}^{(\text{SNPs}1:m)}\to I_{\text{D}},n=1,\cdots,N\}$.

The returned rules $\{\mathbf{J}_{(n)}^{(\text{SNPs}1:m)}\to I_{\text{D}},n=1,\cdots,N\}$ constitute a Markov chain with (6) being its stationary distribution. Therefore, the same three methods described in section 2.3 can be used to find the most important rules from the generated Markov chain.

3 Experiments

In the experiments, we will test the proposed algorithm with simulated datasets and a real-world dataset. The simulated datasets are used to show that the algorithm can accurately and efficiently find the most important association rules, and the real-world dataset containing much larger number of SNPs is for demonstrating the capability of the algorithm to work with big data.

3.1 Simulation Studies

In the simulation studies, we apply our algorithm to 3 simulated transaction datasets to demonstrate its performance under a wide range of scenarios. The datasets were generated by using the R package "SNPSetSimulations", which is used to simulate GWAS-related data. Simulation setting is specified by the users of the package before starting the simulation, which includes the number of SNPs (m), the minor allele frequency (MAF) of the SNPs, and the correlation structures among SNPs (R) and between SNPs and phenotype ($\boldsymbol{\beta}$). Then, based on the given information, it will generate a matrix of

genotyped SNPs data, along with binary phenotype values calculated from a logistic regression model with certain genotyped SNPs as its covariates [8].

It is natural that those SNPs used in calculating the phenotype response values will be inside the antecedents (LHS) of the most important association rules of the corresponding transaction data. However, the SNPs-phenotype association specified by the logistic regression model is not the only important one that can be identified by association rule mining. We expect the most important association rules may contain other SNP items not used in calculating the phenotype response values.

In the following we will use I_{vu} ($v = 1, \cdots, m$ and $u = 1, 2, 3$) to represent the uth level of level-$(u-1)$ item of SNP_v, and use I_{D} and I_{ND} for the positive and negative phenotype, respectively.

Simulation 1 In the first dataset, we start with a small number of SNPs to show that the algorithm can find the most important association rules by comparing them with the rules obtained from the Apriori algorithm.

Consider a dataset having 3 SNP variables with a binary phenotype response. MAF of the SNPs are set to 0.4 and they are assumed to be uncorrelated. In logistic regression under an additive model, the intercept is set to -3 and the coefficient of SNP3 is set to 5. Namely,

$$R = \begin{pmatrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 1 \end{pmatrix}$$

$$\boldsymbol{\beta} = (-3, 0, 0, 5)$$

Then, we use this genotype configuration to generate a sample of 50 cases and 50 controls, amounting to 100 transactions. In association rule mining, each of the SNPs will be represented by 3 items corresponding to the 3 levels of SNP, and there will be 9 SNP items in total, therefore $4^3 - 1 = 63$ possible rules with I_{D} (and respectively I_{ND}) as the consequent (RHS). Ignoring the multinomial constraint within the SNP items, there would be $2^9 - 1 = 512$ possible rules instead.

We first use the Apriori algorithm to find the association rules with $t_{\text{supp}} \geq .01$ and $t_{\text{conf}} \geq .01$. In the experiment, we define "importance" as the product of confidence and support. By the Apriori, 12 rules were found and the top 10 important ones are listed in column 4 of Table 1. Note that I_{ij} represents the jth level of SNP_i. As expected, the top rules in the table are related to SNP3 (items $I_{3\cdot}$). Specifically speaking, they all contain the 2nd level item of SNP3 (I_{32}), and I_{32} alone as the antecedent gives the most important rule.

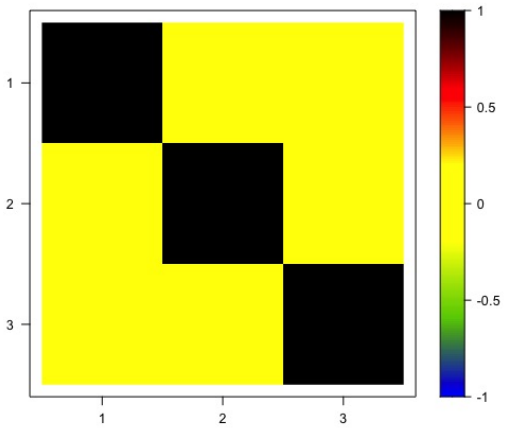

Fig. 1 Sample correlations of the 3 SNPs in Simulation 1

We then generated $N = 1000$ rules from the simulated dataset using either the stochastic multinomial Gibbs sampling ARM method or the Bernoulli Gibbs-ARM method for the tuning parameter $\xi = 10$ or $\xi = 20$. Frequencies of the generated rules in each case are shown in Table 1, from which we see both multinomial and Bernoulli Gibbs-ARM methods have identified the same top 10 important rules as the Apriori algorithm. There are only very small differences in the order of the frequencies in each case.

As mentioned in section 2.3, the tuning parameter ξ controls the ratio of sampling probabilities between each pair of rules generated by Gibbs sampler. Low ξ means this ratio is close to 1, leading to more different rules being generated and less differentiable from each other in terms of the importance. On the other hand, high ξ leads to less different rules being generated and tending to stuck at local maxima with relatively low importance. Therefore, the tuning parameter should be carefully chosen. As compared with $\xi = 10$, when $\xi = 20$, the algorithms tend to result in higher frequencies on the more important rules, which is expected by the sampling properties of our methods.

Simulation 2 SNPs used in Simulation 1 are uncorrelated with each other. This might make it easy for the algorithms to find the SNPs associated with the phenotype. The SNPs are often correlated in the real-world. So, in the second simulation, we use the same setups of Simulation 1 except that the SNPs have an autocorrelation structure with coefficient $\rho = 0.7$ (Figure 2). Namely, the correlation between SNP_i and SNP_j is

$$R(\text{SNP}_i, \text{SNP}_j) = 0.7^{|i-j|}; \quad i, j \in \{1, 2, 3\}$$

Rules ($\rightarrow I_D$)	Supp	Conf	Imp($\downarrow$)	Multinomial Frequency		Bernoulli Frequency	
				$\xi = 10$	$\xi = 20$	$\xi = 10$	$\xi = 20$
I_{32}	0.360	0.878	0.316	0.182	0.740	0.210	0.746
I_{12}	0.290	0.569	0.165	0.039	0.029	0.043	0.039
I_{12},I_{32}	0.190	0.864	0.164	0.036	0.026	0.041	0.047
I_{22},I_{32}	0.150	0.938	0.141	0.037	0.019	0.031	0.02
I_{33}	0.120	1.000	0.120	0.023	0.018	0.024	0.015
I_{21},I_{32}	0.140	0.824	0.115	0.033	0.017	0.024	0.018
I_{11},I_{32}	0.120	0.923	0.111	0.028	0.013	0.017	0.012
I_{22}	0.230	0.469	0.108	0.022	0.007	0.020	0.007
I_{12},I_{22}	0.140	0.583	0.082	0.021	0.008	0.018	0.009
I_{21}	0.170	0.472	0.080	0.017	0.002	0.012	0.001

Table 1 The 10 most important rules found by Apriori ($t_{\text{supp}} \geq 0.01, t_{\text{conf}} \geq 0.01$) with their frequencies of appearing in the $N = 1000$ rules generated by multinomial and Bernoulli Gibbs-ARM methods (Simulation 1).

For this example, the correlation matrix is

$$R = \begin{pmatrix} 1 & 0.70 & 0.49 \\ 0.70 & 1 & 0.70 \\ 0.49 & 0.70 & 1 \end{pmatrix}$$

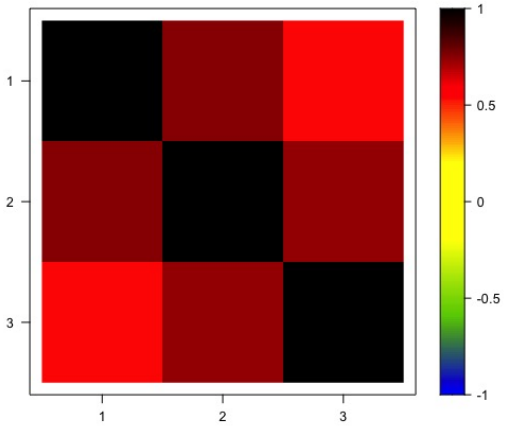

Fig. 2 Sample correlations of the SNPs in Simulation 2

The results are shown in Table 2. Similar to Simulation 1, the algorithms successfully found all the important rules, even when the SNPs are correlated.

Simulation 3 In the third simulation, we generate a more complex dataset to show the computational advantages of using our algorithm.

This dataset contains 100 SNPs with a similar correlation structure as Simulation 2 (Figure 3):

$$R(\text{SNP}_i, \text{SNP}_j) = 0.7^{|i-j|}; \quad i, j \in \{1, 2, ..., 100\}$$
$$\boldsymbol{\beta} = (-3, 1, 2, 3, 0, ..., 0)$$

In this example, MAFs are still set to 0.4, and in the logistic model, the intercept is -3 and the coefficients

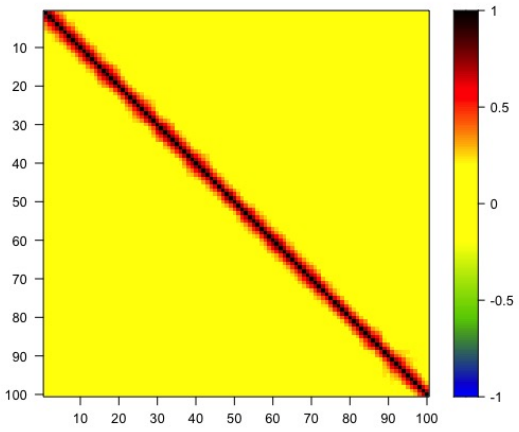

Fig. 3 Sample correlations of the SNPs in Simulation 3

for the first 3 SNPs are 1, 2, and 3 respectively, 0 for the other SNPs. So, we expect to find the rules containing items from SNP1 to SNP3 to be important. Then, a sample of $L = 500$ transactions consisting of 250 cases and 250 controls are generated from this configuration.

When the Apriori algorithm is applied to the 250 cases dataset, we find 216,084 rules satisfying $t_{\text{supp}} \geq 0.05$ and $t_{\text{conf}} \geq 0.5$. The top 10 most important ones among them are displayed in Table 3. From the table, we can see that items I_{12}, I_{22}, I_{32} appear frequently in the top rules. This is because SNP1, SNP2, and SNP3 have non-zero coefficients in the underlying data-generating logistic model and they should have strong associations with the phenotype. Items of SNP4 and SNP5 also appear in the top 10 rules as they are highly correlated with SNP2 and SNP3.

Then, both Gibbs-ARM algorithms are applied to the 250 cases dataset and the associated frequencies in the $N = 100$ generated rules are shown in Table 3. For a large item space, the tuning parameter ξ needs to be large in order to find the important rules efficiently. So we have gradually increased ξ and found that by choos-

Rules ($\rightarrow I_D$)	Supp	Conf	Imp($\downarrow$)	Multinomial Frequency		Bernoulli Frequency	
				$\xi = 10$	$\xi = 20$	$\xi = 10$	$\xi = 20$
I_{32}	0.380	0.884	0.336	0.216	0.717	0.209	0.684
I_{22},I_{32}	0.260	0.867	0.225	0.072	0.080	0.079	0.070
I_{22}	0.290	0.674	0.196	0.053	0.045	0.064	0.048
I_{13}	0.170	0.895	0.152	0.032	0.011	0.031	0.023
I_{23}	0.150	1.000	0.150	0.034	0.011	0.030	0.021
I_{12},I_{32}	0.180	0.783	0.141	0.035	0.013	0.034	0.019
I_{13},I_{23}	0.130	1.000	0.130	0.031	0.007	0.040	0.006
I_{12},I_{22},I_{32}	0.160	0.800	0.128	0.019	0.008	0.021	0.011
I_{12},I_{22}	0.190	0.633	0.120	0.024	0.008	0.030	0.015
I_{33}	0.120	1.000	0.120	0.018	0.012	0.030	0.009

Table 2 The 10 most important rules found by Apriori ($t_{\text{supp}} \geq 0.05, t_{\text{conf}} \geq 0.5$) with their frequencies of appearing in the $N = 1000$ rules generated by multinomial and Bernoulli Gibbs-ARM methods (Simulation 2).

ing $\xi = 70$ and $\xi = 250$ for multinomial and Bernoulli Gibbs-ARM methods respectively, they can find some important association rules even only $N = 100$ rules are generated.

Specifically, the multinomial Gibbs-ARM method found 3 of the top 10 rules from the $N = 100$ generated rules where the top 1 rule (i.e $I_{32} \rightarrow I_{\text{D}}$) has the highest frequency 0.79. It is expected that more of the top 10 rules can be found from the generated rules with higher frequencies if ξ is set smaller. But the involved computation will be more intensive.

The Bernoulli Gibbs-ARM method did not perform as well here in that only the top 1 rule can be found with frequency 0.98 from the $N = 100$ generated rules. This is expected because the items of each SNP have a multinomial constraint and the Bernoulli Gibbs-ARM ignores this constraint.

Control Class For transactions having the control phenotype (I_{ND}), the same methods can be applied to find the most important association rules. For example, in Simulation 3, the phenotype was simulated from logistic regression with coefficients 1, 2, 3 for SNP1, SNP2, and SNP3 respectively. So, we also expect strong associations between certain levels of these SNPs and I_{ND}. The top 10 rules found by Apriori and the Gibbs-ARM results are shown in Table 4. We see the results in Tables 4 are similar to that in Table 3: the multinomial Gibbs-ARM algorithm has found 7 of the top 10 important rules while the Bernoulli one only found 1.

Marginal frequencies From the rules explored by each of the three algorithms, the marginal item frequencies could also indicate the association strength between each SNP item and I_{D} or I_{ND}. Table 5 and Table 6 list the top 10 frequent items for case and control classes, respectively. For the case transactions, I_{32} and I_{22} are among the top 10 frequent items, while for the control transactions, I_{31}, I_{21}, I_{11}, and I_{41} seem to have strong associations with I_{ND}.

From the 3 simulations, we have demonstrated that the multinomial Gibbs-ARM algorithm is able to find the most important rules in all types of scenarios. In complex scenarios, e.g. Simulation 3, the multinomial Gibbs-ARM method could explore a larger number of important rules as compared with the Bernoulli Gibbs-ARM method.

3.2 Real-world Dataset

In a GWAS tutorial paper of Reed et al. [19], they preprocessed the PennCATH dataset and performed GWAS by fitting an additive model for each SNP variable (with sex, age, and principal components to correct the effect of population-substructure). In this section, we will use the same dataset and follow the preprocessing procedure in that tutorial paper, but undertake the association rule mining by the stochastic multinomial Gibbs sampling based method, which allows us to discover the association relationships between the SNPs and the target disease in an efficient way.

3.2.1 Dataset Description

The PennCATH dataset [20] consists of the genotype information of 861,473 SNPs across 3850 individuals for GWAS of coronary artery disease (CAD) and cardiovascular risk factors. A set of 1401 individuals in the dataset were de-identified and selected to be used in the aforementioned tutorial paper: 933 are positive and 468 are negative for CAD. The dataset contains the clinical data, genotypes, and CAD status for each sample.

3.2.2 Preprocessing

In the tutorial paper, 6 steps of preprocessing have been applied to the PennCATH dataset before association analysis:

1. Read PLINK data into R

Rules ($\to I_D$)	Supp	Conf	Imp($\downarrow$)	Multinomial Frequency	Bernoulli Frequency
				$\xi = 70$	$\xi = 250$
I_{32}	0.344	0.815	0.280	0.79	0.98
I_{22}	0.328	0.756	0.248	0.12	0.00
I_{22},I_{32}	0.268	0.893	0.239	0.08	0.00
I_{42}	0.302	0.665	0.201	0.00	0.00
I_{12},I_{32}	0.214	0.907	0.194	0.00	0.00
I_{32},I_{42}	0.240	0.805	0.193	0.00	0.00
I_{52}	0.304	0.626	0.190	0.00	0.00
I_{32},I_{52}	0.220	0.846	0.186	0.00	0.00
I_{12}	0.272	0.663	0.180	0.00	0.00
I_{12},I_{22}	0.228	0.770	0.176	0.00	0.00

Table 3 The 10 most important rules found by Apriori ($t_{\text{supp}} \geq 0.05, t_{\text{conf}} \geq 0.5$) with their frequencies of appearing in the $N = 100$ rules generated by multinomial and Bernoulli Gibbs-ARM methods (Simulation 3 cases).

Rules ($\to I_{ND}$)	Supp	Conf	Imp($\downarrow$)	Multinomial Frequency	Bernoulli Frequency
				$\xi = 65$	$\xi = 240$
I_{31}	0.422	0.917	0.387	0.82	0.98
I_{21}	0.394	0.900	0.354	0.07	0.00
I_{21},I_{31}	0.348	0.972	0.338	0.04	0.00
I_{31},I_{41}	0.328	0.932	0.306	0.02	0.00
I_{11},I_{21}	0.322	0.915	0.295	0.02	0.00
I_{11},I_{31}	0.304	0.962	0.292	0.00	0.00
I_{11}	0.360	0.804	0.289	0.01	0.00
I_{11},I_{21},I_{31}	0.282	0.972	0.274	0.00	0.00
I_{21},I_{41}	0.280	0.972	0.272	0.01	0.00
I_{21},I_{31},I_{41}	0.276	0.979	0.270	0.00	0.00

Table 4 The 10 most important rules found by Apriori ($t_{\text{supp}} \geq 0.05, t_{\text{conf}} \geq 0.5$) with their frequencies of appearing in the $N = 100$ rules generated by multinomial and Bernoulli Gibbs-ARM methods (Simulation 3 controls).

Apriori	item	I_{32}	I_{22}	I_{42}	I_{52}	$I_{29,2}$	I_{12}	$I_{23,2}$	I_{62}	$I_{93,2}$	$I_{16,2}$
	freq	0.0652	0.0613	0.0494	0.0493	0.0425	0.0390	0.0364	0.0364	0.0356	0.0339
Multinomial	item	I_{32}	I_{22}	$I_{100,2}$							
	freq	0.87	0.20	0.01							
Bernoulli	item	I_{32}	$I_{100,3}$								
	freq	0.98	0.02								

Table 5 Marginal frequencies of the 10 most frequent items identified by each algorithm (Simulation 3 cases)

Apriori	item	I_{31}	I_{21}	I_{11}	I_{41}	I_{51}	$I_{72,2}$	$I_{33,2}$	$I_{21,2}$	I_{61}	$I_{32,2}$
	freq	0.0690	0.0639	0.0612	0.0557	0.0418	0.0348	0.0316	0.0310	0.0300	0.0271
Multinomial	item	I_{31}	I_{21}	I_{41}	I_{11}	$I_{100,2}$					
	freq	0.88	0.14	0.03	0.03	0.01					
Bernoulli	item	I_{31}	I_{11}	$I_{88,2}$	$I_{89,2}$	$I_{100,3}$					
	freq	0.99	0.01	0.01	0.01	0.01					

Table 6 Marginal frequencies of the 10 most frequent items identified by each algorithm (Simulation 3 controls)

2. SNP-level filtering (part 1)
3. Sample-level filtering
4. SNP-level filtering (part 2)
5. Create principal components to capture population-substructure
6. Impute non-typed SNP with external data

The population-substructure and SNP imputation are irrelevant for Gibbs sampling based ARM, therefore, we only follow the first 4 steps to filter SNP variables and samples with R.

The program first convert the PLINK data (.bed, .bim, and .fam files) into an object in R. Then, it filters the SNPs based on their call rate, minor allele frequency (MAF), and filtered the samples based on their call rate, heterozygosity, relatedness, and ancestry. Finally, it filters the SNPs based on Hardy–Weinberg equilibrium (HWE) and this preprocessing results in 1401 samples (no individuals filtered out) with 656,890 SNP variables.

3.2.3 Reducing item space

To speed up computation, further preprocessing could be done to reduce the item space. This includes two

steps: select a range of SNPs potentially related to CAD based on historical analysis results, and filter SNPs with low support.

Chromo9p21 Research on GWAS of CAD has achieved substantial progress, indicating some significant associations between SNPs on Chromosome 9p21 and CAD [11,21,5]. In this paper, we use the SNPs on Chromosome 9p21 and expect to find some strong association rules there. According to Genome Reference Consortium GRCh37 (hg19), SNPs on Chromosome 9p21 have positions from 19900000 to 33200000 on Chromosome 9. This information can be used to extract the relevant SNPs with R, and it gives us 3758 SNP variables.

Filtering SNPs As the support of a rule can never exceed the minimum support of its contained items, if we want to generate rules with support $\geq t_{\text{supp}}$, we can first filter out the items with support lower than t_{supp}. Since each rule from the GWAS transactions data is of the form given in (1) and (2), we are only able to filter out itemsets of form $\mathbf{J}^{(\text{SNP})} = (J_{\text{no}}^{(\text{SNP})}, J_0^{(\text{SNP})}, J_1^{(\text{SNP})}, J_2^{(\text{SNP})})$ with their support $< t_{\text{supp}}$. Table 7 lists the number of SNPs remained after filtering out the SNP itemsets with support less than various t_{supp} values. It suggests $t_{\text{supp}} = 0.4$ and 0.3 are good choices for consequent case $CAD = 1$ and $CAD = 0$ respectively, which ends up with 1948 SNPs left for $CAD = 1$ and 1950 SNPs for $CAD = 0$, a roughly 50% reduction from 3758 SNPs.

Since each SNP is expressed by 3 items, so the two transaction datasets (for $CAD = 1$ and $CAD = 0$) would have $1948 \times 3 = 5844$ and $1950 \times 3 = 5850$ SNP items involved, respectively.

t_{supp}	num of SNPs	
	$CAD = 1$	$CAD = 0$
0.1	3758	3758
0.2	3758	3758
0.3	3604	**1950**
0.4	**1948**	384
0.5	1204	0
0.6	384	0
0.7	0	0
0.8	0	0
0.9	0	0

Table 7 Numbers of remaining SNPs for each consequent case after filtering with different t_{supp} values

3.2.4 Tuning parameter

In our experiments, the importance of an association rule is defined as the product of its support and confidence. To mine the most important association rules from the two aforementioned transaction datasets (for $CAD = 1$ and $CAD = 0$ cases), we have used the multivariate Gibbs-ARM algorithm to generate 100 rules with various given tuning parameter ξ values for each case. Results on the number of distinct rules generated in the $CAD = 1$ case are summarized in Table 8, from which we can see the effect of ξ on rule generations.

Taking the results of positive cases with $CAD = 1$ as an example. We can see that there are three kinds of ξ: maybe too low, maybe too high, and seemingly appropriate. For $\xi = 300$ and $\xi = 500$, the tuning parameter is probably too small, so that the algorithm almost always generates different rules at different times. On the other hand, with $\xi = 700$ and $\xi = 1000$ it only generates 11 and 2 distinct rules respectively, suggesting that the tuning parameter value is probably too high so that rule generation process tends to be trapped in a neighbourhood having locally high importance rules. Consequently, if we are interested in finding the top k rules instead of the top 1, we will need more distinct rules to be generated by the multivariate Gibbs-ARM algorithm. For the current datasets, $\xi = 550$ or $\xi = 600$ might be the best options, knowing that the final choice would be depending on the needs. They give 40 or 18 distinct rules, implying it has capacity to find some high importance rules from the generated samples without getting stuck into a local maxima neighbourhood. For the rest of the experiments, $\xi = 550$ is chosen to generate rules for the $CAD = 1$ cases.

ξ	# distinct rules	Comment
300	100	Too low
500	68	Too low
550	40	Appropriate
600	18	Appropriate
700	11	Too high
1000	2	Too high

Table 8 Number of distinct rules generated with different ξ for $CAD = 1$

3.2.5 Rule generation

Table 9 displays the top 10 important rules generated with $\xi = 550$ for $CAD = 1$ cases. From the table, we can see that the frequencies of these rules are ordered similarly as their importance values, and the rule with the highest importance coincide with the most frequent rule in the sample: $(\text{rs41474551} = 2) \rightarrow I_{\text{CAD}=1}$. Moreover, all the SNPs we found in the rules are at level 2, which means that individuals having 2 minor alleles at those important SNPs may have higher risk of the disease. This actually aligns with what we have ex-

pected from the genetic point of view since minor alleles are deemed to be more likely the risk alleles in GWAS. Therefore, we can conclude that the multivariate Gibbs-ARM algorithm worked well by confirming this biological understanding.

3.2.6 Rules associated with negative class

For the negative class $CAD = 0$, we have tried $\xi = 550$ to generate 100 rules using the multivariate Gibbs-ARM algorithm and found they are all distinct rules from each other and have very low importance values. Thus, using $\xi = 550$ did not help us find important rules. By running more tests and increasing ξ up to 2500, the algorithm is finally able to distinguish some interesting rules from the rest, and ends up with 30 distinct rules. The top 10 important rules are listed in Table 10. From the table, we can see that the importance values of negative class rules are much lower than the positive class rules. This is because of the imbalanced labels in the dataset. There are only 468 negative ($CAD = 0$) transactions/individuals and the support of any rules would not exceed $468/1401 = 0.3340$. The confidence of the negative rules are also lower than the positive ones, which means SNPs generally have weak associations with $CAD = 0$. Nevertheless, our algorithm is still capable of finding rules with a relatively high support, 0.3291 which is only slightly lower than the largest possible value, 0.3340. Also, 46% of the generated rules are among the top 10 important rules in the dataset and the most important rule has the highest frequency. This demonstrates the good performance of the algorithm for the negative class as well.

3.2.7 Item frequencies

Top 10 marginal frequencies of those items generated by the multinomial Gibbs-ARM sampling algorithm for both CAD = 1 and CAD = 0 cases are listed in Table 11. Similar to that said for the simulation examples, the marginal frequencies of the items could also be used to assess the associations between the items and the disease. In addition, the items having high frequencies in the generated samples provide a reduced item space to find important rules by using the Apriori algorithm. For example, for the $CAD = 1$ class, there are 20 items appearing more than once in the generated samples and these items, constituting a new item space, could be used to find association rules by the Apriori algorithm. The Gibbs-ARM sampling method used here is largely for reducing the itemset space so that the computation cost from applying the Apriori to this new itemset space is also reduced.

4 Conclusion

In this paper, we proposed a stochastic multinomial Gibbs sampling based association rule mining algorithm to analyze transactions data obtained from GWAS. In the experiments, we have tested the algorithm on 3 simulated datasets and a real-world dataset. The algorithm has shown good performance in all our experiments and case study. In the real-world example, we have considered 1948 SNPs (5844 items) for the positive class and 1950 SNPs (5850 items) for the negative class, which is significantly more than 366 SNPs in the dataset used by Qian et al. [17]. This has demonstrated the capacity of the algorithm in big data ARM scenarios. Also, the multinomial Gibbs-ARM sampling algorithm substantially extends the capacity of the Bernoulli Gibbs-ARM algorithm in that the former in capable of incorporating the constraints in the itemsets to more accurately and efficiently generate a Markov chain of the rules.

The application of our method is not limited to finding the important SNPs related to certain disease in GWAS. It can be used to find the associations between various combinations of the categorical variables with different numbers of levels and a certain level of a response variable in any study field.

Acknowledgements Pei-Yun Sun's research in this paper was supported by the University of Melbourne Graduate Research Training Scholarship.

Author contribution

Guoqi Qian: Conceptualization, Methodology, Validation, Formal Analysis, Investigation, Writing - Original Draft, Writing - Review and Editing, Supervision. **Pei-Yun Sun:** Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Agapito, G., Guzzi, P.H., Cannataro, M.: An efficient and scalable spark preprocessing methodology for genome wide association studies. In: 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp. 369–375. IEEE (2020)

Rules $\rightarrow I_{CAD=1}$	**Supp**	**Conf**	**Imp(↓)**	**Freq**
rs41474551=2	0.6545	0.6684	0.4375	0.04
rs6476155=2	0.6531	0.6689	0.4368	0.01
rs10968275=2	0.6510	0.6691	0.4356	0.02
rs10968147=2, rs12000381=2	0.6495	0.6681	0.4340	0.01
rs12000381=2, rs12005211=2	0.6495	0.6681	0.4340	0.01
rs12353389=2	0.6510	0.6667	0.4340	0.01
rs10968147=2, rs12005211=2	0.6495	0.6672	0.4333	0.01
rs10969920=2, rs7868409=2	0.6488	0.6664	0.4324	0.01
rs12337979=2	0.6488	0.6659	0.4321	0.01
rs12000381=2, rs6476155=2	0.6417	0.6724	0.4315	0.01

Table 9 Top 10 important rules associated with positive class $CAD = 1$ ($\xi = 550$)

Rules $\rightarrow I_{CAD=0}$	**Supp**	**Conf**	**Imp(↓)**	**Freq**
rs12340615=2	0.3291	0.3377	0.1111	0.11
rs10481580=2	0.3283	0.3377	0.1109	0.03
rs17780713=2	0.3233	0.3429	0.1109	0.06
rs7032605=2	0.3283	0.3377	0.1109	0.04
rs10481580=2, rs12340615=2	0.3283	0.3377	0.1109	0.07
rs12340615=2, rs12345834=2	0.3283	0.3375	0.1108	0.06
rs10481580=2, rs12345834=2	0.3276	0.3375	0.1106	0.01
rs12340615=2, rs17780713=2	0.3191	0.3468	0.1106	0.04
rs10481580=2, rs12340615=2, rs17780713=2	0.3183	0.3468	0.1104	0.03
rs10481580=2, rs17780713=2	0.3183	0.3468	0.1104	0.01

Table 10 Top 10 important rules associated with negative class $CAD = 0$ ($\xi = 2500$)

$CAD = 1$		$CAD = 0$	
item	**freq**	**item**	**freq**
rs10967419=2	0.12	rs12340615=2	0.39
rs7045427=2	0.12	rs10481580=2	0.24
rs2094534=2	0.10	rs17780713=2	0.20
rs12000381=2	0.09	rs12345834=2	0.17
rs12005211=2	0.08	rs7032605=2	0.12
rs10967389=2	0.07	rs17759490=2	0.04
rs10967430=2	0.07	rs7020500=2	0.04
rs10812419=2	0.07	rs16907723=2	0.03
rs41474551=2	0.07	rs16923583=2	0.03
rs2011765=2	0.07	rs2453553=2	0.03

Table 11 Top 10 marginal frequencies of the generated SNP items by the multinomial Gibbs-ARM sampling method for both cases.

2. Agapito, G., Guzzi, P.H., Cannataro, M.: Parallel and distributed association rule mining in life science: A novel parallel algorithm to mine genomics data. Information Sciences **575**, 747–761 (2021)
3. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 207–216 (1993)
4. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
5. Chen, Z., Qian, Q., Ma, G., Wang, J., Zhang, X., Feng, Y., Shen, C., Yao, Y.: A common variant on chromosome 9p21 affects the risk of early-onset coronary artery disease. Molecular biology reports **36**(5), 889 (2009)
6. Cho, S., Kim, H., Oh, S., Kim, K., Park, T.: Elastic-net regularization approaches for genome-wide association studies of rheumatoid arthritis. In: BMC proceedings, vol. 3, pp. 1–6. BioMed Central (2009)
7. Fan, J., Lv, J.: Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **70**(5), 849–911 (2008)
8. Florian Hébert Mathieu Emily, D.C.: Simulation of genotypic profiles and binary phenotypes for GWASs (2019). URL https://github.com/fhebert/SNPSetSimulations/
9. Hahsler, M., Grün, K., Hornik, K.: arules – a computational environment for mining association rules and frequent item sets. Journal of Statistical Software **14**(15), 1–25 (2005)
10. He, Q., Lin, D.Y.: A variable selection method for genome-wide association studies. Bioinformatics **27**(1), 1–8 (2011)
11. Jarinova, O., Stewart, A.F., Roberts, R., Wells, G., Lau, P., Naing, T., Buerki, C., McLean, B.W., Cook, R.C., Parker, J.S., et al.: Functional analysis of the chromosome 9p21. 3 coronary artery disease risk locus. Arteriosclerosis, thrombosis, and vascular biology **29**(10), 1671–1677 (2009)
12. Johnstone, I.M., Titterington, D.M.: Statistical challenges of high-dimensional data (2009)
13. Lewis, C.M.: Genetic association studies: design, analysis and interpretation. Briefings in bioinformatics **3**(2), 146–153 (2002)
14. Li, J., Zhong, W., Li, R., Wu, R.: A fast algorithm for detecting gene–gene interactions in genome-wide association studies. The annals of applied statistics **8**(4), 2292 (2014)
15. Li, M., Liu, X., Bradbury, P., Yu, J., Zhang, Y.M., Todhunter, R.J., Buckler, E.S., Zhang, Z.: Enrichment of statistical power for genome-wide association studies. BMC biology **12**(1), 1–10 (2014)
16. Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A., Reich, D.: Principal components analysis corrects for stratification in genome-wide association studies. Nature genetics **38**(8), 904–909 (2006)
17. Qian, G., Rao, C.R., Sun, X., Wu, Y.: Boosting association rule mining in large datasets via gibbs sampling. Pro-

ceedings of the National Academy of Sciences **113**(18), 4958–4963 (2016)
18. Qian, G., Zhao, X.: On time series model selection involving many candidate arma models. Computational Statistics & Data Analysis **51**(12), 6180–6196 (2007)
19. Reed, E., Nunez, S., Kulp, D., Qian, J., Reilly, M.P., Foulkes, A.S.: A guide to genome-wide association analysis and post-analytic interrogation. Statistics in medicine **34**(28), 3769–3792 (2015)
20. Reilly, M., Li, M., He, J., Ferguson, J., Stylianou, I., Mehta, N., Burnett, M., Devaney, J., Knouff, C., Thompson, J., et al.: Identification of adamts7 as a novel locus for coronary atherosclerosis and association of abo with myocardial infarction in the presence of coronary atherosclerosis: two genome-wide association studies. Lancet **377**(9763), 383–392 (2011). DOI doi.org/10.1016/S0140-6736(10)61996-4
21. Shen, G.Q., Li, L., Rao, S., Abdullah, K.G., Ban, J.M., Lee, B.S., Park, J.E., Wang, Q.K.: Four snps on chromosome 9p21 in a south korean population implicate a genetic locus that confers high cross-race risk for development of coronary artery disease. Arteriosclerosis, thrombosis, and vascular biology **28**(2), 360–365 (2008)
22. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (1996)
23. Ueki, M., Tamiya, G.: Ultrahigh-dimensional variable selection method for whole-genome gene-gene interaction analysis. BMC bioinformatics **13**(1), 1–15 (2012)
24. Uffelmann, E., Huang, Q.Q., Munung, N.S., de Vries, J., Okada, Y., Martin, A.R., Martin, H.C., Lappalainen, T., Posthuma, D.: Genome-wide association studies. Nature Reviews Methods Primers **1**, Article number: 59 (2021). DOI https://doi.org/10.1038/s43586-021-00056-9
25. Wang, M.H., Cordell, H.J., Van Steen, K.: Statistical methods for genome-wide association studies. Seminars in Cancer Biology **55**, 53–60 (2019). DOI https://doi.org/10.1016/j.semcancer.2018.04.008
26. Wu, T.T., Chen, Y.F., Hastie, T., Sobel, E., Lange, K.: Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics **25**(6), 714–721 (2009)
27. Yu, J., Pressoir, G., Briggs, W.H., Vroh Bi, I., Yamasaki, M., Doebley, J.F., McMullen, M.D., Gaut, B.S., Nielsen, D.M., Holland, J.B., et al.: A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. Nature genetics **38**(2), 203–208 (2006)
28. Zhang, Z., Ersoz, E., Lai, C.Q., Todhunter, R.J., Tiwari, H.K., Gore, M.A., Bradbury, P.J., Yu, J., Arnett, D.K., Ordovas, J.M., et al.: Mixed linear model approach adapted for genome-wide association studies. Nature genetics **42**(4), 355–360 (2010)
29. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology) **67**(2), 301–320 (2005)