Research on Course Scheduling Optimization Algorithm Based on Inter-Course Association Rules: Postprint

Authors: Pang Shujie, Liu Qicheng, Mou Chunxiao

Date: 2018-07-09T00:00:00+00:00

Abstract

This study investigates the phenomenon of strained course scheduling resources in universities resulting from the current expansion of enrollment, and proposes an association rule-based course scheduling optimization algorithm (SH-AP algorithm) to address this issue. The SH-AP algorithm integrates association rule mining into the scheduling process by extracting association rules from university student course selection data to identify relationships among courses requiring scheduling. These mined inter-course association rules are then applied to the scheduling system to optimize university course scheduling. Experimental results yield data on inter-course association rules and enable humanized course scheduling for universities. Comparative studies demonstrate that the SH-AP algorithm effectively optimizes both the rationality and humanization of course scheduling, assisting institutions in accommodating student needs while simultaneously addressing time and location conflicts.

Full Text

Preamble

Title: Research on Timetabling Optimization Algorithm Based on Inter-Curriculum Association Rules

Authors: Pang Shujie, Liu Qicheng†, Mu Chunxiao
Affiliation: School of Computer & Control Engineering, Yantai University, Yantai, Shandong 264005, China

Abstract: This paper addresses the resource constraints in university timetabling resulting from enrollment expansion by proposing an optimization algorithm based on association rules (SH-AP algorithm). The SH-AP algorithm applies association rule mining to the timetabling process by extracting inter-curriculum association patterns from student course selection data, then leverages these patterns to optimize university scheduling. Experimental results yield inter-curriculum association rules that enable more humanized timetabling. Comparative studies demonstrate that the SH-AP algorithm effectively enhances both the rationality and humanization of scheduling, helping institutions avoid time and location conflicts while satisfying student preferences.

Keywords: student course selection; university timetabling; association rules

0 Introduction

With rising education levels, increasing numbers of students are entering universities, leading to substantial enrollment growth across Chinese higher education institutions. This surge has created severe strain on teaching resources [1]. Contemporary university students seek comprehensive development beyond their major requirements, with each student pursuing different extension directions based on their professional foundation. Consequently, universities offer diverse elective courses spanning liberal arts, sciences, and engineering in addition to mandatory major courses [2].

The establishment of elective systems grants students greater autonomy in shaping their academic trajectories while generating vast amounts of data that reveal relationships among preferred courses. However, electives also intensify timetabling pressures. A critical challenge facing universities is how to schedule courses rationally in time and space while accommodating student preferences and preventing conflicts among popular course combinations [3].

Simultaneously, as big data serves the common goals of education and science, educational data mining technologies have advanced rapidly. Techniques such as association rules, classification, and clustering have been flexibly applied to educational data. Mining university course selection data can uncover valuable inter-curriculum association patterns that administrators can leverage to optimize timetabling and avoid scheduling conflicts.

Association rule algorithms constitute a crucial component of data mining, with the Apriori algorithm ranking among the ten most classic algorithms and representing one of the most influential [4]. However, Apriori generates massive candidate itemsets and incurs substantial I/O overhead during rule mining [5]. The DHP (Direct Hashing and Pruning) algorithm improves upon Apriori by optimizing candidate itemset screening, reducing database access and I/O costs.

1 Related Work

Association rule mining remains a vital data mining component since its inception, with diverse algorithms proposed and widely applied across finance, retail, and predictive monitoring [6]. Current research includes: application of association rules to meteorological observation equipment for data collection stability [7]; analysis of library recommendation systems using association rule mining on borrowing data [8]; mining electronic medical records through improved association rule algorithms to assist diagnosis [9]; and processing food production data for safety diagnosis and early warning [10]. These studies demonstrate varied applications of association rule mining.

University timetabling represents a primary task for institutions and students alike, crucial for promoting holistic student development. Existing research includes: a credit system-based computer timetabling model centered on courses [11]; a 教务 management system improving accuracy and timeliness amid growing student and course numbers [12]; and a microcomputer timetabling system using relational databases to advance educational management informatization [13]. However, [11] focuses solely on credit impacts while neglecting student subjectivity, whereas [12] and [13] address institutional arrangements without considering student preferences.

To address these limitations, this paper proposes the SH-AP algorithm (Scheduling optimization algorithm based on Hashing and Association rules for Pruning), which analyzes student course selection data to further optimize university timetabling by accommodating both institutional requirements and student preferences.

2 SH-AP Algorithm Design

The proposed SH-AP algorithm comprises three main components: (1) preprocessing student course selection data as a training set $D = \{T_1, T_2, T_3, \dots, T_n\}$ where each transaction T_n represents a student ID with its itemset containing selected courses, and $I = \{I_1, I_2, I_3, I_4, \dots, I_n\}$ denotes the available courses (including 11 subjects such as Computer Introduction, Java Object-Oriented Programming, and Microcontroller Principles); (2) mining inter-curriculum association rules from the training data to obtain support and confidence measures; and (3) analyzing mined rules with an additional lift threshold, sorting curriculum relationships by lift value, and feeding results to administrators for optimization.

2.1 DHP Algorithm Introduction

Association rules are determined by two thresholds: minimum support and minimum confidence. Support represents an itemset's frequency in the database, while confidence is a conditional probability derived from support values [14]. Traditional Apriori generates candidate k-itemsets by joining frequent $(k-1)$-itemsets, then prunes candidates by scanning the database and counting support. This process iterates until no new frequent itemsets emerge, requiring multiple database scans and substantial I/O overhead.

For example, 10^4 frequent 1-itemsets could generate 10^7 candidate 2-itemsets, necessitating numerous database scans. DHP improves this by reducing candidate k-itemset counts during Apriori' s pruning phase through four steps:

1. **Database Scanning**: Scan database D to obtain $D = \{T_1, T_2, T_3, \dots, T_n\}$ where each transaction T_i is a subset of itemset I.
2. **Hash Table Construction**: During the k-th scan, generate $(k+1)$-itemsets from each transaction, hash them into buckets using a Hash function, and count bucket elements.
3. **Candidate Itemset Filtering**: If a bucket' s count falls below minimum support, all itemsets hashing to that bucket cannot be frequent and are eliminated from candidates.
4. **Frequent Itemset Generation**: Scan the database with remaining candidate itemsets C_{k+1}, count support, and add those meeting minimum support to frequent itemset L_{k+1}. Repeat until no new candidates or frequent itemsets emerge.

2.2 SH-AP Algorithm Design

The SH-AP algorithm integrates DHP for university timetabling optimization, reducing candidate itemsets and I/O overhead compared to traditional Apriori. It mines historical data to minimize scheduling conflicts.

2.2.1 Student Course Association Rule Mining The mining phase processes student selection data where each transaction contains a student' s chosen courses from project set $I = \{I_1, I_2, I_3, I_4, \dots, I_n\}$. The algorithm executes with minimum support minsup ≥ 0.2 (requiring course combinations appear in at least 20% of transactions).

First, scan the database to generate candidate 1-itemsets $C_1 = \{\{I_1\}, \{I_2\}, \{I_3\}, \{I_4\}, \{I_5\}\}$ and count support to obtain frequent 1-itemsets L_1. Then combine each student' s selected courses to generate personal candidate 2-itemsets. The algorithm proceeds as follows:

```
Input: Student data
Output: Lk
for each Student Data do
    Cdhp = {Courses for each student}
    L1 = {each course   Cdhp}
    L2 = {each course   Cdhp}
    if ((L1[k-1] < L2[k-1]) && L1[n] = L2[n])
        /*n = {1,2,3,⋯,k-2}*/
        c = L1 connect L2
        HashFunction(c)
```

Each student' s candidate itemsets c_k are hashed: course combinations generate hash values determining storage buckets. The algorithm counts elements per bucket and retains only those meeting minimum support:

```
for each ck   each student
    {x,y}   Ck    /*course combination*/
    mod 7
    HashB[HashValue][1] add 1

for each L1$\times$L1 {x,y}   L1$\times$L1
    mod 7
    If(HashB[HashValue][1] >= minsup)
        {x,y} add to Ck add
```

Traditional Apriori generates enormous candidate course sets, requiring a database scan per candidate and causing massive I/O overhead. SH-AP reduces candidates by filtering through Hash buckets before counting, significantly decreasing database scans and I/O costs.

2.2.2 Timetable Optimization Based on Association Rules After mining association rules with support and confidence, SH-AP introduces **lift** as an additional threshold to determine scheduling priority—higher lift indicates stronger association.

The algorithm reads initial timetable database D_1 containing Monday-to-Friday scheduling data, where $X_i = \{T_1, T_2, \ldots, T_5\}$ represents daily periods, $T_n = \{i | i \in I\}$ denotes courses scheduled in period T_n, and $C_i = \{(\text{course}, \text{Lift}) | \text{course} \in I\}$ contains courses and their lift values relative to course I_i. Each course carries a star marker indicating mandatory (star=1) or elective (star=0) status.

The optimization algorithm proceeds as:

```
Input: Timetable data D1
Output: Schedule D2
for each Ix   I    /*Ix   Tn (n=1,2,..,5)*/
    mix = 0
    for each Lift   Cx
        if(Lift < mix) mix = Lift
    for i <= Cx.size
        if i.lift == mix
            c = i.course
            if c   Tn
                for each T   D //T is one period
                    if T != Tn
                        for each Ij   T
                            if Ij.star <= c.star
                                note = c
                                for each Cj /*prevent error replacement*/
                                    if(Cj.lift >= mix)
```

```
                        if(Cj.course   Tn) continue
                        x = c; c = Ij; Ij = x; break
                if note == c break
```

This procedure optimizes the timetable by identifying the highest-lift course c in C_x, finding replaceable courses in other periods, and swapping them to avoid conflicts among highly-associated courses.

3 Experiments

3.1 Experimental Data

The experiment uses undergraduate course selection data from a specific school in a Chinese university, including four years of general electives and major electives across all student directions to ensure comprehensive coverage. The dataset comprises 527 transactions, each representing a student' s complete course selections during their undergraduate studies. The itemset includes 11 courses: Computer Introduction, Database Principles, Art Appreciation, Film Appreciation, Java Object-Oriented Programming, Android Development, Flash Animation, Computer Graphics, Machine Vision, Microcontroller Principles, and Embedded Systems. During preprocessing, courses were replaced with numeric codes.

3.2 Evaluation Criteria

Association rule reliability and practicality are measured by support and confidence. For university timetabling, three metrics are required:

1. **Support**: The frequency of an itemset in the total dataset D, calculated as the ratio of transactions containing item X to the total transaction count.
$$support(X) = P(X) = \frac{|\{T \in D | X \in T\}|}{|D|} \times 100$$

2. **Confidence**: The conditional probability that itemset Y appears given X appears, derived from support values.
$$confidence(X \to Y) = \frac{support(X \cup Y)}{support(X)} \times 100$$

3. **Lift**: The ratio of the proportion selecting both A and B to the proportion selecting B overall.
$$lift(A \to B) = \frac{conf(A \to B)}{support(B)} = \frac{P(A \cap B)}{P(A)P(B)}$$

An association rule is considered credible and valuable when it satisfies minimum support, minimum confidence, and lift > 1 thresholds.

3.3 Results Analysis

Table 1 shows sample preprocessed student selection data where courses are represented numerically (e.g., "4,9,10,11").

Table 1: Data Preprocessing

4,9,10,11
2,5,6,7
1,3,4,6,7,10,11
2,4,5,6,11
2,5,6,8,11
1,2,3,6,10,11
1,3,8,11
3,4,5

These processed data are fed into the SH-AP algorithm with minimum support and confidence thresholds. Table 2 presents the mined association rules with their support, confidence, and lift values.

Table 2: Experimental Results

Association Rule	Support	Confidence	Lift
{2,6} 5	0.23	0.78	1.52
{5,6} 2	0.25	0.81	1.48
{8,2} 5	0.21	0.75	1.65
{8,5} 2	0.22	0.77	1.58
{2,5,6} 8	0.24	0.73	1.42

The results show strong association rules with support > 0.2, confidence > 0.7, and lift > 1. Courses 2, 5, 6, and 8 correspond to Database Principles, Java Object-Oriented Programming, Android Development, and Computer Graphics respectively. These courses exhibit strong co-selection patterns. For optimization, courses with highest lift (e.g., 8 and 2) should be prioritized to avoid temporal and spatial conflicts, followed by others in descending lift order.

3.4 Comparison with Related Work

SH-AP is compared against Apriori-based timetabling and the F2AM algorithm [15], a recent association rule method. Experiments run on an Intel Core i7-7700 (3.60 GHz) with 8 GB RAM, using Java 1.8.0 in Eclipse.

Figure 1 [Figure 1: see original paper] shows runtime comparisons across different minimum support values. At low support thresholds, the three algorithms exhibit significant performance differences: F2AM outperforms Apriori, while SH-AP substantially outperforms both. As support increases, iteration counts and itemset numbers decrease, reducing overhead for all methods, yet SH-AP maintains the lowest runtime.

Figure 2 [Figure 2: see original paper] compares runtimes at fixed minimum support with varying minimum confidence. Runtime fluctuations are minimal across confidence levels because confidence computation relies on precalculated support counts. While F2AM improves upon Apriori, SH-AP demonstrates clear superiority in all scenarios.

These comparisons confirm SH-AP' s efficiency in mining student selection data and its effectiveness for inter-curriculum association analysis.

4 Conclusion

This paper proposes the SH-AP algorithm for university timetabling optimization based on inter-curriculum association rules. By mining student selection data, the algorithm discovers curriculum relationships through lift values and integrates them into the timetabling process. This approach simultaneously avoids scheduling conflicts and accommodates student preferences, yielding more rational and humanized timetables.

With continued enrollment expansion and new course offerings generating ever-larger datasets, serial processing becomes inadequate. Future work will parallelize the association rule algorithm to handle massive data volumes and reduce mining time.

References

[1] Yao S L. Application and research on correlation between colleges courses based on data mining[J]. Bulletin of Science and Technology, 2012, 28(12): 232-234.

[2] Jiang Y C. Research on students course selection and learning behavior analysis algorithm based on data mining[J]. Modern Electronics Technique, 2016, 39(13): 145-148.

[3] Wang N Q, Yao S G. Curriculum scheduling based on improved swarm optimization algorithm[J]. Journal of Computer Applications, 2013, 33(1): 207-210.

[4] Du X P, Ma X L, Tang S W, et al. A fast algorithm for mining of association rules[J]. Computer Engineering and Applications, 2002, 38(11): 1-4.

[5] Yan M J, Luo J, Liu J Y, et al. IABS: Parallel improved apriori based on spark[J]. Application Research of Computers, 2017, 34(8): 2274-2277.

[6] Cui G X, Li L, Wang K K, et al. Research and improvement on apriori algorithm of association rule mining[J]. Journal of Computer Applications, 2010, 30(11): 2952-2955.

[7] Li T, Yu M C, Lu Z B, et al. Consistency detection algorithm for meteorological observation equipment based on association rules mining[J]. Journal of Electronic Measurement and Instrumentation, 2017, 31(10): 1568-1573.

[8] Chen D Q, Zhu W F. Association rules and library bibliographic recommendation[J]. Information Studies: Theory & Application, 2009, 32(6): 81-84.

[9] Ding W P, Qi H, Dong J C, et al. Research and application of the algorithms of the quick and effective date mining based on the association rules[J]. Microelectronics & Computer, 2007, 24(3): 69-73.

[10] Gu X L, Zhang D W, Zhang K, et al. The information Pre-warning model of food safety based on association rules mining[J]. Soft Science, 2011, 25(11): 136-141.

[11] Gao H Y, Liang Z H, Sun X P. The designing of computer courses arrangement system centralized on courses under the credit system[J]. Computer Engineering and Applications, 2005, 41(31): 212-214.

[12] Zhang J A, Yang X J, Wu W Y. Design and implementation of online course arrangement system based on association rule algorithm[J]. Modern Electronics Technique, 2010, 33(2): 60-64.

[13] Xiu L, Yang S T. The design and implementation of the management system of the course arrangement in colleges and universities[J]. Journal of Taiyuan University of Technology, 2006(S1): 75-77.

[14] Cui Y, Bao Z Q. Survey of association rule mining[J]. Application Research of Computers, 2016, 33(02): 330-334.

[15] Ji H M. Improved apriori algorithm based on frequency 2-item set support matrix[J]. Computer Engineering, 2013, 39(11): 183-186.

Note: Figure translations are in progress. See original paper for figures.