UPM: Discovering Course Enrollment Sequences Associated with Success

Asmaa Elbadrawy
Information Technology
Arizona State University

George Karypis
Computer Science
University of Minnesota

ABSTRACT

Identifying enrollment patterns associated with course success can help educators design better degree plans, and students make informed decisions about future enrollments. While discriminating pattern mining techniques can be used to address this problem, course enrollment patterns include sequence and quantity (grades) information. None of the existing methods were designed to account for both factors. In this work we present UPM, a **U**niversal discriminating **P**attern **M**ining framework that simultaneously mines various types of enrollment patterns while accounting for sequence and quantity using an expansion-specific approach. Unlike the existing methods, UPM expands a given pattern with an item by finding a minimum-entropy split over the item's quantities. We then use UPM to extract discriminating enrollment patterns from the high and the low performing student groups. These patterns can be utilized by educators for degree planning. To evaluate the quality of the extracted patterns, we adopt a supervised classification approach where we apply various classification techniques to label students according tho their performance based on the extracted patterns. Our evaluation shows that the classification accuracies obtained using the UPM extracted patterns are higher than the accuracies obtained using patterns extracted by other techniques. Accuracy improves significantly for students with larger numbers of patterns. Moreover, expansion-specific quantitative mining leads to more accurate classifications than the methods that do not account for quantities (grades).

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Student Enrollment Sequences, Discriminative Sequence Analysis

ACM Reference Format:
Asmaa Elbadrawy and George Karypis. 2019. UPM: Discovering Course Enrollment Sequences Associated with Success. In *The 9th International Learning Analytics & Knowledge Conference (LAK19), March 4–8, 2019, Tempe, AZ, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3303772.3303799

LAK19, March 4–8, 2019, Tempe, AZ, USA

ACM ISBN 978-1-4503-6256-6/19/03…$15.00
https://doi.org/10.1145/3303772.3303799

1 INTRODUCTION

Due to the flexibility of the degree requirements, a student can enroll in different course sequences before they graduate. Ample course choices can leave students confused, take unncessary courses, or even drop out. One strategy to promote student timely graduation is to design good degree plans that show course enrollments with particular sequences that would lead to successful learning outcomes as measured by the student's GPA. The question is how to find the enrollment practices that are associated with success in a future courses.

We address the problem of finding the enrollment practices associated with high and low performance in a course. For a student that has taken that course, her enrollments in previous courses over successive terms, along with her grades, can be viewed as a sequence of quantitative itemsets as shown in Table 1. From among prior enrollments of all students that took a target course, we want to find the enrollment patterns that are associated with success and failure. To do so, first, we extract two student groups: high- versus low-performing. Second, we mine the prior course enrollments of both student groups to extract discriminating patterns.

Existing discriminating pattern mining methods [2] [4] [15] [7] [6] [12] [3] can either extract discriminating itemsets or discriminating item sequences. None of them can effectively extract discriminating quantitative patterns and so, they cannot account for students grades in previous courses. We present UPM, a universal discriminating pattern mining technique. UPM extracts discriminating patterns of different types: itemsets, item sequences, quantitative itemsets and quantitative item sequences. It uses a feature-centric approach that extracts the most discriminating pattern, excludes instances covered by that pattern, and repeats until the dataset is covered. For discriminating quantitative patterns, UPM accounts for item quantities in the pattern expansion step by finding a minimum-entropy split over the quantities of the added item to the expanded pattern. This novel approach makes the quantity split conditional on the pattern being expanded, not a static one-time split applied once prior to mining.

We evaluate UPM over many course datasets using a supervised approach. For each course, we extract discriminating enrollment patterns from the groups of high and low performing students. To evaluate the extracted patterns, we use them as features to represent the students, then build a classifier to label students according to performance. We use various types classifiers, including HARMONY, SVM and Random Forests. We Also build another set of classifiers where the students are represented using the universal set of enrollment patterns without extracting discriminating ones, or using the small set of courses that each student enrolled in. We do this to evaluate how well the UPM extracted patterns are compared to the whole set of enrollment patterns, and simple patterns. Our evaluation shows that classification accuracies obtained using UPM patterns are higher than accuracies obtained using the other patterns for most courses. Classification accuracy improves for students represented with larger numbers of patterns. For these students, the extracted patterns can be used with high confidence to make future enrollment recommendations or degree plan modifications.

By analyzing the patterns extracted by UPM, we find that all pattern types are extracted, with more itemsets than other pattern types. In many cases, the number of quantitative item sequences is larger than the number of item sequences. The learned HARMONY and linear SVM models can be used to more specifically identify the enrollments associated with high and low performance. Finally, when evaluating the quantitative pattern mining technique, we find the classification accuracies obtained when using itemsets is less than the accuracies obtained when using quantitative itemsets. The same applies for item sequences versus quantitative item sequences, showing the importance of considering student grades.

Table 1: Students' course enrollments viewed as a sequence of quantitative itemsets over the successive terms, where each course c_i is associated with the grade g_i that the student has obtained in it.

	term_1	term_2	term_3
student_1	$c_1{:}g_1, c_5{:}g_5, c_4{:}g_4$	$c_3{:}g_3, c_7{:}g_7$	$c_2{:}g_2, c_6{:}g_6$
student_2	$c_5{:}g_5, c_2{:}g_2$	$c_6{:}g_6, c_4{:}g_4, c_3{:}g_3$	$c_1{:}g_1$
student_3	$c_4{:}g_4$	$c_1{:}g_1, c_5{:}g_5$	$c_7{:}g_7, c_3{:}g_3, c_6{:}g_6$

This work has three main contributions. First, we introduce the first framework for jointly mining different types of discriminating patterns in a quantitative sequential dataset. Second, we introduce a quantitative pattern mining technique that conditionally splits the quantity range of a given item by considering the pattern that is being expanded by that item, and finding a split that minimizes the total entropy over the expanded pattern. Third, we use UPM to mine students' course enrollment patterns to identify enrollment practices associated with course success. To our knowledge, this is the first work to address the problem through mining discriminating course enrollment patterns and evaluating how well they can label students by their performance.

2 DEFINITIONS AND NOTATIONS

Let $\mathcal{I} = \{i_1, i_2, i_3, \ldots, i_n\}$ be the universal set of distinct items. An itemset p consists of a set of distinct, unordered items $\{i_{p_1}, i_{p_2}, i_{p_3}, \ldots i_{p_l}\}$. An itemset of length l is referred to as an l-itemset. An l-itemset, p_1, is called a superset of another x-itemset, p_2, if $l > x$ and all the items in p_1 are in p_2. An item sequence s consists of an ordered sequence of itemsets $\langle p_1, p_2, p_3, \ldots, p_m\rangle$. The items within each itemset p_i are not ordered whereas the itemsets are ordered. The length of an item sequence is equal to the sum of the lengths of all the itemsets in it. An item sequence of length l is referred to as an l-item sequence. A quantitative itemset r is a set of distinct, unordered items that each is associated with a quantity $\{i_{r_1} : q_{r_1}, i_{r_2} : q_{r_2}, i_{r_3} : q_{r_3}, \ldots i_{r_l} : q_{r_l}\}$. A quantitative item sequence s consists of an ordered sequence of quantitative itemsets $\langle r_1, r_2, r_3, \ldots, r_m\rangle$.

A quantitative sequential dataset *SD* consists of k quantitative item sequences, and they are referred to as the data instances of *SD*. *SD* can be considered a sequential dataset by simply ignoring the item quantities. It can also be considered an itemset dataset by ignoring the sequencing information and considering each data instance as a set of unordered items. Each data instances in *SD* is associated with a class label and an ID. The k instances in *SD* have IDs from 1 to k. Given a frequent pattern p in *SD*, the terms *Conditional Dataset* and *Projected Dataset* of p refer to all instances in *SD* that contain p.

In many mining algorithms, the term *Pattern Expansion* refers to expanding a frequent pattern (itemset or item sequence) into all its frequent supersets. It is also referred to as *Prefix Expansion*.

3 RELATED WORK

Classifiers that utilize frequent itemsets as features [2] [4] [15] [7] [3] can be divided into three categories: 1) methods that first mine all frequent patterns, then extract the discriminating ones, and finally use them as features to represent the data instances and build a classifier, 2) methods that directly mine the discriminating patterns and then build the classifier, and 3) methods that directly builds a rule-based classifier from the input data. The second and third categories were developed to improve over the efficiency of the first category.

HARMONY [15] directly mines the classification rules by using an instance-centric approach and efficient search space pruning to accelerate the mining process. It starts with single frequent items and considers each as a prefix. For each prefix, it builds its conditional dataset, and mines all its rules. For each data instance, it keeps the highest-confidence rule that is mined so far. This is done with each new prefix that is generated during the mining process. At the end, each data instance is associated with its highest-confidence rule. The final classifier is then built by dividing the final set of rules into groups as per the number of classes. To classify a new instance, a score is computed for each group and the instance is classified into the class whose group returns the highest score.

DDPMine [4] uses a feature-centric approach. The dataset is represented using an FPTree (frequent-pattern tree) that saves class label information and it is used for efficiently mining frequent itemsets. Similar to HARMONY, it starts with single frequent items, considered as prefixes. For each prefix, it builds its conditional FPTree and expands it to get longer itemsets that are considered as new prefixes, and so on. The information gain (*IG*) of each itemset is computed during the mining and the maximum *IG* is maintained along with the corresponding itemset. At the end of the mining process, the itemset with the maximum *IG* is selected, and the FPTree is updated to remove the data instances that are covered by it. This process is repeated until all the data instances are covered. In order to prune the mining process, the upper bound on the *IG* is computed for a prefix before expanding it, and if it is less than the maximum *IG*, then the prefix is not expanded.

Classifiers that are based on frequent item sequences [6] [12] are also divided into the same three categories that are described above. BIDE-Discriminative [6] mines discriminating closed item sequences. It considers the class labels while mining the closed frequent item sequences. Similar to HARMONY, it relies on prefix expansion. Before a prefix is expanded, it computes the upper bound on IG, and if it is less than the maximum IG, then the prefix is not expanded. The mining process repeats until k patterns are extracted and it is done with BIDE[14], which mines frequent closed sequences. PrefixSpan [9], on the other hand, mines frequent sequences and it tends to be faster than BIDE with increasing support threshold.

Quantitative pattern mining [8] [11] considers the quantities that are associated with the items. SQUIRE [8] proposed two approaches. The first one considers each item-quantity pair as a new item, and then mines for frequent patterns. The second one uses predefined coarse-grain quantities intervals. So each item-interval pair is considered as a new item, and then it mines for frequent patterns. Then for each frequent pattern, it finds fine-grain frequent patterns within it. The second approach is more efficient as it discards the non-frequent coarse-grained patterns. These approaches are designed to extract frequent patterns, and not discriminating ones.

The work done in [11] combines the quantities for each item into intervals based on information loss. Developed in the context of association rule mining, the intervals are evaluated in terms of how the rules that are generated based on the combined attribute quantities are different from the rules that are generated without combining. In particular, for each rule in the original data (without combining the attribute quantities into intervals), they check how far the closest rule is in the modified data (after combining the attribute quantities into intervals). Rule closeness indicates rule generalization, and rule-rule distance is based on the ratio between the rule supports. The method allows the user to provide a measure for information loss, and the item intervals are determined accordingly. It also introduces a maximum support parameter and uses it to avoid over-combination of adjacent quantities.

Multi-dimensional sequential pattern mining [10] accounts for side information that is associated with sequential data such as time, place, demographics or customer groups.

4 DISCRIMINATING ITEMSET MINING

The problem of mining discriminating itemsets is stated as follows. Given a set of m items $i_1, i_2, \ldots i_m$, a sequential quantitative dataset SD, and a minimum support threshold ϵ, it is required to extract a set of itemsets that covers SD with maximized information gain.

We use the efficient DDPMine method [4] to address this problem. With each step, DDPMine extracts the most discriminating itemset from among all the frequent itemsets. Then it eliminates the data instances that are covered by it, updates the mining structures, and repeats until all data instances are covered.

To apply DDPMine on SD, item quantities and sequencing information are ignored and each data instance is considered as a set of items. DDPMine starts by computing the support per item, removing items with support less than ϵ, and sorting the items in each instance by decreasing item support. Then it represents the sorted dataset using a compact Frequent Pattern Growth Tree (FP-Tree) structure that also stores the sequence IDs and class labels. To find the most discriminating itemset, it starts with single frequent items, or 1-itemsets and uses the FPTree to find longer frequent itemsets. The FPTree efficiently expands a given l-itemset into all its $l+1$-immediate supersets. While mining, DDPMine keeps track of the most discriminating itemset so far. It also uses branch and bound as a mechanism to prune and speedup the mining process.

Information Gain (IG) is used to determine the most discriminating itemset. IG is proportional to the itemset support and it is computed for an itemset X as

$$IG(C|X) = H(C) - H(C|X),$$

where $H(C)$ is the total entropy for all data instances, and $H(C|X)$ is the conditional entropy computed for the data instances that contain X. $H(C)$ is computed as

$$H(C) = -\sum_{c \in C} (\frac{n_c}{n})\ \log(\frac{n_c}{n}),$$

where C is the set of all classes, n is the total number of data instances, and n_c is the number of data instances with class label c. Similarly, $H(C|X)$ is computed as

$$H(C|X) = -\sum_{c \in C} (\frac{n_{c,X}}{n_c})\ \log(\frac{n_{c,X}}{n_c}),$$

where $n_{c,X}$ is the number of data instances that contain X and have class label c.

DDPMine uses branch and bound to prune the mining process. Before expanding an itemset X, it computes the upper bound on its information gain $IG_{ub}(C|X)$. If $IG_{ub}(C|X)$ is less than or equal to the maximum IG over the extracted patterns so far, then it can safely skip expanding X. We assume a multi-class problem with $C \geq 2$ and use a one-vs-all strategy [14] to compute $IG_{ub}(C|X)$ for an itemset X as described in [5].

5 DISCRIMINATING ITEM SEQUENCE MINING

Mining discriminating item sequences is similar to mining itemsets except that we mine a set of item sequences that covers SD with maximized information gain.

PrefixSpan [9] is an efficient algorithm for mining frequent item sequences. For each frequent item f, it constructs a projected dataset SD_f that contains the sequences that have f and the position of f in each sequence. It uses SD_f to expand f to all its frequent sequences of length 2. This process is repeated until all the frequent sequences are discovered.

We modify PrefixSpan to mine discriminating item sequences using the same ideas that were applied in DDPMine. That is, we extract the item sequence that has the highest IG, exclude the data instances that are covered by it, and repeat until all the data instances are covered. The branch and bound approach is also used to prune the mining process based on the upper bound IG for each item sequence before expanding it. The Discriminating PrefixSpan procedure listing, D-PrefixSpan, is described at the end of Section 6.2

6 DISCRIMINATING QUANTITATIVE PATTERN MINING

In many datasets, the items are associated with quantities that represent counts, prices, etc. In the case of mining student enrollment patterns, items are courses, and item quantities represent students grades. Since we want to find the enrollment patterns that are associated with high and low performances in a course, it is essential to consider the students grades. That's because a student's grades in the courses taken before a course c are indicative of his performance in c.

Previous quantitative pattern mining methods, such as SQUIRE [8], either consider each item-quantity pair as a new item, or they consider coarser-grained quantity intervals that are further refined in order to improve the time performance. None of the previous methods mine for discriminating quantitative patterns.

In this work, we modify the discriminating pattern mining techniques that rely on pattern expansion (DDPMine and D-PrefixSpan) in order to consider item quantities. When expanding a pattern p by an item i, instead of simply appending i to p to obtain a new pattern $p' = p \cup \{i\}$, we find a minimum entropy split over the quantities of i in the instances that contain p'. If that split yields a number of intervals v, then quantitatively-expanding p by i leads to v new patterns $p'_1, p'_2, \ldots, p'_v$, one for each interval.

This expansion-specific method finds more discriminating patterns than doing a static quantity-split for each item prior to mining. It can be used to consider quantities in itemset and item sequence mining. In the next subsections we will discuss finding a minimum entropy split over a given set of quantities, and how it is applied for mining discriminating quantitative itemsets and item sequences.

The idea of considering item quantities to discriminate between different classes was previously used in association rule mining [11], where the quantities of a certain item could be divided into intervals in order to generate purer data splits. However, sequencing information, accounting for different types of patterns (e.g., itemsets and item sequences), or different classes cannot be considered by these methods.

6.1 Finding a Minimum-Entropy Split

Given a quantitative pattern p that we want to expand by an item i, we extract the set of instances sd_{pi} that contain p and i along with their class labels. The quantities associated with i in sd_{pi} form the range of values that we want to split such that the entropy computed over the resulting intervals is minimized. Without loss of generality, we assume discrete values for item quantities. Discretization can be applied in the case of continuous values.

We construct a list $\mathcal{L}$ of {quantity, label} pairs that holds the quantities of i in sd_{pi} and their corresponding class labels. We sort $\mathcal{L}$ in ascending order with respect to quantity. We want to find the quantities at which $\mathcal{L}$ can be split so that the total entropy over all sd_{pi} is minimized.

Since the class labels can be distributed randomly throughout the range, then finding the optimal split requires checking all possible splits. If we consider a number of distinct quantities n, and a single split point, then the number of possible splits is ${}^nC_1 = n$, as we try to split at each quantity. Similarly, for two split points, the number of possible splits is ${}^nC_2$, and so on. The total number of possible splits becomes $\sum_{i=1}^{n} {}^nC_i$. Checking all splits can be computationally expensive even for a small n. We use a heuristic approach.

The idea is to repetitively bisect $\mathcal{L}$ as long as the bisection yields a lower total entropy [13]. With each step, we find the split that has the minimum entropy over the given list. If the split entropy is less than the total entropy before splitting, then the list is split into two parts. Then the same process is repeated with each part. We refer to this process as the FindDiscriminatingIntervals precedure.

6.2 Mining Discriminating Quantitative Patterns

The problem of mining discriminating quantitative patterns is stated as follows. Given a quantitative sequential dataset SD that consists of n quantitative item sequences, it is required to extract a set of quantitative patterns that covers SD with maximized information gain. we use the generic word pattern which can be used for itemsets or item sequences.

To mine discriminating quantitative itemsets, we modify DDPMine in order to account for item quantities, and we refer to it as Q-DDPMine. We introduce another variable to the edges of the FP-Tree to hold item quantities, and we refer to it as Q-FPTree. When expanding an itemset p by an item i, Q-DDPMine applies the FindDiscriminatingIntervals procedure to the quantities that i takes in the data instances that contain p and i. It then uses the returned intervals to expand p by i.

Notice that the difference between mining itemsets and mining quantitative itemsets is the step of finding a discriminating interval over the item quantities before expansion. If we discard this step, then we will get the DDPMine procedure for mining discriminating itemsets that is discussed in Section 4

The same ideas are used to mine discriminating quantitative item sequences. We modify D-PrefixSpan into DQ-PrefixSpan, Discriminating Quantitative PrefixSpan, by considering the quantities of an item i when that item is used to expand a sequence p. The DQ-PrefixSpan procedure is shown in Algorithm 1. Its structure is similar to Q-DDPMine, except that it cannot use a Q-FPTree for pattern expansion. Instead, for each item it constructs a projected dataset and uses it for expansion via the procedure DiscrNodeMine-Seq (lines 18-19). DiscrNodeMine-Seq is listed in Algorithm 2. It is used to expand a quantitative item sequence. It is similar to DiscrNodeMine, except that it expands an item sequence using its projected dataset (line 11) rather than using a Q-FPTree. Notice that the difference between mining item sequences and mining quantitative item sequences is finding a discriminating interval over the item quantities before expansion. If we discard these steps in Algorithms 1 and 2, we will get the D-PrefixSpan procedure for mining discriminating item sequences discussed in Section 5.

7 UPM: UNIVERSAL DISCRIMINATING PATTERN MINING

So far we have discussed separate methods for mining discriminating itemset, item sequence, quantitative itemset and quantitative item sequence patterns. In this section we present UPM, a feature-centric framework for simultaneously mining different types of discriminating patterns. UPM combines the four algorithms described above, DDPMine, Q-DDPMine, D-PrefixSpan and DQ-PrefixSpan,

Algorithm 1 Mining Discriminating Quantitative Item Sequences

```
1:  procedure DQ-PrefixSpan
2:  Input: SD: Sequential Dataset, ProjDB: Projected Dataset Structure, ϵ: min support
3:  Output: F_Q: Set of Discriminating Item Sequences with Quantity Intervals
4:  ----------------------------------------
5:      F_Q ← ∅
6:      while number of uncovered instances in SD > 0 do
7:          e ← TotalEntropy(SD)
8:          M^max ← ∅
9:          maxIG ← −1
10:         for each frequent item t in SD do
11:             intervals ←FindDiscriminatingIntervals(t, SD)
12:             for each interval i ∈ intervals do
13:                 M ← {t : i}
14:                 ProjDB ← GetProjDB(M)
15:                 DiscrNodeMine-Seq (e, t, i, ProjDB, M, M^max, maxIG)
16:             end for
17:         end for
18:         F_Q ← F_Q ∪ M^max
19:         SeqIDs ← GetSeqIDs(M_max)
20:         RemoveSeqIDs(SD, SeqIDs)
21:     end while
22: end procedure
```

Algorithm 2 Recursively Expanding a Quantitative Item Sequence to Find the Sequence with the Maximum IG.

```
procedure DiscrNodeMine-Seq
Input: e: Total Entropy, t: item, i: value interval for t, ProjDB: Projected Dataset Structure, M: Current Quantitative Item Sequence, M^max: Quantitative Item Sequence with Maximum IG, maxIG: Maximum IG
Output: Discriminating Quantitative Item Sequence stored in M^max
----------------------------------------
    if IG(M) > maxIG then
        maxIG ← IG(M)
        M^max ← M
    end if
    if maxIG ≥ IG_ub(M) then return
    end if
    support ←Expand(ProjDB)
    for each item n with support[n] > 0 do
        intervals ←FindDiscriminatingIntervals(n, T)
        for each interval j ∈ intervals do
            M^new ← M ∪ {n : j}
            DiscrNodeMine-Seq(e, n, j, ProjDB, M^new, M^max, maxIG)
        end for
    end for
end procedure
```

in order to mine the different patterns simultaneously. It mines the most discriminating pattern among all types, then it removes the data instances that are covered by that pattern, updates the mining structures, and repeats until all data instances are covered.

The UPM procedure is listed in the Algorithm 3. It starts by initializing the structures that are used in mining the different types of patterns (line 5). These structures are a FPTree (T), a Q-FPTree (QT), a Projected Dataset structure (ProjDB), and another Projected Dataset (QProjDB) for mining discriminating itemsets, quantitative itemsets, item sequences, and quantitative item sequences, respectively. The mining process starts at line 6. At lines 7, 8, 9 and 10, the itemset p_1, quantitative item set p_2, item sequence p_3 and quantitative item sequence p_4 that have the maximum IG are returned, along with the IG values ig_1, ig_2, ig_3 and ig_4, respectively. Among these, the pattern with maximum IG, p_{max}, is returned at line 11. Then p_{max} is added to the set of discriminating patterns $\mathcal{F}$ at line 12. In line 13, the IDs of the instances that are covered by p_{max} are returned, and in line 14 the mining structures are updated accordingly. This process repeats until all data instances are covered.

Procedure GetMaxIGItemset (line 7) performs one scan of the data instances to find the most discriminating itemset using the FPTree structure and the IG-based branch and bound mechanism to prune the search space. Procedure GetMaxIGQuantItemset (line 8) performs one scan to get the most discriminating quantitative itemset. Similarly, the procedures at lines 9 and 10 extract the most discriminating item sequence and quantitive item sequence, respectively. They are equivalent to executing lines 7-17 in Algorithm 1 without and with considering item quantities, respectively.

Algorithm 3 UPM: Mining Different Discriminating Patterns

```
1:  procedure UPM
2:  Input: SD: Sequential Dataset, ϵ: Minimum Support Threshold, F: Set of Discriminating Patterns
3:  Output: Set of All Discriminating Patterns F
4:  ----------------------------------------
5:      {T, QT, ProjDB, QProjDB} ← InitMiningStructures(SD, ϵ)
6:      while number of uncovered sequences > 0 do
7:          {ig_1, p_1} ← GetMaxIGItemset(T)
8:          {ig_2, p_2} ← GetMaxIGQuantItemset(QT)
9:          {ig_3, p_3} ← GetMaxIGItemsequence(ProjDB)
10:         {ig_4, p_4} ← GetMaxIGQuantItemsequence(QProjDB)
11:         p_max ← MaxIGPattern(ig_1, ig_2, ig_3, ig_4, p_1, p_2, p_3, p_4)
12:         F ← F ∪ p_max
13:         SeqIDs ← GetSeqIDs(p_max)
14:         UpdateMiningStructures(T, QT, ProjDB, QProjDB, SeqIDs)
15:     end while
16: end procedure
```

8 MINING DISCRIMINATING STUDENT ENROLLMENT PATTERNS

The performance of a student in a course c is determined by many factors. One important factor is the courses that he has taken before c and his performance in them. Prior courses are assumed to provide the necessary knowledge for future courses, and the obtained grades are a quantitative indicator of knowledge acquisition. Taking a course without fully acquiring its knowledge components will lead to poor performance in a future course that builds upon these components.

Our goal is to find, for a target course c, the prior course sets, along with sequence and performance information, that are associated with high and low performance in c. The input data consists of:

- c: the target course
- $\mathcal{S}_c$: the set of students that have taken c
- $Q\{\mathcal{S}_c\}$: A set that holds for each student $s \in \mathcal{S}_c$, the sequence of courses that s has taken prior to c, and his grades in these courses

We formulate a binary classification problem as follows. First, we extract $\mathcal{S}_{c,a} \subset \mathcal{S}_c$, the subset of students with high performance in c, and $\mathcal{S}_{c,b} \subset \mathcal{S}_c$, the subset of students with low performance in c. Second, we create a quantitative item sequence dataset *SD* from $Q\{\mathcal{S}_c\}$ as follows. Each instance in *SD* represents a student s as a quantitative item sequence that shows the course sets that s took in successive terms along with his grades. Grades take the ordinal values $\{A \prec A^- \prec B^+ \prec B \prec B^- \prec C^+ \prec C \prec C^- \prec D^+ \prec D \prec F\}$. Each instance is associated with a 1 or 0 class label depending on whether the student performed highly or low in c, respectively.

Given this input dataset, we apply UPM to extract the discriminating enrollment patterns that are associated with high and low performance in c. Then we use these patterns to represent the students and build classifiers to sort them into the high and low performing classes. These patterns can be itemsets (course sets without any sequencing or grade information), quantitative itemsets (course sets with grade information), item sequences (sequences of course sets without grade information), or quantitative item sequences (sequences of course sets with grade information).

9 EVALUATION METHODOLOGY

In this section we describe how we identify high and low performing students, the course datasets that are used for evaluation, the methods that we compare with, the training procedure and the evaluation metrics.

9.1 Identifying Well- and Poorly-Performing Students

We first define the concept of a *grade tick*. Given a letter grade scale $A, A-, B+, B, B-, C+, C, C-, D+, D, F$, then one grade tick is one step on this scale in any direction. That is, the different between A and $A-$ is one grade tick, and the difference between A and B is three grade ticks.

For a course c, we identify a well-performing student in c as a student who performs above or the same as her average grade prior to taking c. And we identify a poorly-performing student in c as a student that performs at least 3 grade ticks below her average grade prior to taking c.

9.2 Pattern Extraction and Student Representation

After defining the well- and poorly-performing students in a course, we create a quantitative item sequence dataset *SD* for that course as described in Section 8 Then UPM is applied to extract the enrollment patterns that discriminate between the two student groups. These patterns are used as features to represent the students. Let the number of patterns extracted by UPM be l. We define l corresponding binary features, $f_1, f_2, \ldots, f_l$. Then they are used to represent the instances (students) in *SD* as follows. Each student s is represented by l binary features. For feature f_i, if the enrollment pattern that corresponds to f_i exists in the enrollment sequence of s, then we set $f_i = 1$ for s, else it is set to 0.

9.3 Evaluation Datasets

The course datasets that are used for evaluation are obtained from the University of Minnesota. The courses that we have selected for evaluation have a relatively high number of poorly-performing students, and so, would be considered difficult courses. Discovering which enrollment patterns are associated with success in such difficult courses is beneficial for students and college advisers. Table 2 describes the set of courses that are used for evaluation. For each course, it shows the total number of student enrollments, the number of poorly- and well-performing students which are identified as described above. Notice that the total number of enrollments in each course is greater than the number of well and poorly performing students combined. That's because the students that have performed 1 or 2 grade ticks below their average grade are neither considered as well nor poorly performing students and are excluded.

9.4 Comparison Methods

Classification here is only used as a mean for evaluating the effectiveness of the UPM extracted enrollments patterns in differentiating between high and low performing students, we compare various classifiers that are trained using different types of patterns (features). These methods and feature sets are as follows.

- **HARMONY-1**: Each student is represented using the original set of courses that he has taken prior to the target course c. Then the rule-based classifier HARMONY [15] is trained using this data. This method serves as a baseline since no discriminating pattern mining is carried out.
- **HARMONY-UPM**: The students are represented using the set of discriminating enrollment patterns that are extracted by UPM, and HARMONY is used for classification.
- **SVM-UPM**: The students are represented using the set of discriminating enrollment patterns that are extracted by UPM, and SVM with linear kernel is used for classification.
- **SVM-All**: The students are represented using the set of all frequent patterns (itemsets, quantitative itemsets, item sequences and quantitative item sequences) without applying UPM to extract the discriminating patterns. SVM with linear kernel is used for classification.
- **RF-UPM**: The students are represented using the set of discriminating enrollment patterns that are extracted by UPM and Random Forests are used for classification.
- **RF-All**: The students are represented using the set of all frequent patterns (itemsets, quantitative itemsets, item sequences and quantitative item sequences) without applying

Table 2: Description of the courses that are used for evaluation. For the number of poorly- and well-performing cases per course, we also show the percentages of these cases as per the total number of enrollments in that course.

Course Code	Course Name	# poor perf. students	# good perf. students	# enrolls
CPSY-4343	Cognitive Development	205 (23%)	368 (42.2%)	872
CE-4301	Soil Mechanics II	158 (32.4%)	130 (26.7%)	487
LASK-1001	Mastering Skills	182 (42.1%)	250 (57.9%)	432
BIOC-4331	Structure/Catalysis/Metabolism	257 (35.3%)	197 (27%)	728
MATH-5651	Probability & Statistics Theory	192 (31.6%)	223 (36.7%)	608
AEM-4501	Aero Structures	140 (31%)	113 (25%)	453
KIN-4385	Exercise Physiology	244 (37%)	111 (16.8%)	661
EE-4341	Microprocessor System Design	105 (35%)	77 (25.8%)	299
MATS-3011	Introduction to Material Science	421 (32.6%)	338 (26.2%)	1289
ANAT-3601	Principles of Human Anatomy	499 (39.3%)	404 (31.9%)	1268
CHEN-4001	Material & Energy	375 (37.5%)	219 (22%)	999
ACCT-3001	Technology Tools in Accounting	1221 (29%)	1344 (31.8%)	4223
ACCT-5102W	Intermediate Accounting II	301 (32.8%)	151 (16.5%)	917
BIOL-3021	Biochemistry	1397 (28.2%)	1602 (32.4%)	4948
CPSY-4329	Biol Foundations of Development	100 (23%)	208 (48%)	433

UPM to extract the discriminating patterns. Random Forests are used for classification.

Notice that for the different courses, the set of all frequent patterns is around two to three orders of magnitude larger than the set of patterns that is extracted by UPM. For some courses, the number of all patterns and the set of UPM-extracted patterns are around 300,000 and 300, respectively.

9.5 Model Training Procedure

Each course dataset is divided into a 80-20% train-test split. UPM is then applied to extract the discriminating enrollment patterns. The classifiers are trained on the training set and we select the models with the highest classification accuracy on the test set. These models are used for feature analysis in order to extract the enrollment patterns that are associated with the well and poorly performing students.

9.6 Evaluation Metrics

Classification Accuracy: The classification accuracy obtained by the classifiers. It shows how the extracted patterns can generally differentiate between the well and poorly performing student populations.

Geometric Mean Improvement (GMI): This is a generic metric that shows the overall performance of each classification method with respect to the best method in general.

For a given method M, GMI_M is computed as follows, given a number of comparison methods x, and a number of dataset d,

$$GMI_M = \sqrt[x]{\Pi_{i=1}^{d} \frac{a_{i,M}}{a_{i,best}}}, \tag{1}$$

where $a_{i,M}$ is the classification accuracy achieved by method M on dataset i, and $a_{i,best}$ is the best accuracy achieved by all methods on dataset i. A high GMI_M indicates that method M performs better than other methods in general.

10 RESULTS AND ANALYSIS

We assess the effectiveness of the developed methods in order to answer the following questions:

- Q1. Are the enrollment patterns extracted by UPM more discriminating between highly and low performing students, and so, lead to higher classification accuracy than using previously-taken courses as is without pattern extraction?
- Q2. How do the patterns extracted by UPM perform compared to the performance of each type of pattern separately?
- Q3. Does considering the item quantities (course grades) lead to improved classification accuracy?
- Q4. How do the enrollment patterns contribute to student classification in the various models?

10.1 Classification Accuracy Results

Table 3 shows the overall classification accuracies by the different methods for the different courses. HARMONY-1 gives the lowest classification accuracy for all the courses. This indicates the relatively poor discriminating power by the set of previously taken courses when they are used as is. For most courses, HARMONY-UPM and SVM-UPM give the highest accuracy, except for a few cases in which SVM-All and RF-All outperform them. This shows that the enrollment patterns that are extracted by UPM can better discriminate between high and low performance.

The last row in Table 3 shows the GMI for the various methods. SVM-UPM and HARMONY-UPM have the highest GMI, indicating that they generally perform better than the other methods. HARMONY-1 has the lowest GMI, indicating the effectiveness of discriminative pattern extraction in providing better separation between the high and low performance classes.

In order to investigate the gain achieved by UPM over the other techniques that mine a single type of pattern, we explored the discriminating ability of each pattern type on its own. Table 4 shows the classification accuracies obtained by training a linear-SVM classifier using each type of discriminating pattern separately to represent the students. UPM patterns give higher accuracy for most courses, showing the effectiveness of combining different pattern types in achieving better discrimination between high and low performance.

Table 4 also shows that for most courses, the accuracies obtained using itemsets only are higher than the ones obtained using quantitative itemsets only. The same applies for item sequences versus quantitative item sequences. This shows the effectiveness of considering item quantities in mining more discriminating patterns.

10.2 Classification Accuracy for Different Number of Patterns

UPM extracts some number of patterns, l, to cover a course dataset. Students are represented using l binary features that correspond to the extracted patterns. For each student, the number of features with non-zero value is determined by how many of the l patterns exist in the enrollment sequence of that student. Different students can have a different number of non-zero features. In this experiment, we investigate how the number of non-zero features influences classification accuracy. For each course, we divide the test set students into groups based on their number of non-zero features. Then we compute the classification accuracy for each group.

The plots in Figure 1 show, for each course, the number of non-zero features per group versus the classification accuracy. Accuracy tends to increase with increasing number of features. In some courses, such as ACCT-3001 and BIOL-3021, accuracy reaches 100%. These plots can be used to determine the number of non-zero features at which confidence in the classification results is high, and so, they can be used by instructors, advisers or students to predict future course success.

10.3 Analysis of Pattern Types Extracted by UPM

Table 5 shows the percentage of each pattern type that was extracted by UPM. For most courses, itemset patterns have higher percentages than other pattern types. We think this happens because the discriminative power of a pattern is determined using *IG*, which is proportional to the pattern support [2], and the more sophisticated patterns have lower support counts that the less sophisticated ones. For example, a quantitative itemset pattern that contains a set of items, $\mathcal{S}$, only tends to have a lower support count than the itemset pattern that contains $\mathcal{S}$, only. Similarly, an item sequence pattern that contains only the set of items $\mathcal{S}$ tends to have a lower support count than the itemset pattern that contains only $\mathcal{S}$.

Usually, at the beginning of the mining process, more sophisticated patterns are selected. Then as the process continues, less sophisticated patterns are more selected due to their relatively higher support counts that contributes the *IG* computation.

10.4 How Course Enrollments Contribute to Labeling Students as High- or Low-Performing

The HARMONY and Linear-SVM models that we use to classify students based on their past enrollment patterns are not black box classifiers. In fact, they can provide useful insights about how the different enrollment patterns contribute to classification. We investigate how the patterns extracted by UPM contribute to student classification in each model.

10.4.1 Classification Rules Mined by HARMONY. For each performance group, HARMONY provides the set of rules that are used to classify instances for that group. It also provides rule support and confidence, which give insights on the level to which the rule body is found in the data, and the fraction of instances in which the rule body was found along with the corresponding group, respectively. Therefore, HARMONY is a good model choice for understanding how the features contribute to classification.

Table 6 shows a few rules for EE-4341 (Microprocessor System Design). The high performance rules show obtaining A or A- in various courses such as EE-3101 (Cir Elec Lab I), whereas a low performance rule shows a C+ in EE-2011 (Lin Sys Cir & Elec). Such rules look reasonable as high performance in the target course is associated with high performance in some previous courses, and the same for low performance.

10.4.2 Feature Importance of Linear SVM. The feature weights that are learned by linear SVM represent feature importance [1]. In our case, features with the highest positive weights and the lowest negative weights represent the enrollment patterns that are mostly associated with high and low performances, respectively.

Table 7 shows the enrollment patterns for EE-4341 (Microprocessor System Design). The patterns associated with high performance show obtaining A- or A in various courses such as EE-3101 (Circuit Electronic Lab I) as well as certain course sequences. Patterns associated with low performance show obtaining C+ in EE-2011 (Linear System Circuit Electronics)

The overlap between the linear-SVM patterns and the HARMONY patterns is obvious. Also as with HARMONY, some patterns associated with low performance show itemsets, which requires further investigation of the underlying students in order to understand the missing factors contributing to their low performance in the target course.

11 CONCLUSIONS AND FUTURE WORK

In this work we presented UPM, a method for simultaneously mining different types of discriminating patterns. UPM accounts for item quantitiesand finds a minimum-entropy split over the quantities of an item based on the pattern that is being expanded by it. We used UPM to mine student enrollment practices that discriminate between high and low performing students in a target course. We have quantitatively evaluated the effectiveness of the extracted patterns by using them as features to represent students and building classifiers that sort students into their different performance classes.

Table 3: Classification Accuracy for the different methods. The Geometric Mean Improvement (GMI) for the various methods are listed in the last row.

Course	HARMONY-1	HARMONY-UPM	SVM-UPM	RF-UPM	SVM-All	RF-All
CPSY-4343	65.0%	70.8%	**75.8%**	71.7%	67.5%	75.0%
CE-4301	63.8%	**70.7%**	**67.2%**	64.9%	67.2%	64.6%
LASK-1001	60.2%	**72.7%**	**72.7%**	64.0%	70.4%	66.9%
BIOC-4331	53.8%	**81.3%**	**76.9%**	**73.1%**	69.23%	70.7%
MATH-5651	54.8%	**71.4%**	**66.7%**	**67.2%**	60.7%	65.3%
AEM-4501	53.7%	**79.6%**	**81.5%**	70.8%	77.8%	68.4%
KIN-4385	76.1%	77.5%	76.1%	77.5%	**84.5%**	79.2%
EE-4341	40.5%	**71.4%**	**73.8%**	70.3%	71.4%	65.9%
MATS-3011	66.9%	**74.8%**	73.5%	71.6%	73.5%	73.4%
ANAT-3601	60.5%	76.2%	**77.8%**	73.8%	**77.8%**	71.4%
CHEN-4001	59.5%	69.4%	68.6%	72.6%	68.6%	**74.4%**
ACCT-3001	65.7%	**68.0%**	**70.1%**	64.7%	66.3%	65.0%
ACCT-5102W	65.9%	74.7%	74.7%	72.1%	**80.2%**	75.4%
BIOL-3021	64.9%	71.9%	**73.3%**	69.6%	**74.9%**	69.6%
CPSY-4329	65.1%	73.0%	**79.4%**	69.5%	73.0%	74.0%
GMI	0.791	**0.966**	**0.968**	0.921	0.942	0.926

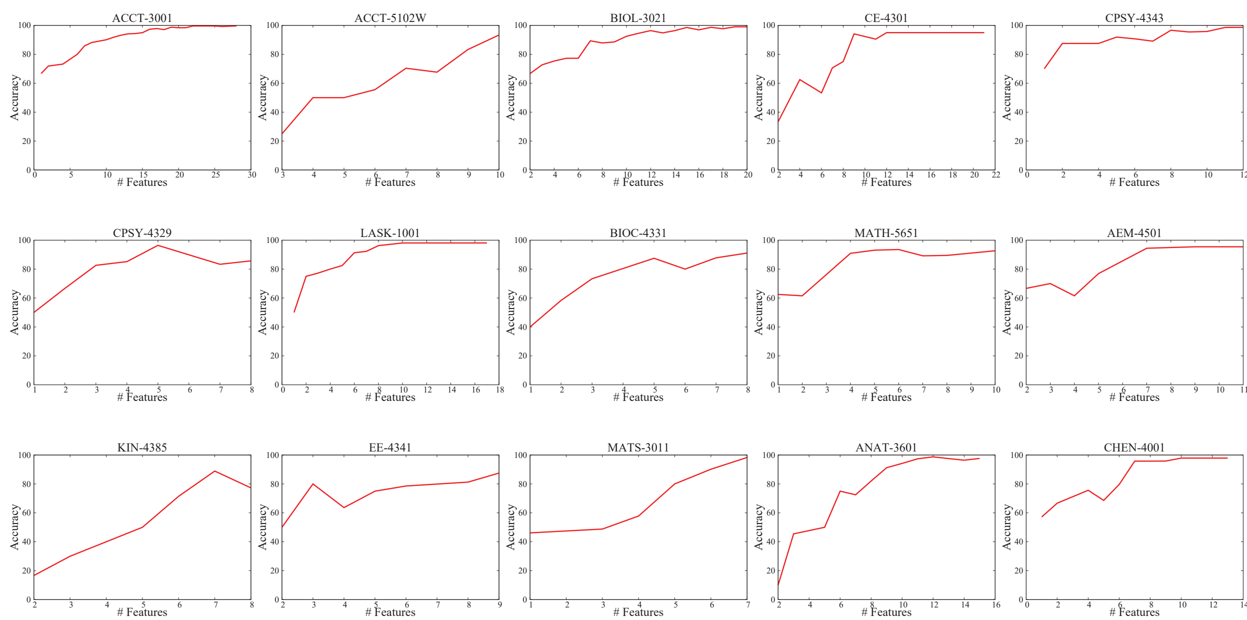

Figure 1: The classification accuracy versus the number of features (enrollment patterns) for the different courses.

Our evaluations showed that effectiveness of the UPM-extracted patterns as compared to the set of all patterns, and the set of previously taken courses without any pattern extraction. Our evaluation also showed the effectiveness of accounting for item quantities, representing course grades, in mining more discriminating patterns. We used the estimated SVM linear model to better identify the enrollment patterns that are associated with high and low performances for a target course. It showed that in some cases, taking courses in a certain sequence and performing high (low) in them is associated with high (low) performance in a target course. These findings can be utilized by instructors and degree programs to design better degree plans.

In the future we will investigate merging the mining of different patterns (itemsets, quantitative itemsets, or item sequences). We will investigate leveraging the fact that the computations for mining an itemset does not only cover its supersets, but also other pattern types (quantitative itemsets, item sequences and quantitative item sequences) that have the same items. We will also further investigate

Table 4: Classification accuracy for training an SVM model with linear kernel over each type of discriminating pattern separately.

Course	Itemsets	Quantitative Itemsets	Item Sequences	Quantitative Item Sequences
CPSY-4343	70.1%	**74.1%**	70.8%	73.0%
CE-4301	64.1%	71.9%	62.1%	**75.4%**
LASK-1001	65.6%	68.9%	60.2%	**70.4%**
BIOC-4331	71.9%	73.4%	64.8%	**74.8%**
MATH-5651	**60.1%**	58.5%	58.7%	59.1%
AEM-4501	62.1%	65.2%	61.1%	**70.3%**
KIN-4385	74.1%	76.5%	77.4%	**80.3%**
EE-4341	61.9%	71.4%	69.0%	**71.7%**
MATS-3011	66.4%	67.5%	67.4%	**69.8%**
ANAT-3601	70.1%	**71.8%**	65.9%	69.0%
CHEN-4001	68.6%	71.0%	**78.5%**	70.0%
ACCT-3001	64.2%	65.3%	**66.3%**	65.9%
ACCT-5102W	65.1%	69.4%	67.0%	**72.0%**
BIOL-3021	64.8%	**70.6%**	65.9%	69.3%
CPSY-4329	65.8%	**76.0%**	69.8%	73.3%

Table 5: Percentage of the features that are itemsets (course sets), quantitative itemsets (graded course sets), item sequences (course sequences) and quantitative item sequences (graded course sequences) for each course.

Course	Itemsets	Quantitative Itemsets	Item Sequences	Quantitative Item Sequences
CPSY-4343	48%	37%	2%	11%
CE-4301	0%	60%	33%	6%
LASK-1001	34%	39%	4%	21%
BIOC-4331	70%	20%	4%	4%
MATH-5651	58%	23%	10%	9%
AEM-4501	46%	38%	0%	15%
KIN-4385	71%	29%	0%	0%
EE-4341	31%	50%	18%	0%
MATS-3011	38%	45%	0%	16%
ANAT-3601	55%	33%	3%	7%
CHEN-4001	56%	34%	0%	8%
ACCT-3001	54%	29%	3%	11%
ACCT-5102W	52%	48%	0%	0%
BIOL-3021	71%	19%	2%	7%
CPSY-4329	72%	24%	0%	3%

patterns associated with low performance focusing on what other courses students should have taken.

Table 6: Classification rules returned by HARMONY for EE-4341 (Microprocessor System Design).

Class	Conf	Sup	Rule Body
1	0.92	11	Quant-Itemset: EE-3101(Cir Elec Lab I)[A-,A], ECON-1101 (Principals of Microeconomics) [A-,A]
1	0.74	13	Quant-Itemset: CSCI-1902 (Computer Programming II) [A-,A]
1	0.70	6	Quant-Itemset: EE-2301 (Intro Digital Design) [A-,A], CSCI-2011 (Discrete Structures) [A-,A]
0	0.92	11	Quant-Itemset: EE-2011 (Linear Systems, Circuits & Electronics) [C+]
0	0.85	11	Itemset: EE-2002 (Circuits & Electronics Lab), EE-3101 (Circuit Electronics Lab I), EE-3115 (Analog Electronics), EE-3102 (Circuit Electronics Lab II), EE-3161 (Semiconductor Development), BIOL-1009 (General Biology)

Table 7: Enrollment patterns returned by Linear-SVM with highest feature weights for EE-4341 (Microprocessor System Design).

Class	Feature (Enrollment Pattern)
1	Quant-Itemset: EE-3101 (Circuits & Electronics Lab I) [A-, A], ECON-1101 (Principals of Microeconomics) [A-, A]
1	Itemsequence: PHYS-1302W (Physics for Science&Engnrng II) → EE-2361 (Intro Micro-controllers) → {EE-3015(Signals&Systms),EE-3101 (Circ Elctrncs Lab I)}
0	Quant-Itemset: EE-2011(Linear Systems, Circ&Elctrncs) [C+]
0	Itemset: EE-2002(Circ&Elctrncs Lab),EE-3101 (Circ Elctrncs Lab I), EE-3115 (Analog Electronics), EE-3102 (Circ Elctrncs Lab II), EE-3161 (Semiconductor Dev), BIOL-1009 (General Biology)

REFERENCES

[1] Y.-W. Chang and C.-J. Lin. Feature ranking using linear svm. In *WCCI Causation and Prediction Challenge*, volume 3 of *JMLR Proceedings*, pages 53–64. JMLR, 2008.

[2] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *ICDE*, pages 716–725. IEEE Computer Society, 2007.

[3] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *ICDE*, pages 716–725. IEEE Computer Society, 2007.

[4] H. Cheng, X. Yan, J. Han, and P. S. Yu. Direct discriminative pattern mining for effective classification. In *ICDE '08: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 169–178. IEEE Computer Society, 2008.

[5] A. Elbadrawy. *Prediction, Ranking and Mining Methods for Higher Educational Data.* PhD thesis, University of Minnesota, Minneapolis, MN, 2017.

[6] D. Fradkin and F. Morchen. Mining sequential patterns for classification. *Knowl. Inf. Syst.*, 45(3):731–749, 2015.

[7] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pages 1–12, May 2000.

[8] C. Kim, J.-H. Lim, R. T. Ng, and K. Shim. Squire: Sequential pattern mining with quantities. *J. Syst. Softw.*, 80(10):1726–1745, Oct. 2007.

[9] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. on Knowl. and Data Eng.*, 16(11):1424–1440, Nov. 2004.

[10] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 81–88. ACM, 2001.

[11] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *ACM SIGMOD Conf. Management of Data*, June 1996.

[12] Z. Syed, P. Indyk, and J. Guttag. Learning approximate sequential patterns for classification. *J. Mach. Learn. Res.*, 10:1913–1936, Dec. 2009.

[13] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition).* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[14] J. Wang and J. Han. Bide: Efficient mining of frequent closed sequences. In *Proceedings of the 20th International Conference on Data Engineering*, ICDE '04, 2004.

[15] J. Wang and G. Karypis. Harmony: Efficiently mining the best rules for classification. In *SDM*, pages 205–216. SIAM, 2005.