FuzzyGap: Sequential Pattern Mining for Predicting Chronic Heart Failure in Clinical Pathways

Eric W. Lee, MS, Joyce C. Ho, PhD
Department of Computer Science, Emory University, Atlanta, GA, United States

Abstract

The rapid growth of electronic health records (EHRs) facilitates the use of clinical pathways, an actionable plan for patients which is represented as sequences of diagnostic records ordered by visit dates. We propose to extract discriminative and representative clinical pathways from EHRs using sequential pattern mining. However, existing sequential patterns cannot efficiently extract patterns due to patient variations in length and time period between visits. To resolve this problem, we propose FuzzyGap, a sequential pattern mining-based framework that extracts a discriminative subsequent pattern from the proper representation of the sequence of encounters which also emphasizes the last visit that is more significant than others. We demonstrate FuzzyGap using a case study of heart failure and show the effectiveness of sequential pattern mining.

Introduction

The rapid growth of electronic health records (EHRs) and their availability provides great opportunities to design data-driven approaches to improve clinical decisions[1]. One important source for decision-making process is clinical pathways, a sequence of medical evidence and actions taken for treatment intervention[2]. Clinical pathways are implemented internationally to support physicians to diagnose diseases and determine appropriate treatment as well as to reduce the medical expenses[2]. More than 80% of the hospitals in the US use clinical pathways for the intervention and to improve the decision-making process[3].

Based on the clinical practice guidelines, clinical pathways translate the recommendations into an actionable plan for patients[2]. Thus, the sequence of diagnoses that are recorded when a patient visits a healthcare provider can be considered as a potential clinical pathway. In this paper, we define the clinical pathway as a sequence of diagnoses as described in Figure 1. In each visit (or claim), patient's conditions or diagnoses codes such as International Classification of Disease (ICD-9) codes are recorded and describes patient's disease or medical condition. However, clinical pathways vary in length such as outpatient visiting the physician a few times, while inpatient getting diagnoses regularly. Figure 1 illustrates two patients with different number of claims and different time periods between two visits.

The use of machine learning and data mining models for clinical pathway extraction is based on the idea that patients with similar medical conditions have similar patterns (or clinical pathways). Thus, clustering or disease-based efforts can be used to find diagnostic pathways. Zhang et al[4] proposed a method to cluster patients using clinical pathways derived from EHRs as they contain patients' biochemical conditions that are reflected by their laboratory observations. Lee et al[5] predicted 7-day heart failure mortality in emergent care using vital signals, and clinical features. Yet, these algorithms are reliant on specific conditions, they cannot generalize readily to varying observation lengths and incorporate multiple sources of information.

An alternative data mining methodology is to use temporal pattern mining approaches. The extracted patterns can be exploited as significant features for analyzing or classifying patients. For example, Batal et al[6] proposed a temporal pattern mining approach based on Allen's temporal logic[7] to classify cardiac surgical patients. However, Allen's temporal logic requires continuous diagnostic records (e.g., blood pressure or heart rate measurements). Thus, they do not readily encapsulate the structured data that is commonly present in EHRs. Moreover, data can vary across patients in terms of frequency and length (see Figure 1). As two extreme examples, Khoshnevisan et al[8] extracted clinical pathways based on minute resolutions from the arrival of emergency patients to predict septic shock using Blood Pressures and Lactate, whereas Zhang et al[9] extracted clinical pathways for 4 years using drug classes and diagnoses to cluster chronic kidney disease patients.

Instead, we posit that sequential pattern mining can be used to extract discriminative and representative clinical pathways from EHRs. Sequential pattern mining identifies interesting subsequences in a set of sequence, similar to frequent

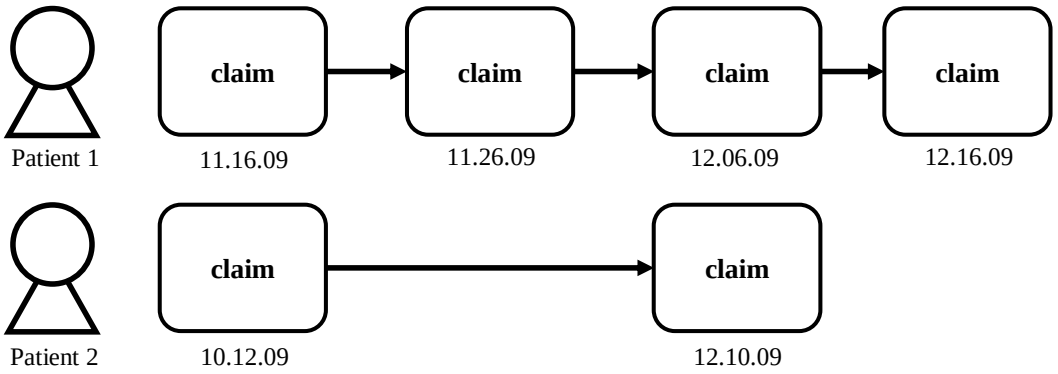


Figure 1: Example of the sequence of encounters for two patients.

itemset in the database[10]. For a time-series dataset, sequential pattern mining can be used by converting timestamps into a sequence by using a discretization technique[11]. There are numerous algorithms for sequential pattern mining[10–13] where the basic idea is to find these subsequences that exceed a user-specified minimum support count (number of sequences that contains the pattern). The main difference between the variants lies in the utilization of the data structure they use to scan through the dataset for finding patterns efficiently. Yet, even the state-of-the-art fast sequential pattern mining algorithms cannot readily scale to large patient cohorts and are limited by the patient representation.

In this paper, we present FuzzyGap, a sequential pattern mining-based framework to extract discriminative clinical pathways. As patient encounters vary in length and frequency, we propose a FuzzyGap-based patient representation to extract the sequential patterns. Without this representation, the extracted patterns are either too unique or too common to distinguish the two patient populations (e.g., heart failure vs. non-heart failure). A pattern that is too unique will not generalize well to other patients while common patterns will offer limited discriminative power. Moreover, the most recent visits should be more significant and visits that are close in time should be modeled using a similar representation[14]. In other words, two visits in a short time period may be more suitable in a single visit representation, and thus have less impact on the prediction process. We also introduce a simple filtering process that can lead to better patient representation and consequently improved predictive accuracy. We demonstrate our model using a case study of heart failure prediction in diabetic patients. We show that sequential pattern mining can improve the predictive power over non-temporal models.

Methods

A Case Study of Chronic Heart Failure in Diabetic Patients. Diabetes is an epidemic, affecting approximately 11% of the U.S. population in 2015[15]. Diabetic patients are at risk of a wide spectrum of comorbidities, which may lead to complications in care and result in a heavy economic burden[16]. HF is also a leading cause of healthcare use with a projected medical cost in 2015 of $32.5 billion[17]. HF affects roughly 5.7 million people in the US and is mentioned as the contributing cause for 1 out of every 9 deaths[18]. There exists a particularly strong correlation between HF ad diabetes with at least 68% of elderly diabetics (age 65 or older) dying from some form of heart disease[19]. Not only is the healthcare expenditure and resource utilization exceptionally high in diabetic patients with HF, but early intensive intervention in high–risk HF patients can be cost–effective and lead to favorable health outcomes[20]. Thus, we focus on diabetic and predict whether the patient will develop HF in the future by using their clinical pathways.

FuzzyGap Overview. We propose a sequential pattern mining-based framework to extract discriminative, and representative clinical pathways from a large cohort. FuzzyGap consists of the 3 steps as depicted in Figure 2. The first step is to restructure the sequence of encounters to be coherent with the sequential pattern mining representation. Next, our framework extracts clinical pathways from each class (i.e., case and control) and determines the discriminating patterns for each class. Finally, these representative patterns are used to construct a new patient feature representation. A statistical model is then learned on this feature representation to predict the risk of heart failure for each patient. For the purpose of our study, we construct clinical pathways using the Clinical Classifications Software (CCS) codes, a categorization scheme for the International Classification of Diseases, Ninth Revision (ICD-9) codes that are used in

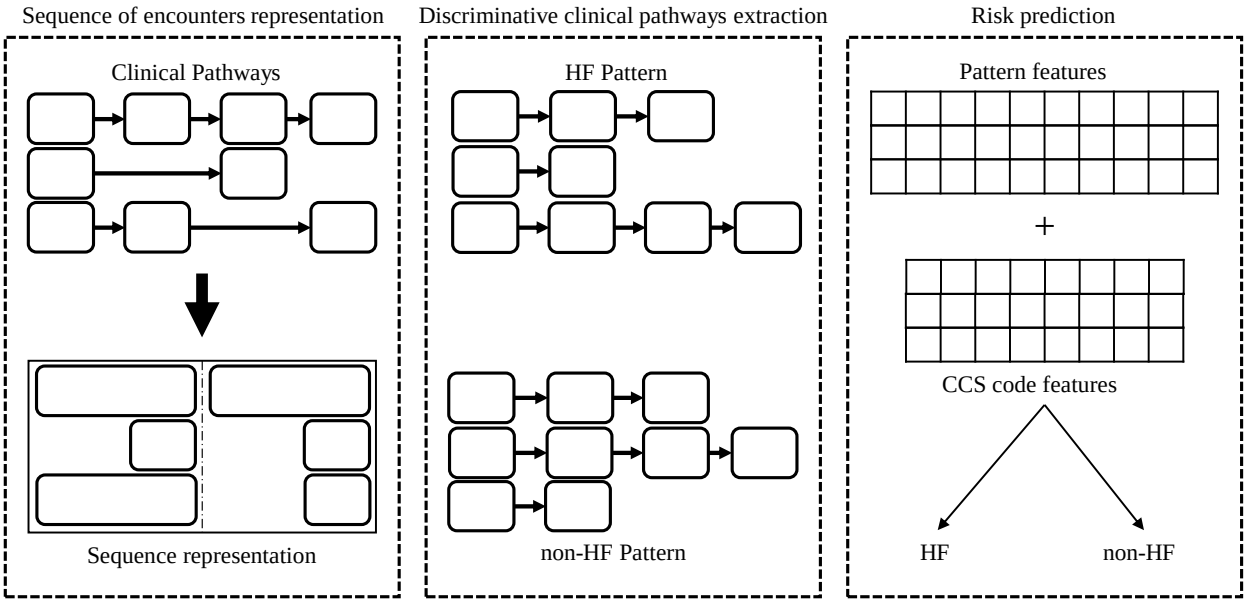


Figure 2: An overview of FuzzyGap for chronic heart failure prediction which has three steps.

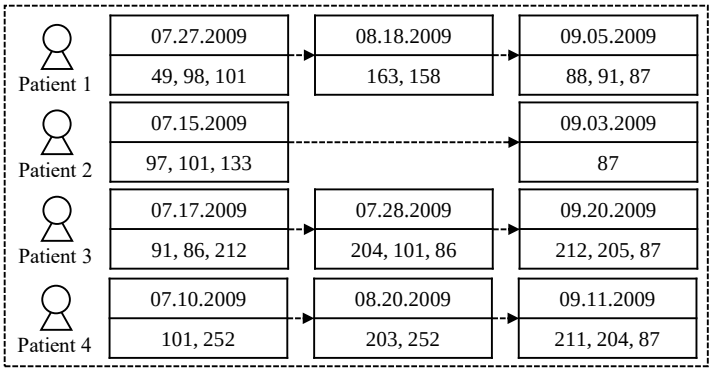
disease-specific studies[21]. However, FuzzyGap can be used for any structured information found in EHRs.

Sequence of Encounters Representation. Proper representation of the sequence of encounters is an important step as using the encounter data directly can pose problems for sequential pattern mining. We propose 5 different patient representations to capture a range of temporal information present in the claims data. A toy example of each variation is illustrated in Figure 3. Figure 3(a) is the original encounter sequence and Figures 3(b)-(e) are the different representations, which are further described below.

1. Flattened Encounter Sequence. One simple method for patient representation is to remove all the temporal information by flattening all the encounters into a single claim (or set). Thus for each patient, the set contains the combination of CCS codes present in all their codes. While a single CCS code itself may or may not represent the patient's condition, the combination of CCS codes can provide an improved representation of the medical condition of the patient. From Figure 3(a), we can see four patients each with 3 encounters and each encounter has a different number of CCS codes with some duplicates such as 86 in 'Patient 3'. Any duplicate codes are removed for the patient, and thus the flattened representation is simply the union of all CCS codes, shown in Figure 3(b). With this representation, the dataset is similar to the "transaction-style" representation that is used in frequent itemset mining. Therefore, using this representation we can extract the combination of frequent CCS codes that are representative of each class. A major limitation of flattening the sequence of encounters is the loss of information both in the number of occurrences of a CCS code as well as the order of the CCS codes, which can play an important factor in clinical pathways.

2. Event-Preserving Sequence. An alternative to the flattened representation is to encode the sequence of events. The encounter for each patient can be viewed as a list of visits ordered by the date. Therefore, each visit (or date) is simply a new event. Unfortunately, the number of visits is not uniform across all the patients and can cause alignment issues for sequential pattern mining. To resolve this issue, we propose to shift all events to be right aligned. This is because more recent visits are more important than previous ones[14], thus extracting representative patterns from the start may not yield as discriminative patterns as those extracted based on the last visit. As shown in Figure 3(c), since Patient 2 has only 2 visits, there is an empty cell in this alignment scheme. Therefore, all the patterns will always include information from the last visit.

3. Interval-Based Sequence. The number of visits for each patient can vary drastically and lead to low support counts for sequential pattern mining. Extracted patterns maybe too unique and not representative of other patients. Moreover, patients with a small number of visits are ignored due to their short sequences. Thus, we propose an interval-based



(a) Clinical pathways

	CCS codes
Patient 1	49, 98, 101, 163, 158, 88, 91, 87
Patient 2	97, 101, 133, 87
Patient 3	91, 86, 212, 204, 101, 212, 205, 87
Patient 4	101, 252, 203, 211, 204, 87

(b) Flatten encounter sequence

	Date 1	Date 2	Date 3
Patient 1	49, 98, 101	163, 158	88, 91, 87
Patient 2		97, 101, 133	87
Patient 3	91, 86, 212	204, 101, 86	212, 205, 87
Patient 4	101, 252	203, 252	211, 204, 87

(c) Event-preserving sequence

	Interval 1	Interval 2	Interval 3
Patient 1	49, 98, 101	163, 158	88, 91, 87
Patient 2		97, 101, 133	87
Patient 3		91, 86, 212, 204, 101	212, 205, 87
Patient 4	101, 252	203, 252	211, 204, 87

(d) Interval-based sequence

	Interval 1	Interval 2	Interval 3
Patient 1	49, 98, 101	49, 98, 101, 163, 158, 88, 91, 87	88, 91, 87
Patient 2		97, 101, 133, 87	87
Patient 3		91, 86, 212, 204, 101	204, 101, 86, 212, 205, 87
Patient 4	101, 252	203, 252	211, 204, 87

(e) FuzzyGap sequence

Figure 3: A toy example of sequence of encounters representation.

method that merges the encounters into a single visit within a specified interval. Figure 3(d) provides an example of setting the interval to be within the same month. Any duplicating CCS codes are removed such that a CCS code only appears once in each interval. Thus, the first two claims in 'Patient 3' are merged into the same group.

4. Gap-Sensitive Interval-Based Sequence. Given an interval-based sequence, there can be two approaches to obtain the pattern. Gap-sensitive models allow gaps to occur within the pattern while non-gapped models do not explicitly model the gap between the sequential events. As an example based on 'Patient 1' and 'Patient 4' from Figure 3(d), a non-gapped approach will receive the pattern $[\{101\}, \{87\}]$. Thus, it doesn't matter if CCS 101 occurs in the interval immediately preceding CCS 87 or several intervals before. However, a gap sensitive model would retrieve the pattern $[\{101\}, \{\}, \{87\}]$ which specifies that there needs to be a gap between 101 and 87. Gap-sensitive pattern mining models are useful in some tasks such as DNA sequence analysis or clickstream analysis[22]. Similar to these applications, in clinical pathways, modeling the gaps between each interval is important.

5. FuzzyGap Sequence. One problem with the gap-sensitive interval-based sequence representation is the hard constraints imposed by the specified interval. This is important as events close to the borderline of two intervals are forced into a specific interval. Moreover, claims that are close to each other can be considered as a single claim because of insufficient information[14]. Therefore, we propose a fuzzy interval representation. Given a user-specified boundary range, events within this value of the boundary between two intervals will be added to both intervals. For example, if we set the fuzzy range to 7 days, events within a 7 day period from the boundary will be added to both intervals. 'Patient 1' in Figure 3(a) illustrates this representation, where the last claim is close to the boundary when we set the interval to be in the same month. Thus, CCS codes 88, 91, and 87 are added to both 'Interval 2' and 'Interval 3' of 'Patient 1'.

Discriminative Clinical Pathways Extraction. Given an appropriate patient representation, the next step is to extract sequential patterns for each class (i.e., case and control). While there are several fast pattern growth algorithms that can discover sequential patterns[11], our preliminary experiments using several of these algorithms implemented in the SPMF library[23] either ran out of memory or failed to find any patterns. Thus, we discovered patterns by performing a sequential pairwise comparison between two patients and recording the intersection of their sequence representations. We also required that the pattern contain at least one CCS code from the last interval (or event). Although this can be computationally expensive, our algorithm can extract patterns from a large cohort (> 1000 patients).

Once the patterns are extracted from both classes, they are filtered to obtain only the discriminative patterns. This process is done by checking the existence of patterns from the dataset of the other class. There are two filtering options: (1) obtaining only pure patterns, or (2) specifying a ratio (threshold) which the patterns must exceed. Pure patterns mean selecting patterns that only exist in one class. While this is more discriminative, the number of pure patterns is limited as many of the discovered patterns exist in both classes. Thus, there can be no patterns returned at all in the worst case. Instead, we can set a threshold and allow some patterns to exist in the other class but have higher support count. While these patterns will be less distinctive than the pure patterns, it will return more patterns. For example, if there is a pattern in heart failure class with support count 100, and 25 in the non-heart failure class, if we set the threshold to 4, then the pattern will survive in the filtering process, and be a representative pattern in heart failure class. Usually, representative patterns have low support count, and most of the patterns with high support counts are filtered because of their commonness.

Risk Prediction Model. Once the discriminative patterns are extracted from the patients, we construct a new feature representation based on the patterns. The pattern itself may not represent the entire class, however, the combination of patterns can represent the class. Thus, we use the pattern existence as the features for a machine learning model. Suppose 4 patterns were extracted, $[p_1, p_2, p_3, p_4]$, then for each patient the feature is set based on the existence of that pattern. If the patient contains patterns p_2 and p_4, then the patient's new features will be $[0, 1, 0, 1]$. For pure patterns, patients in heart failure classes will only have 1's in heart failure patterns. However, if a threshold is used during the extraction process, the patient will have 1's in patterns from both classes.

Unfortunately, there are scenarios where patients may not exhibit any of the extracted patterns. In other words, the patients will have all features set to 0. Consequently, no machine learning model will be able to predict accurately as there is not enough information to determine whether the patient will have heart failure. Thus, for these patients, we use a different set of features which is the presence of the CCS codes. Similar to pattern features, we check whether the patient contains the CCS code, and set the feature as binary. For example, if there are 5 CCS codes, $[c_1, c_2, c_3, c_4, c_5]$, and the patient has c_1, c_2 and c_5, then we set the feature as $[1, 1, 0, 0, 1]$. Thus two separate machine learning models are learned, one for patients that have at least one pattern, and another based on patients who do not exhibit any patterns. For comparison purposes, we use a decision tree but note that the predictive model can potentially be any off-the-shelf machine learning technique.

Experiment Design

Dataset. We adopted a publicly accessible dataset provided by the Centers for Medicare and Medicaid Services[a] (CMS). CMS dataset is a synthesized data that was taken from 5% random sample of Medicare beneficiaries from

[a] https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF.html

2008 and their claims from 2008 to 2010. CMS dataset is split into 20 random samples and each sample contains 5 different parts, summary, inpatient, outpatient, carrier, and prescription. Since we are only focusing on CCS codes (obtained from ICD-9 codes), we use only the first four files. The summary file, while it does not contain ICD-9 codes, contains information about the chronic disease diagnosis for diabetes and heart failure. For example, if a patient was diagnosed with Heart Failure in 2010, then the summary file captures this information. The other three files contain the ICD-9 codes associated with a patient's encounter. Instead of fine-grained ICD-9 codes, CCS codes are used to group ICD-9 into broader categories to yield better interpretability of the patterns. For evaluation purposes, we created two training sets and one test set. The first training set contains 1000 randomly sampled patients from CMS with a 50-50 split for each class (heart failure or not heart failure) and is used to compare against existing sequential pattern mining algorithms. The second training set contains 20,000 randomly sampled patients with a balanced split to evaluate the computational feasibility on a large dataset. Patterns are only extracted from the training set. The test set contains 4000 randomly sampled patients with 2000 patients in each class.

Data Preprocessing. Our classification task was to predict diabetic patients who would develop HF in 2010. We first filtered the dataset to find patients with diabetes, before splitting them into two classes, HF, and non-HF. Patients who had HF prior to 2010 were omitted from this study. The remaining diabetic patients were labeled as HF if they were diagnosed with HF in 2010, otherwise, they were considered non-HF.

After labeling the patients, all the ICD-9 codes were collected from the claims dataset. For the HF patients, any events after the first encounter of an HF-related ICD-9 code (including that encounter) were removed. Once the post-HF events have been removed, all the ICD-9 codes are mapped to CCS codes to cluster similar ICD-9 codes into a category. The main reason for this grouping is that the diversity of ICD-9 codes is computationally prohibitive for pattern mining purposes.

Baseline. The 5 different patient representations are evaluated and compared against the results with the CloFast[24] algorithm. CloFast was chosen as it is the latest, fast pattern growth algorithm[24]. We used the implementation in the SPMF[23] library and note that the small training set (1000 patients) was based on CloFast's computational and memory footprint. To evaluate and compare between 5 different patient representations, we use a large training set (20000 patients). For sequential pattern mining, only data from the training set is used. Once patterns were extracted from each class, they were filtered based on the two options. All models are tested using decision tree in scikit-learn[25] library, and the prediction results are evaluated by AUC score. We provide additional details on the 8 different pattern extraction techniques:

- *Flatten representation:* We take the union of the CCS codes for each patient, and extract the patterns by finding the interaction of the codes between the patients. We also extract only the pure patterns.
- *Date representation:* The patients are right-aligned based on their last visit date. We use our extraction process to find patterns that contain at least one CCS code from the last visit and filter the patterns to be pure.
- *Interval sequence (non-gapped):* The interval is set to one month, where claims within the same month are recorded into a single event. A non-gapped pattern mining algorithm is run to extract the patterns and use only the pure patterns.
- *Interval sequence (gap-sensitive):* Similar to the non-gapped interval, a month interval is used. However, a gap-sensitive algorithm is used to model the gaps between the events, and only pure patterns are used.
- *FuzzyGap:* We use a one-month interval and a 7-day fuzzy window to construct the patient representation. Thus, if an encounter is within 7 days (before or after the 1st of the month), the CCS code for that encounter is included in both intervals. Only pure patterns are found.
- *FuzzyGap+ threshold:* Same as the previous setting, FuzzyGap, however, the threshold for filtering the patterns is set to 2 (at least 66% of the patients must be from HF or non-HF).
- *CloFast + Fuzzy + threshold:* We used CloFast algorithm on the Fuzzy gap representation. For the initial pattern search, we used 15% support count because CloFast algorithm searches for all non-gapped closed sequence

Table 1: Prediction results in AUC score and extracted subsequent patterns in each class using randomly sampled 1000 patients as train set and 4000 test set. In the model column, thr refers to the threshold which is set to 2. M1 is the model introduced in Figure 2, and M2 is the model that used both pattern features and CCS codes together for training. # of patterns summarizes the number of extracted discriminative patterns and the numbers inside the parenthesis indicates the total number of patterns found before the setting the threshold. # patients indicate the number of patients used in each model (pattern features or CCS codes features) in the test set.

Model	AUC		# of Patterns		# of Patients	
	M1	M2	HF	non-HF	Pattern	CCS
CCS code	0.5277	-	0 (0)	0 (0)	0	4000
Flatten	0.5328	0.5289	4 (794)	34 (395)	258	3742
Event-based	0.5331	0.5289	26 (7039)	111 (8827)	1124	2876
Interval-based (non-gapped)	0.5337	0.5277	5 (5646)	20 (5975)	245	3755
Interval-based (gap-sensitive)	0.5281	0.5314	71 (2537)	54 (1423)	810	3190
FuzzyGap	0.5439	0.5225	10 (672)	4 (223)	110	3890
FuzzyGap + thr	0.5454	0.5396	21 (672)	7 (223)	322	3678
CloFast + thr	0.5669	0.5401	0 (610098)	201806 (1233540)	4000	0
FuzzyGap + CloFast + thr	**0.5784**	0.5421	21 (610770)	201813 (1233763)	4000	0

patterns, which returns a huge amount of patterns. Since most of the patterns with high support counts are common patterns, we used $threshold = 2$ to filter and retrieve the final patterns for each class.

- *FuzzyGap+ CloFast + threshold:* Both extracted patterns from CloFast and FuzzyGapwith $threshold = 2$ are used to set the patient representation.

Results

First we evaluated the impact of the different patient representations and two different feature sets. Throughout our experiments, no other features are used besides pattern features and CCS codes feature to see the impact of our model. M1 is the model introduced in Figure 2, and M2 is the model that used both pattern features and CCS codes together for training. We tuned the decision tree parameters for each representation and pattern extraction combination via cross-validation to find the optimal depth, splitting criteria, and minimum number of samples per leaf.

Table 1 summarizes the predictive performance of the different representations and sequential pattern mining approaches on the 1000 patients in the train set and 4000 patients in the test set. For the different patient representations, FuzzyGap with a threshold of 2 provides the highest AUC whereas the CCS features only has the lowest AUC score. Also, the M2 column shows that using the combined features (CCS and the patterns) did not yield better predictive performance that using a separate model for patient with patterns and patients with only CCS features (M1). To determine the statistical significance of the AUC improvements, we performed multiple paired t-tests based on the M1 AUC scores. Using the null hypothesis that our model did not improve the predictive performance, we obtained the following results for each pair: (FuzzyGap + thr, FuzzyGap) = 1.17×10^{-7}, (FuzzyGap + thr, gap-sensitive) = 0.3122, (FuzzyGap + thr, non-gapped) = 1.04×10^{-9}, and (FuzzyGap + thr, event-based) = 7.15×10^{-10}. These results show that there is a performance improvement between FuzzyGap and other representations. We note that although the AUC score is low (0.5784), this is due to the size of the training set (illustrated later in this section) as well as the noisy nature of the CMS synthetic dataset.

Table 1 also summarizes the number of extracted discriminative patterns, with the values inside the parenthesis indicating the total number of patterns that are found before setting the threshold. For example, in HF column, 4 (794) means 4 patterns that satisfy the threshold are found out of 794 patterns. If the extracted patterns, $4 + 34$ in the case of Flatten, are not found in the test set, instead of the pattern feature, only CCS features are used to learn the model. The numbers of patients that use the pattern feature are shown in "Pattern" column and the number of patients that only use CCS features is shown in CCS column. Although not shown in Table 1, CloFast without setting a threshold above 0 returns no patterns after filtering. This indicates that no discriminative patterns are found. However, with a

Table 2: Prediction results in AUC score, and extracted subsequent patterns in each class using randomly sampled 20000 patients as train set and 4000 test set. M1 refers to the model explained in Figure 2.

Model	M1	# of Patterns		# of Patients	
		HF	non-HF	Pattern	CCS
Flatten	0.6282	960 (832721)	4114 (497681)	668	3332
Event-based	0.6206	1211 (1085910)	5743 (1279032)	1492	2508
Interval-based (non-gapped)	0.6206	633 (2091671)	2656 (2230431)	992	3008
Interval-based (gap-sensitive)	0.6273	21432 (1462878)	13077 (1070761)	3113	887
FuzzyGap	0.6321	2561 (558953)	967 (263550)	961	3039
FuzzyGap + thr	**0.6472**	5914 (558953)	2161 (263550)	3923	77

threshold set to 2, a large number of patterns are found from the non-HF patient cohort, while no patterns are found for HF. Moreover, the large number of patterns yields more features than samples, which suggests the potential for overfitting.

We note that our model (FuzzyGap) with a threshold yields a higher percentage of discriminative patterns. FuzzyGap + threshold model only uses 28 features, while CloFast + threshold uses 201,813 features. Once we add patterns from FuzzyGap + threshold with CloFast + threshold, which is only 28 patterns, it gives us about 1% increment in AUC score, which shows our models ability to identify distinct and unique patterns. The number of patterns of FuzzyGap are less than Interval (gap-sensitive) model because it is capturing unique patterns that are more representative. Even though FuzzyGap or FuzzyGap + threshold model uses less number of patterns as a feature, the AUC score is greater than Interval (gap-sensitive) model which uses more number of features. Thus, our representation helps find patterns with more support counts, mitigates against the consequences of setting a hard constraint for the boundary and can capture patterns that cover patients with a limited number of encounters. By relaxing the filtering to include less discriminative patterns, the pattern features are able to capture more patient information.

Table 2 summarizes the predictive performance on 20,000 patients in the train set and 4000 patients in the test set. In this experiment, CloFast was not able to extract patterns due to insufficient memory (on a machine with 32GB of RAM), and therefore not included in this experiment. For the different patient representations, similar to Table 1, FuzzyGap with a threshold of 2 provides the highest AUC. The AUC scores in Table 2 shows a significant improvement in AUC scores in all representations compare to Table 1. This is potentially due to the extraction of more patterns that allowed the framework to use more pattern features than CCS codes features. FuzzyGap with threshold in Table 1 has 322 patients in pattern feature model and 3678 in CCS codes feature model while Table 2 has 3923 patients in pattern feature model and 77 patients in CCS codes feature model. This result shows the effectiveness and the importance of using pattern features in the prediction process. Thus, FuzzyGap with threshold filtering not only is computationally efficient (from a pattern mining perspective) but can find discriminative patient features.

Figure 4 provides a visualization of the top two levels of the learned decision tree based on the FuzzyGap + CloFast + threshold representation. For ease of understanding, we have converted the CCS codes to their shorthand category description. The root node is the pattern that is one of the most important feature (or pattern) that is used for the initial split of the model. From the figure, we observe that the first pattern feature used consists of 7 CCS codes that span 7 months. According to the pattern, two "DiabMel no c" (diabetes mellitus with no complications) and three HTN (Hypertension) occur in a row with "Other screen" at the end. We also observe that back problem is a common pattern found in the second level, where the left node contains back problem at the beginning of the pattern while the other side is at the end of the problem.

Discussion and Conclusions

In this paper, we propose FuzzyGap and shown the effectiveness of our method. However, one major challenge is the computation cost of pattern extraction. Although patterns with low support count are preferred, existing algorithms are unavailable to extract patterns with low support count on a large cohort. Thus, in our work, we introduce a simple pairwise comparison between patients that are significantly cheaper in our training set for 1000 patients. As the number

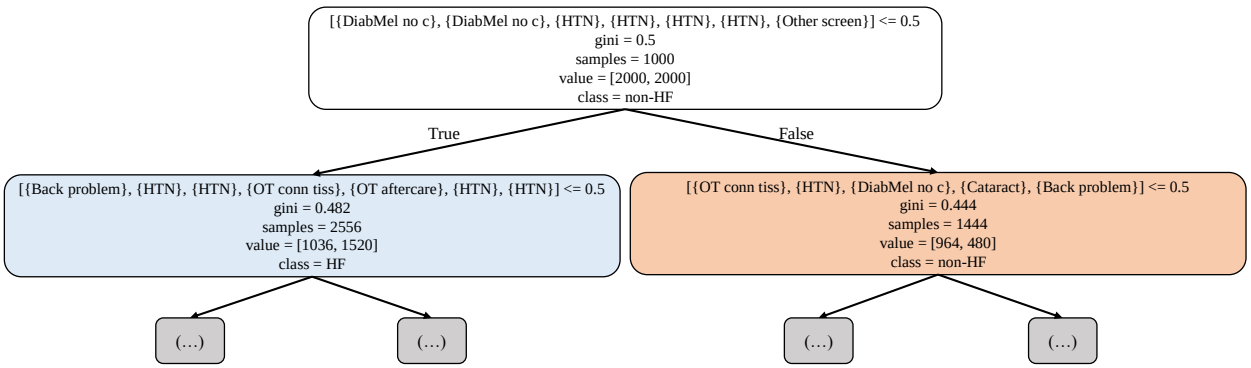


Figure 4: Visualization of top 2 levels of FuzzyGap + CloFast + threshold model.

of patients increases, the computational cost of pairwise comparison will increase, Therefore, a more computationally efficient method is necessary to extract discriminative patterns.

Furthermore, for a fair comparison, we evaluate our models with fixed training size in the experiment. However, there is a chance to increase or decrease the number of discriminative patterns extracted as the training set increases. An increased in the number of extract patterns can yield more discriminative patterns from the filtering process. Therefore, verifying the impact of changing the training size is also an important task.

One limitation of our case study is the use of the single source of patient information (ICD-9 code). One possible extension is to use more information that is available such as procedure coding and prescriptions. While there is nothing to restrict the use of FuzzyGap on multiple sources, the pattern extraction process will be significantly much more extensive. Nevertheless, FuzzyGap shows promising results, thus we leave this as a future work.

In conclusion, we presented FuzzyGap, a chronic heart failure prediction model by extracting discriminate patterns in clinical pathways. To resolve issues of patients encountering various length in clinical pathways, we propose a sequence of encounters representation to emphasize the last claim while merging or separating claims based on the time period of two visits. Overall, the prediction results show the effectiveness of discriminative patterns extracted from the proper sequence of encounters representation.

References

1. Davis J, Lantz E, Page D, Struyf J, Peissig P, Vidaillet H, et al. Machine learning for personalized medicine: Will this drug give me a heart attack. In: the Proceedings of International Conference on Machine Learning (ICML); 2008. .

2. Rotter T, Kinsman L, James E, Machotta A, Gothe H, Willis J, et al. Clinical pathways: effects on professional practice, patient outcomes, length of stay and hospital costs. Cochrane Database of Systematic Reviews. 2010;(3):CD006632–1.

3. Saint S, Hofer TP, Rose JS, Kaufman SR, McMahon JL. Use of critical pathways to improve efficiency: a cautionary tale. The American journal of managed care. 2003;9(11):758–765.

4. Zhang Y, Padman R. Innovations in chronic care delivery using data-driven clinical pathways. The American journal of managed care. 2015;21(12):e661–8.

5. Lee DS, Stitt A, Austin PC, Stukel TA, Schull MJ, Chong A, et al. Prediction of heart failure mortality in emergent care: a cohort study. Annals of internal medicine. 2012;156(11):767–775.

6. Batal I, Valizadegan H, Cooper GF, Hauskrecht M. A temporal pattern mining approach for classifying electronic health record data. ACM Transactions on Intelligent Systems and Technology (TIST). 2013;4(4):63.

7. Allen JF, et al. Towards a general theory of action and time. Artificial intelligence. 1984;23(2):123–154.

8. Khoshnevisan F, Ivy J, Capan M, Arnold R, Huddleston JM, Chi M. Recent Temporal Pattern Mining for Septic Shock Early Prediction;.

9. Zhang Y, Padman R, Patel N. Paving the COWpath: Learning and visualizing clinical pathways from electronic health record data. Journal of biomedical informatics. 2015;58:186–197.

10. Srikant R, Agrawal R. Mining sequential patterns: Generalizations and performance improvements. In: International Conference on Extending Database Technology. Springer; 1996. p. 1–17.

11. Fournier-Viger P, Lin JCW, Kiran RU, Koh YS, Thomas R. A survey of sequential pattern mining. Data Science and Pattern Recognition. 2017;1(1):54–77.

12. Zaki MJ. SPADE: An efficient algorithm for mining frequent sequences. Machine learning. 2001;42(1-2):31–60.

13. Pei J, Han J, Mortazavi-Asl B, Wang J, Pinto H, Chen Q, et al. Mining sequential patterns by pattern-growth: The prefixspan approach. IEEE Transactions on Knowledge & Data Engineering. 2004;(11):1424–1440.

14. Huang Z, Lu X, Duan H. On mining clinical pathway patterns from medical behaviors. Artificial intelligence in medicine. 2012;56(1):35–50.

15. Centers for Disease Control and Prevention. National diabetes statistics report, 2017. Atlanta, GA: Centers for Disease Control and Prevention; 2017.

16. Dieren Sv, Beulens JWJ, Schouw YTvd, Grobbee DE, Neal B. The global burden of diabetes and its complications: an emerging pandemic. European Journal of Cardiovascular Prevention & Rehabilitation. 2010 May;17(1_suppl):s3–s8.

17. Heidenreich PA, Trogdon JG, Khavjou OA, Butler J, Dracup K, Ezekowitz MD, et al. Forecasting the future of cardiovascular disease in the United States: a policy statement from the American Heart Association. Circulation. 2011 Mar;123(8):933–944.

18. Roger VL, Go AS, Lloyd-Jones DM, Benjamin EJ, Berry JD, Borden WB, et al. Heart disease and stroke statistics–2012 update: a report from the American Heart Association. Circulation. 2012 Jan;125(1):e2–e220.

19. American Heart Association. Cardiovascular disease & diabetes; 2017. http://www.heart.org/HEARTORG/Conditions/More/Diabetes/WhyDiabetesMatters/Cardiovascular-Disease-Diabetes_UCM_313865_Article.jsp#.WckMddOGMUE.

20. Feldman DI, Valero-Elizondo J, Salami JA, Rana JS, Ogunmoroti O, Osondu CU, et al. Favorable cardiovascular risk factor profile is associated with lower healthcare expenditure and resource utilization among adults with diabetes mellitus free of established cardiovascular disease: 2012 Medical Expenditure Panel Survey (MEPS). Atherosclerosis. 2017 Mar;258:79–83.

21. Geraci JM, Ashton CM, Kuykendall DH, Johnson ML, Wu L. International Classification of Diseases, 9th Revision, Clinical Modification codes in discharge abstracts are poor measures of complication occurrence in medical inpatients. Medical care. 1997;p. 589–602.

22. Han J, Pei J, Yan X. Sequential pattern mining by pattern-growth: Principles and extensions. In: Foundations and Advances in Data Mining. Springer; 2005. p. 183–220.

23. Fournier-Viger P, Lin JCW, Gomariz A, Gueniche T, Soltani A, Deng Z, et al. The SPMF open-source data mining library version 2. In: Joint European conference on machine learning and knowledge discovery in databases. Springer; 2016. p. 36–40.

24. Fumarola F, Lanotte PF, Ceci M, Malerba D. CloFAST: closed sequential pattern mining using sparse and vertical id-lists. Knowledge and Information Systems. 2016;48(2):429–463.

25. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825–2830.