Reduction of Readmissions to Hospitals Based on Actionable Knowledge Discovery and Personalization

Mamoun Almardini[1], Ayman Hajja[1], Zbigniew W. Raś[1,2(✉)], Lina Clover[3], David Olaleye[3], Youngjin Park[3], Jay Paulson[3], and Yang Xiao[3]

[1] College of Computing and Informatics, University of North Carolina,
Charlotte, NC 28223, USA
{malmardi,ahajja,ras}@uncc.edu
[2] Institute of Computer Science, Warsaw University of Technology,
00-665 Warsaw, Poland
[3] SAS Institute Inc, Cary, NC 27513, USA
{Lina.Clover,David.Olaleye,Youngjin.Park,Jay.Paulson,Yang.Xiao}@sas.com

Abstract. In this work, we define procedure paths as the sequence of procedures that a given patient undertakes to reach a desired treatment. In addition to its value as a mean to inform the patient of his or her course of treatment, being able to identify and anticipate procedure paths for new patients is an essential task for examining and evaluating the entire course of treatments in advance, and ultimately rectifying undesired procedure paths accordingly. In this paper, we first introduce two approaches for anticipating the state of the patient that he or she will end up in after performing some procedure p; the state of the patient will consequently indicate the following procedure that the patient is most likely to undergo. By clustering patients into subgroups that exhibit similar properties, we improve the predictability of their procedure paths, which we evaluate by calculating the entropy to measure the level of predictability of following procedure. The clustering approach used is essentially a way of personalizing patients according to their properties. The approach used in this work is entirely novel and was designed specifically to address the twofold problem of first being able to predict following procedures for new patients with high accuracy, and secondly being able to construct such groupings in a way that allows us to identify exactly what it means to transition from one cluster to another. Then, we further devise a metric system that will evaluate the level of desirability for procedures along procedure paths, which we would subsequently map to a metric system for the extracted clusters. This will allow us to find desired transitions between patients in clusters, which would result in reducing the number of anticipated readmissions for new patients. AQ1

Keywords: Personalization · Side-effects · Clustering

S. Kozielski et al. (Eds.): BDAS 2016, CCIS 613, pp. 1–17, 2016.
DOI: 10.1007/978-3-319-34099-9_3

1 Introduction

Recently, expenditure on healthcare has risen rapidly in the United States. According [1], healthcare spending has been rising at twice the rate of growth of our income, for the past 40 years; the projection of the growth rate in healthcare spending is 5.8 percent during the period 2014–2024, which means that the spending will rise to 5.4 trillion by 2024. That said, the gross domestic product (GDP) growth rate is 4.7 percent (as of 2014) [2]. This increase can be attributed to several factors as listed by Price Waterhouse Coopers (PWC) research institute: over-testing, processing claims, ignoring doctors orders, ineffective use of technology, hospital readmissions, medical errors, unnecessary ER visits, and hospital acquired infections [3]. Figure 1 shows that 25 billion are spent annually on readmissions. Hospital readmissions and surgery outcomes prediction has taken a great interest recently [4–7]. Analyzing the reasons behind readmissions and reducing them can save a great amount of money. A hospital readmission is defined as a hospitalization of the patient after being discharged from the hospital. The period in average is 30 days [7].

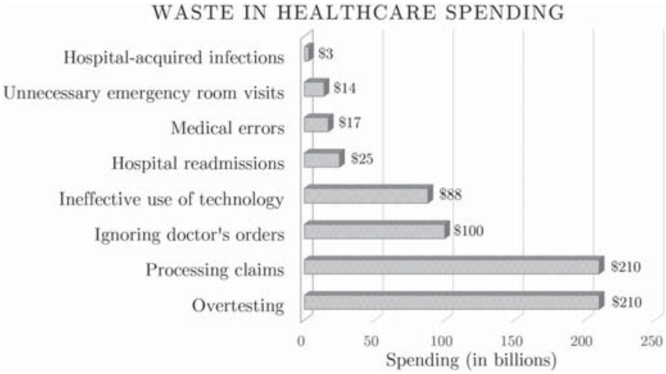


Fig. 1. Waste in healthcare spending as listed by Price Waterhouse Coopers (PWC) research institute [3]

One of the reasons for readmissions is negative side effects that may appear after the prescribed procedures and may not be known in advance, as a result patients may require hospital readmissions [8]. In this work, we shift our interests to the entire course of treatments for patients, and we propose ways to increase the predictability of what we call *procedure paths*, which is the sequence of procedures that a given patient undertakes to reach a desired treatment. By being able to predict the procedures that a given patient is expected to go through according to his or her diagnoses, domain experts can reevaluate the procedure path and possibly alter it to a more desired path.

2 HCUP Dataset Description

In this paper, we used the Florida State Inpatient Databases (SID) that is part of the Healthcare Cost and Utilization Project (HCUP) [13]. The Florida SID dataset contains records from several hospitals in the Florida State. It contains over 7.8 million visit discharges from over 3.6 million patients. The dataset is composed of five tables, namely: AHAL, CHGH, GRPS, SEVERITY, and CORE. The main table used in this work is the *Core* table. The *Core* table contains over 280 features; however, many of those features are repeated with different codification schemes. In the following experiments, we used the Clinical Classifications Software (CCS) [14] that consists of 262 diagnosis categories, and 234 procedure categories. This system is based on ICD-9-CM codes. In our experiments, we only used the features, listed in Table 1, that are relevant to the problem. Each record in the *Core* table represents a visit discharge. A patient may have several visits in the table. One of the most important features of this table is the *VisitLink* feature, which describes the patient's ID. Another important feature is the *Key*, which is the primary key of the table that identifies unique visits for the patients and links to the other tables. As mentioned earlier, a *VisitLink* might map to multiple *Key* in the database. This table reports up to 31 diagnoses per discharge as it has 31 diagnosis columns. However, patients' diagnoses are stored in a random order in this table. For example, if a particular patient visits the hospital twice with heart failure, the first visit discharge may report a heart failure diagnosis at diagnosis column number 10, and the second visit discharge may report a heart failure diagnosis at diagnosis column number 22. Furthermore, it is worth mentioning that it is often the case that patients examination returns less than 31 diagnoses. The *Core* table also contains 31 columns describing up to 31 procedures that the patient went through. Even though a patient might have gone through several procedure in a given visit, the primary procedure that occurred at the visit discharge is assumed to be the first procedure column. The *Core* table also contains a feature called *DaysToEvent*, which describes the number of days that passed between the admission to the hospital and the procedure day. This field is anonymized in order to hide the patients' identity. Furthermore, the *Core* table also contains a feature called *DIED*, that informs us on whether the patient died or survived in the hospital for a particular discharge. There are several demographic data that are reported in this table as well, such as race, age range, sex, living area, etc. Table 1 maps the features from the *Core* table to the concepts and notations used in this paper.

Table 1. Description of the used core table features.

Features	Concepts
VisitLink	Patient Identifier
DaysToEvent	Temporal visit ordering
DXCCSn	n^{th} Diagnosis, flexible feature
PRCCSn	n^{th} Procedure, meta-action
Race, Age Range, Sex,..	Stable features
DIED	Decision feature value

3 Background

The idea of extracting knowledge for the purpose of guiding decision makers to make more educated decisions is certainly not an all-new concept in data mining; however, the rather new idea of extracting actionable knowledge, being-in-itself a guide that serves as a blueprint for decision makers, has only been studied recently [10,11]. The concept and motivation of actionability is based on providing another, yet more relevant, layer of knowledge to decision makers. Actionable rules specify the actions needed to be performed to transition an instance (patient in our case) from one state to another.

One main area of research that heavily involves extracting actionable patterns from mining knowledge is action rules [11]. Action rules describe the necessary transitions that need to be applied on the classification part of a rule for other desired transitions on the decision part of a rule to occur. It is often the case however, that decision makers do not have immediate control over specific transitions, instead they have control over a higher level of actions, which in the literature of action rules are referred to by the term *meta-actions* [9]. Meta-actions are defined as higher level procedures that trigger changes in flexible features of a rule either directly or indirectly according to the influence matrix [9]. For example, by extracting action rules, we may reach the conclusion that to improve a patient's condition, we would need to decrease his (or her) blood pressure. Although this may seem as an oversimplified example of action rules, it is still nonetheless essentially what action rules are meant and designed to do; that is, to provide actionable patterns of transitions. Note here however that to perform the required transition of lowering the blood pressure of the patient, we would ultimately need to perform other actions of higher-levels, called meta-actions. For this particular example, perhaps this means performing a surgery on the patient or prescribing some medication. In summary, mining actionable rules is not only the study of discovering patterns, rather it is more centered around finding ways to transition instances from one pattern to another in a way that aligns with the desires of the domain being studied.

In this work, we apply the concept of mining actionable patterns to the domain of healthcare; by identifying the level of desire for procedure paths as will be discussed in Sect. 6, we will set the foundation that will allow us to extract actionable patterns that will transition new patients from a less desired procedure path to a more desired one.

4 Introducing Procedure Paths and Personalization

Procedure paths are defined as the sequence of procedures that a given patient undertakes to reach a desired treatment. In other words, a procedure path is a detailed description for the course of treatments provided to an admitted patient. The length of any given procedure path is an indicator of the number of readmissions that occurred or will occur throughout the course of treatment.

For example, one procedure path for a patient could be the following: $path_x = (p_1, p_3, p_3, p_6)$, where p indicates a particular procedure; according to procedure path $path_x$, the number of readmissions was 3.

In this work, we lay the foundation for predicting procedure paths by devising a system that will anticipate the following procedure (or readmission); we will also introduce a way to extract action rules that will describe transitions that will rectify the following procedure for new patients.

Although there could exist multiple metric systems for evaluating procedure paths, we decided in this work to devote our efforts on tackling the problem of reducing the number of readmissions for new patients. This means that we are interested in transitioning patients from a procedure path with more readmissions to another more desired path with less number of readmissions. That being said, our system for extracting action rules that describe transitions between procedure paths is entirely independent of the metric system used; this means that if domain experts decide to devise a new metric system by incorporating other criteria such as the cost of operation or duration in the hospital, then that would not affect our system and it would still function according to the new evaluation system.

The *procedure graph* for some procedure p is defined as the tree of all possible procedure paths extracted from our dataset for patients who underwent procedure p as their first procedure. The number of all procedure paths is extremely high. This high number of unique procedure paths indicates that it is not true that there exists a single universal course of treatment that patients typically follow to reach the desired state. For example, the number of patients that underwent

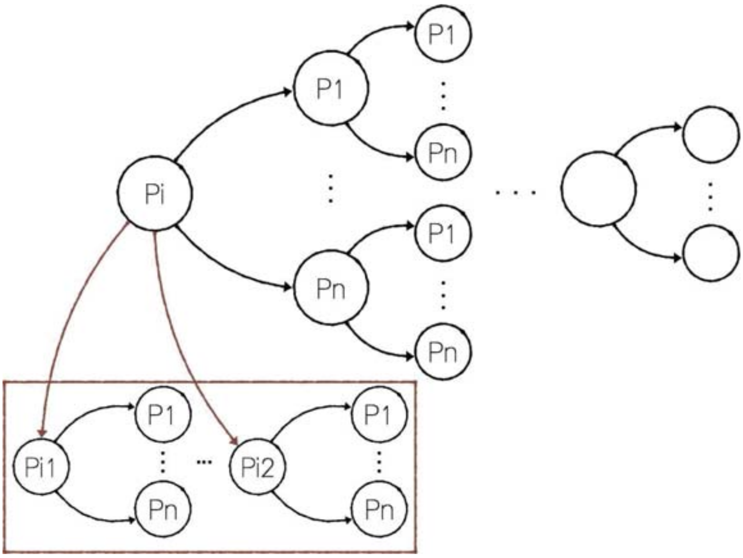


Fig. 2. Depiction of a procedure graph

procedure 78 (colorectal resection) is 41,753; and the number of unique procedure paths that those patients underwent is 6,774. Figure 2 shows a depiction of the *procedure graph*; $P_{(0,1)}$ is the initial procedure that patients start with; the next procedure could be any procedure from $P_{(1,1)}$ to $P_{(1,n)}$, which is determined by the resulting set of diagnoses after performing the initial procedure $P_{(0,1)}$. The first argument x in the notation $P_{(x,y)}$ refers to the number (or rather level) of readmission, and the second argument y refers to the procedure identifier at that level. For example, $P_{(1,2)}$ refers to the procedure with identifier 2 that occurred at the first level of readmissions (e.g. first readmission following the initial procedure). The portions of the graph that are contained in dashed boxes depict the personalization part that we introduce in the next section. The idea of personalization is to cluster patients that are scheduled to undergo procedure $P_{(0,1)}$ according to their diagnoses; as a result of this clustering, we will be able to anticipate with higher accuracy the following procedure (readmission) that the patient will undergo by identifying which cluster the new patient belongs to. In the next section, we will provide an elaborate description of the clustering approaches we propose.

5 Personalizing Through Diagnoses Clustering

Given a new patient, being able to anticipate the procedure path for that patient is an invaluable asset to medical doctors since it can be used as a mean to inform the patient of his or her course of treatment, and ultimately altering or amending the course of treatment accordingly. Our assumption that we use in this work, which aligns with the definition of our dataset, is that the procedure performed on the patient is determined from the set of diagnoses that the patient is diagnosed with. In addition to that, the set of diagnoses will also determine the state in which the patient ends up in; in this case, the state would be the set of diagnoses after performing the first procedure, which as result, will determine the second procedure.

Although it is theoretically possible to create a chain of predictions that will provide a complete prediction for the entire procedure path, we only examine predicting the following (next) procedure in this work for higher prediction accuracy; if the following procedure is part of a desired procedure path then no intervention is needed; otherwise, transitioning patients to another state that alters the following procedure to make it part of a more desired procedure path would be recommended if at all feasible.

To predict the procedure path, or rather the following procedure, we start by extracting knowledge from our existing dataset. The approach used to predict the following procedure is an unsupervised clustering technique based on the set of diagnoses that patients exhibit at the time of their first admission. Our assumption is that patients that exhibit similar set of diagnoses will end up with a similar set of diagnoses after the procedure; and again, by definition of our dataset, these set of diagnoses will determine the next procedure. Next, we provide two different clustering approaches; the exact matching clustering approach and the fuzzy approach.

5.1 Exact Matching Clustering

In the exact match clustering approach, we define a cluster by a set of diagnoses. For a given patient to belong to any cluster, he or she must have the exact same diagnoses set.

Table 2 shows some of the most common sets of exact diagnoses for patients who undertook procedure ‘158’ (spinal fusion). From Table 2, we observe that for procedure 158, the cluster with the exact set of diagnoses {205} contains 502 patients; this means that there are 502 patients that were diagnosed with ‘205’ and nothing else, before undertaking procedure 158. Similarly, the second row of Table 2 means that there are 128 patients that were diagnosed with both diagnostic codes ‘205’ and ‘98’ and nothing else, before having to undergo procedure 158.

Table 2. Some of the most common set of exact diagnoses for procedure 158 (spinal fusion)

Set of diagnoses	Number of patients	Entropy
{205}	502	3.015
{205, 98}	128	2.752
{205, 663}	86	2.543
{205, 209}	67	2.798
{205, 211}	51	2.510

Here is a description of each diagnostic code provided in the table:

- 205: Spondylosis; intervertebral disc disorders; other back problems
- 98: Essential hypertension
- 663: Screening and history of mental health and substance abuse codes
- 209: Other acquired deformities
- 211: Other connective tissue disease

We should also mention here that the entropy for the entire system before this personalization attempt is 3.667, which is higher than all the entropy values in the table. The weighted entropy for all exact matching clusters that have size above threshold 50 is 2.892. This implies that by applying this clustering approach, we will be able to have a higher level of predictability of which following (next) procedure is likely to be undertaken. In other words, by knowing the cluster of which a patient belongs to, we would be able to anticipate (with higher accuracy), where that patient is likely to end up after performing the first procedure.

One advantage of using the exact matching clustering is that transitions can be precisely described. For example, if we discovered that patients in cluster $c_1 = \{205\}$ tend to end up in a state that is more desired than patients in cluster $c_2 = \{205, 98\}$, then we can precisely describe the transition that needs to be done; in this case, only removing diagnostic code 98.

The fact that patients are usually admitted with other diagnoses that are often irrelevant to the main diagnosis (or diagnoses), makes this approach quite limited, which is rather evident in the frequencies (number of patients) that exhibit the most common set of exact diagnoses, compared to the number of patients that exhibit the same diagnoses but along with other diagnoses that may be irrelevant. For example, the number of patients that exhibit diagnosis code 205 along with other diagnoses is 13,096, which is substantially larger than the number of patients that only exhibit diagnosis code 205. In the next subsection, we present a new clustering approach that addresses this limitation.

5.2 Fuzzy Matching Clustering

In this section, we define a new and novel way for specifying the criteria that patients' properties need to satisfy for a given patient to join a cluster. Unlike the exact matching clustering approach where we define one unique set of diagnoses that needs to exactly match the patient's set of diagnoses; in this fuzzy matching clustering approach, we define three sets of diagnoses that will determine whether a patient belongs to a particular cluster or not.

The first set, which we call the *included* set, describes the set of diagnoses that any given patient needs to exhibit for that patient to belong to the cluster; the second set, which we call the *excluded* set, is the set of diagnoses that patients cannot exhibit for them to belong to that cluster; and finally the third set, which we call the *optional* set, is the set of diagnoses that patients can, but do not need to, exhibit for them to belong to that cluster.

Since the *optional* set is the complement of the *included* and *excluded* sets combined, we decided not to specify it each time we define a cluster. For example, if the entire set of diagnoses is $D = \{d_1, d_2, ..., d_{10}\}$, the *included* set of some cluster c is $included(c) = \{d_1, d_2, d_5, d_7\}$, and the *excluded* set of the same cluster c is $excluded(c) = \{d_6, d_8, d_9, d_{10}\}$, then the *optional* set for cluster c is $optional(c) = \{d_3, d_4\}$; which is equal to $D - [included(c) \cup excluded(c)]$.

Now let's examine one real example extracted from our data set. By examining the same procedure (code 158; spinal fusion), one of the extracted clusters was: $c = \{(included : \{98\}, excluded : \{49, 138, 211\})\}$, which contained all patients that exhibited diagnostic code 98 (essential hypertension) and did not exhibit diagnostic codes 49 (diabetes mellitus without complication), 138 (esophageal disorders) and 211(other connective tissue disease).

In our approach of selecting fit diagnostic codes candidates, we started by discarding the diagnostic codes that were either always or never evident in the set of diagnoses; although this may seem counterintuitive at first, the reasoning behind doing so is rather logical. Recall again that our goal is to create clusters of patients that are similar in terms of the resulting states after applying a given procedure. Clearly, if all patients have a common diagnosis, then this diagnosis will not play any role in determining the state for which patients end up in. Accordingly, we only considered diagnostic codes that lie within a specific range; for example, when we choose the allowed range to be between 20 % and 80 %, this means that we only consider diagnoses for which the number of patients that exhibit that diagnosis is between 20 % and 80 %; this procedure is also applied to the *excluded* set, meaning that we also only consider diagnoses that were missing from 20 % to 80 % of the total number of patients. Table 3 shows different ranges that we tested, with different outcomes that aligns with our expectations.

Table 3. Number of clusters and the entropy for different element clusters and different ranges for procedure 158 (spinal fusion)

	Range					
	20 % to 80 %		10 % to 90 %		5 % to 95 %	
	# of clusters	Entropy	# of clusters	Entropy	# of clusters	Entropy
1-element clusters	14	5.105	36	5.109	66	5.117
2-element clusters	37	5.051	379	5.028	1532	5.041
3-element clusters	50	4.988	2097	4.958	19167	4.971
4-element clusters	44	4.916	6969	4.905	155028	4.91

The main reason why this fuzzy approach is more superior than the exact matching approach is because it is not based on the assumption that each diagnosis must be relevant to the procedure. However, note that although this fuzzy approach was designed to be less strict so that it counteracts or rectifies the main disadvantage of the exact matching approach, this approach is in fact still precise enough to describe transitions rather specifically for a patient to transition from one cluster to another, as a result of the way our clusters are defined.

For example, we can achieve the transition from cluster $c_1 = \{(included : \{98\}, excluded\{53, 211\})\}$ to cluster $c_2 = \{(included : \{\}, excluded\{53, 98, 211)\}$ by shifting diagnostic code 98 from the included set to the excluded set; which simply means applying some treatment to the patient so that he or she is no longer diagnosed with diagnostic code 98 (essential hypertension).

Here is a description of each diagnostic code provided in the two clusters from the example above:

- 53: Nutritional deficiencies
- 98: Essential hypertension
- 209: Other acquired deformities
- 211: Other connective tissue disease

The methodology used to extract all combinations of clusters is similar to the association action rules extracting approach presented in [12], we start by extracting all 1-diagnosis clusters that lie within the range specified (for both the *included* and *excluded* sets). Then we build 2-diagnosis clusters by combining all possible pairs of 1-diagnosis clusters. Next step would be to construct 3-diagnosis clusters by combining all the 2-diagnosis clusters with all the 1-diagnosis clusters, so on and so forth. Using this procedure however, the number of generated clusters will grow extremely fast. Iterating through all possible clusters for each patient to verify whether that patient belonged to a given cluster or not would be a highly inefficient implementation. Instead, we used a retrieval digital tree implementation that starts the verification process at the root of the tree, which would allow us to discard entire subtrees anytime the patient does not satisfy a node constraint (whether that constraint was an *included* constraint or an *excluded* constraint).

As mentioned earlier in this section, the main goal of clustering patients according to the diagnostic codes is to increase the predictability for the next procedure which can be measured by calculating the entropy. Constructing more combinations of the diagnostic codes enforces more personalization on the patients and as a result improves the predictability of the next procedure; the entropy decreases. Figure 3 illustrates the process of generating the clusters. The top of the tree, Level 1, shows the 1-element set clusters; which are represented by all possible diagnostic codes for a given procedure; *included sets* : $\{d1\}, \{d2\}, \text{and } \{d3\}$ and their negations; *excluded sets* : $\{-d1\}, \{-d2\}, \text{and } \{-d3\}$. The process continues and in order to generate clusters for any Level 'L_n', we pair all elements in Level '$L_{(n-1)}$' with elements from 'L_1'.

It is also worth mentioning here that any diagnostic code can not exist with its negation in the same cluster; for example, we can not have $\{d1\}$ and $\{-d1\}$ in the same cluster, as this means that the patients that belong to that cluster have the diagnostic code $d1$ and do not have it at the same time.

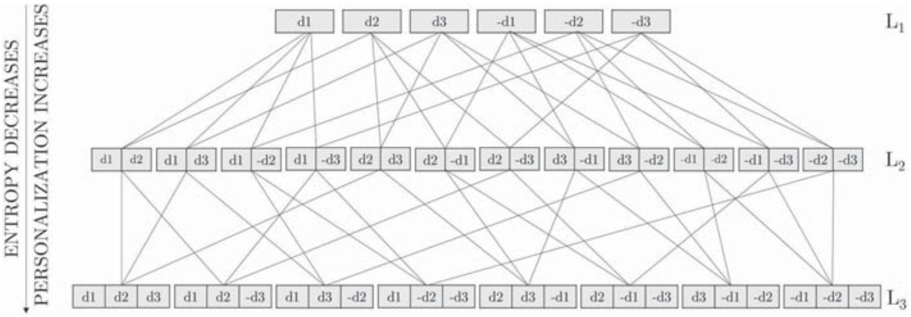


Fig. 3. Construction of the patients' clusters

Now let us show the benefits for clustering patients by providing a hypothetical example that mimics a real life scenario. In Fig. 4, we show a sample of some of the clusters that could be generated for the patients that are about to undertake procedure $P_{(0,1)}$ as their first procedure. The first cluster (Cluster 1) contains 60 patients; from the 60 patients that belong to Cluster 1, 10 out of which ended up undergoing procedure $P_{(1,1)}$, 45 ended up undergoing procedure $P_{(1,2)}$, only 5 patients ended up undergoing procedure $P_{(1,3)}$, and some patients didn't come back to the hospital. According to our example, this distribution of following procedures implies that if a patient exhibits the set of diagnoses that the first cluster is defined by, then that patient will most likely end up in a state that will require him or her to undergo procedure $P_{(1,2)}$. Similarly, the distribution of Cluster 2 and Cluster 3 will imply that most new patients that will belong to Cluster 2 will end up undergoing procedure $P_{(1,1)}$, and that most new patients that will belong to Cluster 3 will end up undergoing $P_{(1,3)}$. In the following section, we introduce a novel algorithm that assigns a score to each procedure in the procedure graph by taking into consideration the number of patients and the length of the procedure path. This procedures' score is later consequently used to calculate the score for each cluster to determine their level of desire, which would guide us to extracting cluster transitions.

6 Calculating Score

In this section, we propose a system to evaluate nodes (procedures) in a procedure graph by using the number of following anticipated readmissions as our criterion. We will propose subsequently, a score mapping system that will be used to transfer the scores of procedures in the procedure graph to clusters, which would thereafter allow us to identify feasible cluster transitions and ultimately reduce the average number of readmissions for new patients.

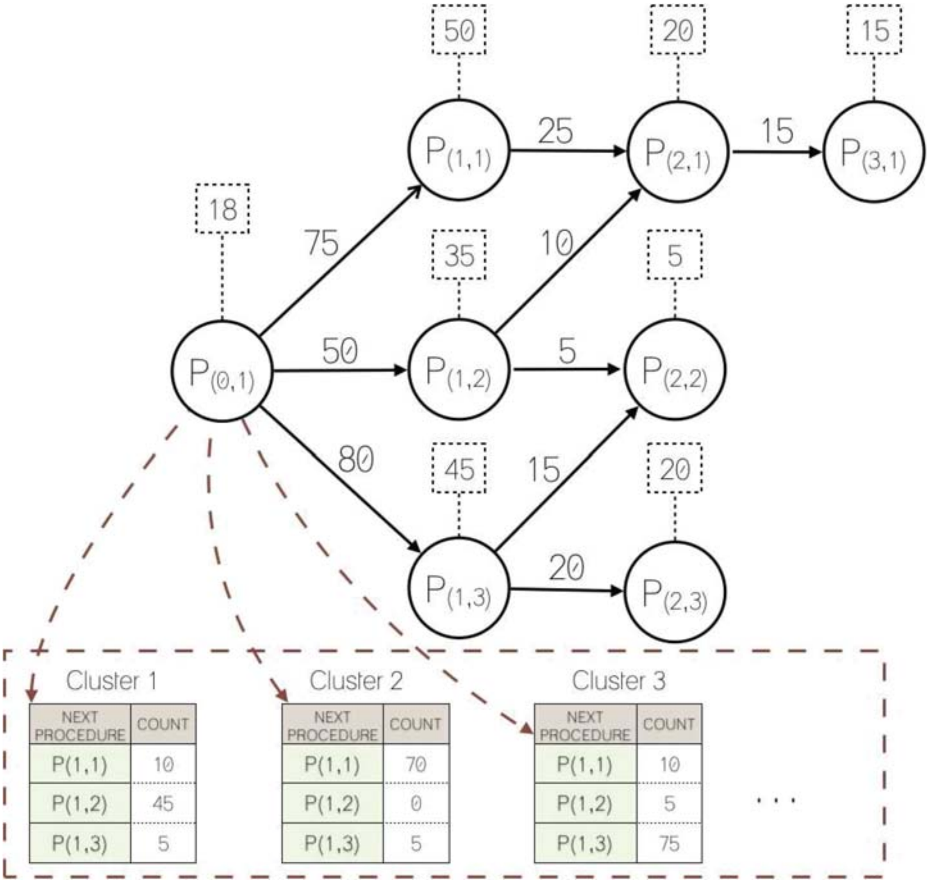
6.1 Procedure Graph

In Sect. 5, we introduced a method that will cluster patients according to their set of diagnoses; as a result, we were able to increase the predictability of the next procedure. This means that by examining the set of diagnoses for a new patient, we will be able to identify the cluster(s) that he or she belongs to, which would allow us as a result to increase the accuracy of predicting the following procedure. That being said, without having a metric system that will score the level of desirability for following procedures, we wouldn't be able to determine whether there exist(s) following procedures that are more desired than the anticipated following procedure. As we discussed in Sect. 4, our ultimate goal is to transition patients from an undesired anticipated procedure path to a more desired one; to do so however, we would need to devise a system that evaluates the level of desirability for any given procedure path.

In this section, we provide a metric system that will score each node (procedure) in the procedure graph defined in Sect. 4. There are two reasons why we decided to evaluate and score procedure graph nodes (actual procedures) rather than evaluating procedure paths. The first reason is because there is a significant number of procedure paths for any given procedure, and it would be inefficient to calculate the score for all of them; the second more important reason is the fact that by transitioning a patient from one cluster to another, we are essentially attempting to change the next procedure, which is a node in a procedure graph and not an entire procedure path.

The metric system devised in this section is based on the number of readmissions for patients. Next, we will introduce a method for calculating the score of any given node (procedure) in the procedure graph, which will represent the average number of following readmissions for patients at that particular procedure.

Let us demonstrate with an example by calculating the score for some nodes from our hypothetical procedure graph in Fig. 4. According to Fig. 4, 15 patients have undergone procedure $P_{(3,1)}$, and since there are no procedures following $P_{(3,1)}$, this means that all 15 patients have an average number of future readmission equal to 0; hence, the score of node $P_{(3,1)}$ will be equal to zero. Now let us examine node $P_{(2,1)}$; the number of patients that undergone $P_{(2,1)}$ is 35 (25 from $P_{(1,1)}$ and 10 from $P_{(1,2)}$), out of the 35, 20 patients did not come back to the hospital and 15 were readmitted to undergo procedure $P_{(3,1)}$. The score of procedure $P_{(2,1)}$ is the sum of weighted score of each possible procedure directly following the procedure $P_{(2,1)}$:



NEXT PROCEDURE	COUNT
P(1,1)	10
P(1,2)	45
P(1,3)	5

NEXT PROCEDURE	COUNT
P(1,1)	70
P(1,2)	0
P(1,3)	5

NEXT PROCEDURE	COUNT
P(1,1)	10
P(1,2)	5
P(1,3)	75

Fig. 4. Depiction of patients clustering

- The weight/probability of the first possibility (no further readmissions) is 20/35. The score (average number of readmissions) for patients who did not come back to the hospital is zero.
- The weight/probability of the second possibility (undergoing $P_{(3,1)}$) is 15/35, for which the score (average number of readmissions) will be 1 (which essentially reflects undergoing one more procedure) plus the score of $P_{(3,1)}$, which is zero for this example.

The score of procedure $P_{(2,1)}$ therefore becomes:

$$\text{score}(P_{(2,1)}) = \left(\frac{20}{35} * 0\right) + \left(\frac{15}{35} * \left(1 + \text{score}(P_{(3,1)})\right)\right) = \frac{15}{35} * (1 + 0) = .43$$

This essentially means that if a patient were to undergo procedure $P_{(2,1)}$, then the number of following readmission on average is .43; also, in this particular example, since we know that patients can only have one readmission ($P_{(3,1)}$), we can also state that since the score is .43, then this also means that for any patient who undertakes $P_{(1,2)}$, there will be a 43 % chance that he or she will undergo one additional readmission.

Now let us examine one more node: procedure $P_{(1,2)}$. The number of patients that underwent procedure $P_{(1,2)}$ is 50, from which we have three possibilities:

- **Possibility 1:** 35 out of 50 did not come back to the hospital.
- **Possibility 2:** 10 out of 50 were readmitted to undergo procedure $P_{(2,1)}$.
- **Possibility 3:** 5 out of 50 were readmitted to undergo procedure $P_{(2,2)}$.

To calculate the score in this case, we need to calculate the weighted score for each possible following procedure

- The weight/probability of the first possibility is 35/50; again however, the score (average number of readmissions) for patients who did not come back to the hospital is zero.
- The weight/probability of the second possibility is 10/50, for which the score (average number of readmissions) will be 1 (which essentially reflects undergoing $P_{(2,1)}$), plus the score of $P_{(2,1)}$.
- The weight/probability of the third possibility is 5/50, for which the score (average number of readmissions) will be 1 (which essentially reflects undergoing $P_{(2,2)}$), plus the score of $P_{(2,2)}$. Note here that the score of $P_{(2,2)}$ is zero since procedure $P_{(2,2)}$ was the last procedure for all patients that went through procedure $P_{(2,2)}$.

The score of procedure $P_{(1,2)}$ hence becomes:

$$\text{score}(P_{(1,2)}) = \left(\frac{35}{50} * 0\right) + \left(\frac{10}{50} * \left(1 + \text{score}(P_{(2,1)})\right)\right) + \left(\frac{5}{50} * \left(1 + \text{score}(P_{(2,2)})\right)\right)$$

$$\Rightarrow \text{score}(P_{(1,2)}) = 0 + \left(\frac{10}{50} * (1 + .43)\right) + \left(\frac{5}{50} * (1 + 0)\right) = .386$$

Which again, would mean that for patients that undergo procedure $P_{(1,2)}$, the number of following readmission on average is .386; this however does not mean that there is a 39 % chance that the patients will undergo additional readmissions, since a single patient may undergo two readmissions.

We define the procedure score recurrence function as:

$$\text{score}(P_x) = \begin{cases} \sum_{k=1}^{n} \frac{|P_k|}{|P_x|} * (1+\text{score}(P_k)) & \text{if } n \geq 1 \\ 0, & \text{otherwise} \end{cases}$$

where n denotes the number of procedures directly following procedure P_x, P_k denotes the kth procedure right after P_x, and $|P_k|$ denotes the number of patients that underwent the kth procedure.

Figure 5 shows the score for each node in our procedure graph.

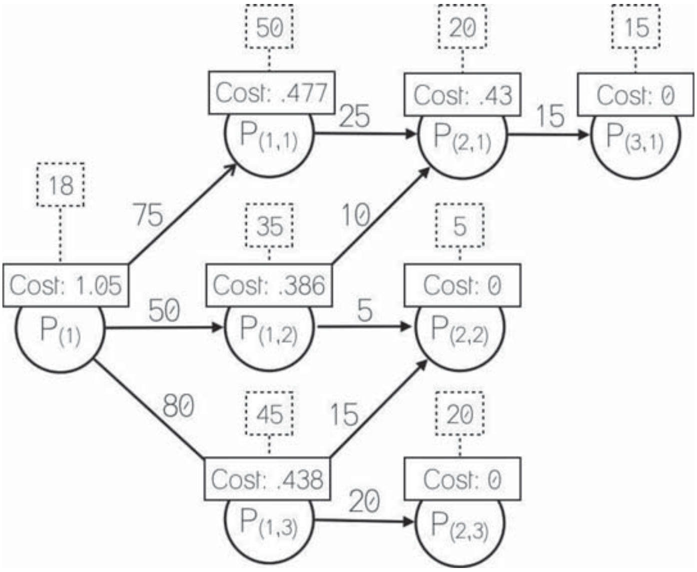


Fig. 5. Procedure graph with procedure scores

6.2 Calculating the Score for Clusters

By following the graph clustering approach, proposed in Sect. 5, we can identify the cluster(s) that the patient belongs to by only examining the set of diagnoses characterizing that patient. In this section, we will use the procedure graph metric system devised in the previous section to introduce a mapping between the scores of nodes in our procedure graph to the scores of clusters.

Since clusters contain patients that undergo the same procedure but the following procedures may differ for them, the score of a cluster is therefore defined as the sum of the weighted score of the procedures directly following that cluster. So, we define the score or cluster C_x as:

$$\text{score}(C_x) = \sum_{k=1}^{m} \frac{|P_{(x,k)}|}{m} * \text{score}(P_k)$$

where m denotes the total number of patients in Cluster C_x, and $|P_{(x,k)}|$ denotes the number of patients that underwent the kth next procedure for Cluster C_x. Clearly, $C_x = \bigcup\{P_k : 1 \leq k \leq n\}$.

For example, according to Fig. 4 shown in Sect. 5, the score for each cluster is calculated as follows:

- score(Cluster 1) $= \frac{10}{60} * .477 + \frac{45}{60} * .386 + \frac{5}{60} * .438 = .406$
- score(Cluster 2) $= \frac{70}{75} * .477 + \frac{0}{75} * .386 + \frac{5}{75} * .438 = .474$
- score(Cluster 3) $= \frac{10}{90} * .477 + \frac{5}{90} * .386 + \frac{75}{90} * .438 = .439$

According to the scores above, the most desired cluster would be Cluster 1. By transitioning patients from Cluster 2 (least desired) to Cluster 1 (most desired), we would be able to reduce the number of following readmissions by almost 7 %; meaning that we would decrease the total number of following readmissions for 100 patients by 7.

7 Results of Cluster Transitions from Our Dataset

In this section, we provide some of the results obtained after extracting clusters of patients from the Florida State Inpatient Databases (SID), followed by calculating their scores according to the definitions presented above. After calculating the scores for the extracted clusters, any clinically feasible transition from a cluster with a higher score to another cluster with a lower score would essentially be considered a valid action rule (applicable set of actions) that will help reduce the average number of following readmissions.

Recall that as the number of elements (diagnostic codes) defined in our clusters increases, the number of actual clusters extracted will also increase, as shown in Table 3. Also, as the number of clusters increases, the number of transitions between clusters will therefore increase as well. The number of cluster transitions for n clusters will be roughly $n(n-1)/2$, since the patients in the cluster with the highest score can transition to all other clusters ($n-1$ transitions), and the patients in the cluster with the second highest score can transition to all other clusters with lower score ($n-2$ transitions). Clearly, most of these transitions are not clinically valid; nevertheless, we believe that the number of clinically valid transitions is still substantial.

In this section, we examine few 3-element clusters extracted from procedure 44 (coronary artery bypass graft) using the range 10 % to 90 %. Table 4 shows three clusters extracted from patients who undertook procedure 44, which is the most common type of open-heart surgery, as their first procedure.

Table 4. A sample of three clusters with their scores and their included and excluded sets for procedure 44 (coronary artery bypass graft)

Name	# of Patients in Cluster	Included Set	Excluded Set	Score
Cluster 1	250	{}	{-53, -59, -238}	0.058
Cluster 2	258	{59, 49}	{-257}	0.073
Cluster 3	257	{238}	{-108, -663}	0.078

Here is a description of the diagnostic codes shown in Table 4:

- 49: Diabetes mellitus without complication
- 53: Nutritional deficiencies
- 59: Deficiency in the red blood cells
- 108: Congestive heart failure; nonhypertensive
- 238: Complications of surgical procedures or medical care
- 663: Screening and history of mental health and substance abuse codes

According to all extracted clusters, Cluster 1 (which has score 0.058 as shown in Table 4) is the best cluster for procedure 44 (amongst all 13,140 extracted clusters); this means that if a new patient were to belong to Cluster 1, then there is no transition that could reduce the anticipated number of readmissions for that patient. However, if a new patient belongs to any other cluster, then there is at least one transition that would reduce the anticipated number of following readmissions.

By examining Table 4, we can infer that a transition from Cluster 2 to Cluster 1 will reduce the number of following readmissions on average by 1.5 %; this means that by treating the diagnostic code 59 (shifting it from the included set to the excluded set) and making sure that the patients do not have diagnostic codes 53 and 238 before performing procedure 44, we would decrease the number of following readmissions by 3.87 for the 258 patients in Cluster 2.

Similarly, by transitioning patients from Cluster 3 to Cluster 1 (by treating the diagnostic code 238 and making sure that the patients do not have diagnostic codes 53 and 59 before performing procedure 44), we would decrease the number of following readmissions by 5.14 for the 257 patients in Cluster 3.

8 Conclusion

Predicting all possible paths that a new patient may undertake during his or her stay at a hospital is of great help to physicians; since it allows them to choose the best treatment in order to achieve the desired outcomes. In this research, we examined the problem of predicting the following procedure by proposing two novel approaches to personalize patients by clustering them into subgroups that exhibit similar properties. To evaluate our personalizing approach we calculated the entropy and showed that by personalization, the accuracy of predicting following procedures indeed increases. We then introduced and devised a metric system that evaluated nodes (procedures) in the procedure graph, followed by a mapping that will transfer the scores to the extracted clusters by calculating the entropy to measure the predictability of the next procedure.

References

1. Goodman, J.C.: Priceless: Curing the Healthcare Crisis. Independent Institute, Oakland (2012)
2. Keehan, S.P., et al.: National health expenditure projections, 2014–24: spending growth faster than recent trends. Health Aff. **34**(8), 1407–1417 (2015)
3. Pricewaterhouse Coopers: The Price of Excess: Identifying Waste in Healthcare Spending (2006)
4. Touati, H., Raś, Z.W., Studnicki, J., Wieczorkowska, A.A.: Mining surgical meta-actions effects with variable diagnoses' number. In: Andreasen, T., Christiansen, H., Cubero, J.-C., Raś, Z.W. (eds.) ISMIS 2014. LNCS, vol. 8502, pp. 254–263. Springer, Heidelberg (2014)
5. Lally, A., Bachi, S., Barborak, M.A., Buchanan, D.W., Chu-Carroll, J., Ferrucci, D.A., Glass, M.R., et al.: WatsonPaths: Scenario-based Question Answering and Inference over Unstructured Information. Technical Report Research Report RC25489. IBM Research (2014)
6. Tremblay, M.C., Berndt, D.J., Studnicki, J.: Feature selection for predicting surgical outcomes. In: Proceedings of the 39th Annual Hawaii International Conference System Sciences, HICSS 2006, vol. 5. IEEE (2006)
7. Silow-Carroll, S., Edwards, J.N., Lashbrook, A.: Reducing hospital readmissions: lessons from top-performing hospitals. Care Manage. **17**(5), 14 (2011) AQ2
8. Hajja, A., Touati, H., Raś, Z.W., Studnicki, J., Wieczorkowska, A.A.: Predicting negative side effects of surgeries through clustering. In: Appice, A., Ceci, M., Loglisci, C., Manco, G., Masciari, E., Ras, Z.W. (eds.) NFMCP 2014. LNCS, vol. 8983, pp. 41–55. Springer, Heidelberg (2015)
9. Raś, Z.W., Dardzińska, A.: Action rules discovery based on tree classifiers and meta-actions. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 66–75. Springer, Heidelberg (2009)
10. Touati, H., Raś, Z.W., Studnicki, J., Wieczorkowska, A.: Side effects analysis based on action sets for medical treatments. In: Proceedings of the Third ECML-PKDD Workshop on New Frontiers in Mining Complex Patterns, Nancy, France, pp. 172–183, 15–19 September 2014
11. Raś, Z.W., Wieczorkowska, A.A.: Action-rules: how to increase profit of a company. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 587–592. Springer, Heidelberg (2000)
12. Raś, Z.W., Dardzińska, A., Tsay, L.S., Wasyluk, H.: Association action rules. In: IEEE/ICDM Workshop on Mining Complex Data (MCD 2008), Pisa, Italy. ICDM Workshops Proceedings, pp. 283–290. IEEE Computer Society (2008)
13. HCUP-US: Overview Of The State Inpatient Databases (SID). Web. 12 February 2016
14. HCUP-US: Clinical Classifications Software (CCS). Web. 12 February 2016