OBJECT-DRIVEN AND TEMPORAL ACTION RULES MINING

by

Ayman Hajja

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2013

Approved by:

_____
Dr. Zbigniew W. Raś

_____
Dr. Jing Yang

_____
Dr. Wensheng Wu

_____
Dr. Wlodek Zadrozny

_____
Dr. James Studnicki

ABSTRACT

AYMAN HAJJA. Object-driven and temporal action rules mining. (Under the direction of DR. ZBIGNIEW W. RAŚ)

In this thesis, I present my complete research work in the field of action rules, more precisely object-driven and temporal action rules. The drive behind the introduction of object-driven and temporally based action rules is to bring forth an adapted approach to extract action rules from a subclass of systems that have a specific nature, in which instances are observed from assumingly different distributions (defined by an object attribute), and in which each instance is coupled with a time-stamp. In previous publications, we proposed an object-independency assumption that suggests extracting patterns from subsystems defined by unique objects, and then aggregating similar patterns amongst all objects. The motivation behind this approach is based on the fact that same-object observations share similar features that are not shared with other objects, and these features are possibly not explicitly included in our dataset. Therefore, by individualizing objects prior to calculating action rules, variance is reduced, and over-fitting is potentially avoided. In addition to the object-independency assumption, temporal information is exploited by taking into account only the state transitions that occurred in the valid direction.j

The common nature of object-driven and temporal action rules made us believe that this work is general enough to solve a diverse fields of areas where it is highly needed. In our case study, we show how our approach was applied to an information system of hypernasality patients; our results were further investigated by physicians

collaborators to confirm them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: DATA MINING AND EXPERT SYSTEMS

1.1 Motivation: From Concepts to Applications

Data mining, which is also referred to as Knowledge Discovery in Databases (KDD), can be succinctly defined as the process of extracting nontrivial, useful, and valid structural patterns of knowledge. By structural patterns of knowledge, we refer to valuable information extracted from historical data, which will normally be used as a guide (or blueprint) by domain experts to aid in the process of decision making. Data mining however, is an exceptionally broad subject; and though in the past, the problem definition was partly different, and the vision and future applications were barely apparent, some argue that the roots of data mining goes to as back as 1910. The term "scientific management", coined in 1910 by Frederick Winslow Taylor, refers to a system of measurements and analysis in which data is collected for the purpose of redesigning work environments to optimize efficiency; in a certain degree, this is what data mining is about.

Today however, after more than an entire century, the world in which we live has significantly transformed. The emergence of technology advancements that have been in development for the past few decades have made a profound impact on the way our digital data warehouses came to existence. In hospitals for example, policymakers are heavily promoting the adoption of electronic medical records; for the cost of

storing, collating, copying, and maintaing is dramatically larger in paper-based medical records [25]. The convenience of smartphones and other portable devices, along with the enhanced availability of internet services such as social networks, have made sharing information so effortless, which as a result flooded the World Wide Web with available public data. By all measurements, one cannot disclaim the existence of countless other examples of technologies that made storing digitized information as effortless and as convenient as possible. However, perhaps what could be considered a yet more important recent transformation, is the fact that organizations, especially in the field of Information Technology, only recently started to realize the power and importance of data. Google for example, stores all search queries that happen at a rate higher than three billion queries every day, and as a result, it was able to predict the the spread of the 2009 H1N1 winter flu in the United States, not only nationally, but down to specific regions and even states; such challenges have enormous advantages, and the information for this particular case was of vital importance, as this flu was so feared that some warned of an outbreak on the scale of the 1981 Spanish flu that infected half a billion people and killed millions. While big companies such as Google, IBM, and Microsoft are designating dedicated research units to be operated by data mining experts, other smaller companies are posting their data mining problems as prized challenges for public prediction competitions; Kaggle, founded in 2010, is a crowd-sourcing platforms that hosts data mining competitions for other companies, with prizes ranging from $3 million (Heritage Health Challenge) to $150 (R Package Recommendation Engine); the topics of data mining challenges are diverse, including challenges from fields such as astronomy, computer vision, transportation, healthcare,

and many others.

As a consequence of this digital revolution, and the sudden realization of the importance and great potentials of predicting, the amount of digital data has been increasing exponentially; in 2007, about as much as 93% of our data existed in digital format, while it is estimated for digital data to exceed 98% out of all data in 2013 [24]. This revolution of digital data is demanding a yet vaster body of work in the field of data mining, which remains to be one of the most important and essential fields in computer science and statistics. Scientific researchers around the globe are racing to take the most advantage of the vastly available data to extract useful and hidden patterns that will help make decisions, or better yet, find solutions to existing problems.

Our lives are overwhelmed with an ever-increasing amounts of data; the applications of data mining are countless, spanning a wide range of different domain areas. Next we discuss some of the common applications of data mining.

Web Content Mining is the subarea of data mining that is concerned in extracting knowledge from data existing in the World Wide Web. The technology witnessed in our society, today as we speak, is allowing us to digitally shift not only our structured explicit knowledge found in registered publications such as books and journal papers, but also our unstructured implicit thoughts, which include, but are not limited to, opinions of products such as music, movies, and books; opinions of places such as hotels, restaurants, and service shops; and opinions of events and states, whether economical, social, or political.

Previously, we introduced one example of Google predicting the spread of the H1N1

flu in 2008, by mining query data through the use of search engines in the web. Needless to say, the fact is, there exist a hundredfold other examples of using web mining to solve real-world problems.

Publicly shared data through social media on the web such as Twitter has been used for tracking illness over times (syndromic surveillance), measuring behavioral risk factors, localizing illness by geographical region, and analyzing symptoms and medication usage [29].

In a recent study, Bermingham and Smeaton investigate Twitter posts (also known as tweets) by sentiments, to measure political public opinion to predict election results [30]. In another study, Eisenstein et al. used tweets to study lexical variations across geographical regions to recover coherent topics and their regional variations, while identifying geographic areas of linguistic consistencies [31]. Other examples of the use of web mining span areas of entertainment, education, and quality of life (QOL). Arguably speaking, based on the recent spur of its numerous applications, web mining could be considered the most appealing area of applications amongst all others. However, web mining still suffers from its disorganized nature, which is ironically, in some certain sense, the reason why it gained popularity between researchers; with unstructured quality and disorganized nature, new challenges emerge, hence new research opportunities! Next, we introduce two other equally common, but profoundly more organized, applications of data mining.

Market Basket Analysis is another application of data mining that had its own worthy history in the literature. Its clear problem definition, basic formulation, and direct applicability (and gain), made it one of the earliest data mining applications to be

studied by researchers. Just like most other areas in the research literature, the utility and advantages of a particular application is the driving force for the complementary work of research, and although the usefulness may only seem minor at first, as the research contributions elevate to higher degrees of sophistication, more functionalities become apparent; the application of market basket analysis is no exception.

The most elementary utility of market basket analysis is to find correlations of products purchases; records of items purchased together are hence used to find patterns of products bought in groups, as for each transaction undertaken by a client, the list of items purchased through that transaction are digitally recorded; the need to extract correlation patterns such as the following hypothetical one: 80% of people who buy bread also buy butter, have major implications of the decision making process, for example, one could redesign the structure of the store inventory, by positioning the two items, namely bread and butter, far from each other, assuming that customers will be more inclined to purchase other items while walking from one end to the other. More advanced utilities of market basket analysis exist nonetheless, one rather more involved data mining study would involve the time in which products were purchased; this of course, does not need to be confined to the examination of time with respect to the normal day, but also may include the study purchase time with respect to days of the week, or the period of the year in which particular items were bought; this utility will help decision makers anticipate timely customer needs, hence make decision with regard to store inventory management.

Lastly, yet another more advanced function would be to investigate, through exercising existing techniques of data mining, patterns of customers' purchases over time.

In most relatively medium to large retail stores, discounts on products are provided through what is known in the retail store terminology as 'loyalty card', which could be alternatively described as a fancy way for retail stores to keep track of what particular customers buy over time; this will allow store management to make not only general decisions about the overall purchases, but also highly valuable 'personalized' decisions on whom (and at what time) to offer coupons and deals to maximize customer loyalty.

Healthcare is another enormously essential application of data mining; its main importance lies at the heart of its utility. In addition to the extensive popularity and great support it has gained in recent years, diagnosis applications of data mining are with no doubt the closest type of applications under which the work presented in this thesis could be classified. Applications of data mining in the field of healthcare are many however, few examples would be the use of data mining to help detect insurers fraud and abuse, improve customer relationship management, and identify effective treatments and best practices. In this thesis, we present a complete case study, in which we detect proper treatments for hyper-nasal speech disorder.

Methodologies in data mining are plenty; however, the details of the particular problem to be solved, and the nature in which our dataset exist in, pose prime restrictions on our selection criteria, those restrictions meanwhile, serve as a filtering phase in which data mining experts use to get clues on the best knowledge discovery approach to be chosen. The level of interpretability, which could be defined as the degree of insight learned from the resulting mined patterns, as will be shown in future section, is of significant importance in the field of healthcare applications; hence, we limit our discussion in this thesis to the subcategory of data mining known as rule-

based knowledge discovery. In section 1.3, we provide a thorough exploration of the concept of rule-based knowledge discovery, and it should be apparent only then, how vital to our study is to have patterns of high degree of interpretability.

1.2 Knowledge Representation

Knowledge can be represented in many ways. In this section, and for computational reasons, we discuss and examine the tabular representation form of knowledge, which we will continue to use for the rest of the thesis. Tabular representation of knowledge can be viewed as a special kind of "formal language", used to represent equivalence relations (or partitions) in symbolic form suitable for computer processing [17]. Information systems and attribute-value systems are two terms that will also be used interchangeably with the tabular knowledge representation.

The knowledge representation can be intuitively perceived as a data table, in which columns are labelled by *attributes*, and in which rows are labeled by *values* (or *states*) of their instances. Each row will represent an independent observation about the corresponding instance. Next, we formally define our information system.

By information system [14], we mean a sequence $S = (X, A, V)$, where:

1. X is a nonempty, finite set of instances,

2. A is a nonempty, finite set of attributes;

   $a : X \to V_a$ is a function for any $a \in A$, where V_a is called the domain of a,

3. $V = \bigcup\{V_a : a \in A\}$.

Despite the fact that elements of X are sometimes referred to as *objects*, in this

Table 1: Information system S_1

	e	f	g	h
x_0	e_1	f_1	g_1	h_1
x_1	e_2	f_1	g_1	h_1
x_2	e_2	f_2	g_2	h_2
x_3	e_1	f_2	g_1	h_1
x_4	e_2	f_1	g_1	h_2
x_5	e_1	f_2	g_1	h_2
x_6	e_2	f_1	g_1	h_1

note we will not use the two terms interchangeably, objects will possibly consist of, as will be discussed in future sections, multiple instances. The distinction between instances and objects is necessary for the understanding of the work presented in this thesis.

For example, Table 1 shows an information system S_1 with a set of instances $X = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6\}$, set of attributes $A = \{e, f, g, h\}$, and a set of their values $V = \{e_1, e_2, f_1, f_2, g_1, g_2, h_1, h_2\}$. Each row in Table 1 shows one complete observation about its corresponding instance; the first row for example, shows values for instance x_0; its state for attribute e is e_1, which would also be denoted by the expression (e, e_1); attribute f has state f_1, or (f, f_1); attribute g has state g_1, or (g, g_1); and the state of attribute h is h_1, or (h, h_1).

Note that it is not necessarily for our data system to exist in a tabular format when first presented; as will be seen in the following section, it is often the case that we would need to transform our data system into the tabular format, after in which we apply knowledge discovery techniques. Needless to mention, it is also important to keep in mind that when we deal with a typical information system, we make the assumption that the instances are i.i.d.; independent and identically distributed,

unless mentioned otherwise; this means that the order of rows does not have any significance, it also means that each complete observation (or row) is identical in significance as any other row, and that the only difference(s) between observations is in their attribute states.

1.3 Rule-based Knowledge Discovery and Association Rules

In this section, we examine the concept of association rules, originally proposed in [28]. Association rule discovery is a highly researched sub-area in the field of data mining. In addition to its vital importance, the notion of association rules is closely related to action rules, hence related to the work presented in this thesis.

Association rule discovery uncovers hidden relations between attributes in information systems; the structure of relations is described by a set of if/then statements, where the *if* side is referred to as the *antecedent*, and the *then* side is referred to as the *consequent*.

One of the earliest applications for association rules is market basket analysis (or shopping analysis) for customers' purchases; for the (initial) purpose of analyzing and predicting customer behavior, and for the (eventual) purpose of making profit-generating decisions accordingly. To illustrate association rules in practice, we present a hypothetical example of market analysis. Table 2 shows a sample transaction of a hypothetical market basket.

As mentioned in previous section, the first step would be to represent our information system in a tabular format. For this particular case, the most appropriate approach would be to transform our market basket table into a truth table. Table 3

Table 2: Market basket table

ItemID	*Items*
1	{Bread, Milk}
2	{Bread, Jam, Butter, Eggs}
3	{Milk, Jam, Butter, Cola}
4	{Bread, Milk, Jam, Butter}
5	{Bread, Milk, Jam, Cola}

is the result of this transformation.

Table 3: Truth table extracted from Table 2

ItemID	*Bread*	*Milk*	*Jam*	*Butter*	*Eggs*	*Cola*
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

The basic function of market basket analysis is to find associations (or correlations) between products bought by customers. For example, only by looking at Table 2 (or Table 3), we would be able to observe that *Bread* and *Milk* are two items that seem to be bought together; clearly, from such a small dataset, this observation would not mean much, but only for the sake of explanation, these small datasets will be considered. So, by observing the two items, namely *Bread* and *Milk*, we could extract a rough rule that states the following: customers who buy *Milk* are also inclined to buy *Bread* (or $\{Milk\} \rightarrow \{Bread\}$) or vice-versa; however, next, we will show that although these two rough inverse rules might seem identical in likelihood at first, in reality they're not, and one is in fact stronger (or more likely to occur) than the other. To that end, we introduce the concepts of *support* and *confidence* for association rules.

First, let us introduce the concept of an an *itemset*. An itemset could be defined

as a particular configuration (or sub-configuration) of an observation. For example, referring to our market basket example, a candidate 2-element itemset would be {Bread, Butter}. Itemsets have two properties that need to be defined, the first one is the support, and the second one is the confidence. The support of an itemset is the number of observations that satisfy that (sub)-configuration. So, using the same 2-element itemset {Bread, Butter}, we can observe that the support is equal to 2, since there are only two instances that satisfy that sub-configuration, namely the second and fourth. Two points are worth mentioning here, the first one is that when itemsets are only stating a sub-configuration, the other attributes unmentioned could have any value. For example, the sub-configuration {Bread, Butter} does not mean that observations need to only contain Bread and Butter, which would on the hand be expressed as {Bread, ∼ Milk, ∼Jam, Butter, ∼Eggs, ∼Cola}; this confusion is only apparent in logical/truth tables. The second point that is worth mentioning is the way regular tables represent itemsets, since attributes normally have more than two states, mentioning the actual value of their states is necessary; for example, if we refer to Table 1 shown on page 8, an itemset need to be of the following format: {([attribute label 1], [state 1],), ([attribute label 2], [state 2], ...)}, so a valid example of a 3-element itemset in Table 1 would be the following: $\{(e, e_2),\ (g, g_1),\ (h, h_2)\}$. Using the definition of the information system presented in section 1.2 (page 7), we can mathematically define the support $\sigma(P)$ of an itemset P as the following:

$$\sigma(P) = |\ \{x_i \mid P \subseteq x_i,\ x_i \in X\}\ |$$

Which is essentially the cardinality of the subset in X that satisfies that config-

uration specified by P; hence, it can be easily verified that the support of {Bread, Jam} and {Bread, Cola} are 3 and 1, respectively. It can also be as easily verified that the support of $\{(e, e_2),\ (g, g_1),\ (h, h_2)\}$ and $\{(f, f_1),\ (g, g_2)\}$ in Table 1 are 1 and 0 respectively. The methodology to extract itemsets with high support values is vital in the extraction of association rules; however, we will not provide any details on how to achieve that here, as we believe it is outside the scope of this thesis, and that it would only be helpful to present the reader with a general understanding of the concept of support. Next, we introduce the notion of the confidence; however, we introduce the concept of association rules prior to that; as the confidence is a term attached to an association rule and not an itemset.

Association rules are implication expressions that are denoted by a rule-based structure (or if/then). The format of association rules is as follows: $X \to Y$, where both X and Y are itemsets; the expression hence means that if X is satisfied, the Y is also satisfied; for example, the simple association rule {Bread} $\to$ {Milk} would mean that if a customer buys {Bread} then he will buy {Milk}; note that the attributes of the antecedent (left side of right arrow) and the consequent (right side of right arrow) need to be mutually exclusive. Each association rule has both the support and the confidence. The support for an association rule is the support of the disjoint of the antecedent side and the consequent side, divided by the number of observations in our dataset:

$$\text{Support}(X \to Y) = \frac{\sigma(X \cup Y)}{N},$$

where N is the number of observations in our information system. The support of a particular association rule is a good indicator of how often it is seen in our information system, in other words, how common is that association rule.

The confidence of an association rule on the other hand, is an indicator of how accurate that association rule is; and it is denoted by the support of the disjoint of the antecedent side and the consequent side, divided by the support of only the antecedent side:

$$\text{Confidence}(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)},$$

For example, let us use Table 3 to calculate the support and confidence for the two inverse association rules {Milk} → {Bread} and {Bread} → {Milk}. Starting with the first association rule; {Milk} → {Bread}:

$$\text{Support}(\{Milk\} \to \{Bread\}) = \frac{\sigma(\{Milk, Bread\})}{5} = \frac{3}{5} = .6,$$
$$\text{Confidence}(\{Milk\} \to \{Bread\}) = \frac{\sigma(\{Milk, Bread\}}{\sigma(\{Milk\})} = \frac{3}{3} = 1.$$

Now let us examine the second association rule; {Bread} → {Milk}:

$$\text{Support}(\{Bread\} \to \{Milk\}) = \frac{\sigma(\{Bread, Milk\})}{5} = \frac{3}{5} = .6,$$
$$\text{Confidence}(\{Bread\} \to \{Milk\}) = \frac{\sigma(\{Bread, Milk\})}{\sigma(\{Break\})} = \frac{3}{4} = .75.$$

Note that although the two association rules have identical support, their confidence

is different. Next chapter we will build on the readers' understanding of support and confidence provided here, and will therefore expand our definitions for the two terms, to cover the literature of action rules.

CHAPTER 2: ACTION RULES

2.1 Motivation: From Concepts to Applications

As we have discussed in Chapter 1, the general goal of any data mining system is roughly the same, that is, to extract useful patterns that describe nontrivial, useful, and valid knowledge; usually in the form of relations between system attributes. In Section 1.3, we introduced the arrangements of patterns that are commonly extracted through rule-based methods and association rules; hence, providing a good understanding of the types of queries they address, and the nature of questions they answer. In this section however, we will introduce an entirely different class of data mining techniques, a category that is considered, by many researchers, more applicable and more useful in today's vast amount of data available; in this section we introduce action rules.

To better understand the strengths of action rules, it would help to bring forward the previously presented concept of association rules, to provide few key comparisons and therefore clarify the exclusive advantages of action rules. As discussed earlier, the outcome of association rule learning is to discover interesting relations amongst attributes in large sets of data. For example, a candidate outcome of association rules learning would be the following rule: {High Fever, Severe Fatigue} $\rightarrow$ {Flu}, which states that if a patient is experiencing high fever and severe body fatigue, then he or

she is more likely to have caught the flu. Though this might seem at first to be the sole and ultimate goal of any data mining system, it would soon appear, with little additional inspection, that the passive nature of an association rule is rather lacking, especially in a world where we're most in need for suggestions and recommendations, rather than mere analysis. It is often the case that system users are also, if not more, interesting in ways of transitioning (or shifting) an attribute condition from an undesired state to a desired state; by referring to our patient example, system users would be potentially interested in patterns that would shift the patient's condition from having the flu, to not having the flu. In other words, the idea of action rules is not only to provide hidden patterns in the data, but also to suggest viable changes that, if applied according to the action rule, will result in a desired change in our decision attribute from a less desirable state to a more desirable one. Hence, the motivation of action rules is to bring forth a new category of techniques and tools to provide system users with concise actionable patterns that have high overall interestingness level.

Since its introduction in 2000, action rules have been successfully applied in many domain areas including business [1], medical diagnosis and treatment [5], [6], [7] and music automatic indexing and retrieval [12], [13].

2.2 Definition and Interpretation

The notion of action rules was first proposed by Z. W. Raś and A. Wieczorkowska in [1]. Action rules describe possible transition of objects from one state to another with respect to a specific attribute, called the decision attribute. The goal of action

rules is to provide system users with actionable tasks that can be directly applied to objects listed in information systems to reach a desired goal. Table 4 shows an example of an information system S_2.

Table 4: Information system S_2

	e	f	g	d
x_0	e_1	f_1	g_1	d_1
x_1	e_1	f_1	g_1	d_1
x_2	e_2	f_2	g_2	d_2
x_3	e_1	f_2	g_1	d_1
x_4	e_2	f_1	g_1	d_2
x_5	e_1	f_2	g_1	d_2
x_6	e_2	f_2	g_1	d_1

In this section, we will use the definition of information system introduced in Section 1.3. However, we will be expanding few definitions starting with the definition of a decision system (or decision table). By a decision system (table), we mean an information system that makes a clear explicit distinction between attributes in A, and will therefore label each attribute as either a *decision attibute*, or a non-decision attribute, called *condition attribute*. The decision attribute(s), which normally but not necessarily is a single attribute, is the attribute that we are interested in the most. For system users, the eventual goal would be to change the decision attribute from a less desirable to a more desirable state. For example, a company would be interested in moving clients' states of loyalty from lower to higher.

All non-decision, or condition, attributes are further partitioned into two mutually exclusive sets; the first one is the *stable* attributes set, and the second one is the *flexible* attributes set. By stable attributes set we mean the set that contains attributes that we have no control over; their values cannot be changed by the users

of our system. An example of a *stable* attribute is the place where the person was born. On the other hand, values of *flexible* attributes can be influenced and changed; an example of a *flexible* attribute is the patient's prescribed medications. In this paper, A_{St}, A_{Fl}, and $\{d\}$ will represent the set of stable attributes, the set of flexible attributes, and the decision attribute, respectively. Hence, the set of attributes A can be redefined as $A = A_{St} \cup A_{Fl} \cup \{d\}$.

An *atomic action set* is an expression that defines a change of state for a single distinct attribute. For example, $(a, a_1 \rightarrow a_2)$ is an atomic action set which defines a change of state for the attribute a from a_1 to a_2, where $a_1, a_2 \in V_a$. Clearly in this case, the attribute a is a flexible attribute, since it changes its state from a_1 to a_2. In the case when there is no change, we omit the right arrow sign, so for example, (b, b_1) means that the value of attribute b is b_1 and remains b_1, where $b_1 \in V_b$.

The *action set* is defined as follows:

1. If t is an atomic action set, then t is an action set.

2. If t_1, t_2 are action sets and $\wedge$ is a 2-argument functor called composition, then $t_1 \wedge t_2$ is a candidate action set.

3. If t is a candidate action set and for any two atomic action sets $(a, a_1 \rightarrow a_2), (b, b_1 \rightarrow b_2)$ contained in t we have $a \neq b$, then t is an action set.

4. No other sets are called action sets.

The *domain* $Dom(t)$ of an action set t is the set of attributes of all atomic action sets contained in t. For example, $t = (a, a_1 \rightarrow a_2) \wedge (b, b_1)$ is an action set that consists

of two atomic action sets, namely $(a, a_1 \rightarrow a_2)$ and (b, b_1). Therefore, the domain of t is $\{a, b\}$.

Action rules are expressions that take the following form: $r = [t_1 \Rightarrow t_2]$, where t_1, t_2 are action sets. The interpretation of the action rule r is that by applying the action set t_1, we would get, as a result, the changes of states in action set t_2. We also assume that $Dom(t_1) \cup Dom(t_2) \subseteq A$, and $Dom(t_1) \cap Dom(t_2) = \phi$.

For example, $r = [[(a, a_1 \rightarrow a_2) \wedge (b, b_2)] \Rightarrow (d, d_1 \rightarrow d_2)]$ means that by changing the state of the attribute a from a_1 to a_2, and by keeping the state of the attribute b as b_2, we would observe a change in the attribute d from the state d_1 to d_2, where d is commonly referred to as the *decision attribute*.

Standard interpretation N_s of action sets in S is defined as follows:

1. If $(a, a_1 \rightarrow a_2)$ is an atomic action set, then

   $N_s((a, a_1 \rightarrow a_2)) = [\{x \in X : a(x) = a_1\}, \{x \in X : a(x) = a_2\}]$.

2. If $t_1 = (a, a_1 \rightarrow a_2) \wedge t$ and $N_s(t) = [Y_1, Y_2]$, then

   $N_s(t_1) = [Y_1 \cap \{x \in X : a(x) = a_1\}, Y_2 \cap \{x \in X : a(x) = a_2\}]$.

Let us define $[Y_1, Y_2] \cap [Z_1, Z_2]$ as $[Y_1 \cap Z_1, Y_2 \cap Z_2]$ and assume that $N_s(t_1) = [Y_1, Y_2]$ and $N_s(t_2) = [Z_1, Z_2]$. Then, $N_s(t_1 \wedge t_2) = N_s(t_1) \cap N_s(t_2)$.

If t is an action set and $N_s(t) = [Y_1, Y_2]$, then the support of t in S is defined as $supp(t) = min\{card(Y_1), card(Y_2)\}$.

Let $r = [t_1 \Rightarrow t_2]$ be an action rule, $supp(t_1) > 0, N_s(t_1) = [Y_1, Y_2]$, and $N_s(t_2) =$

$[Z_1, Z_2]$. Support $supp(r)$ and confidence $conf(r)$ of r are defined as:

$$supp(r) = min\{card(Y_1 \cap Z_1), card(Y_2 \cap Z_2)\},$$

$$conf(r) = \left[\frac{card(Y_1 \cap Z_1)}{card(Y_1)}\right] * \left[\frac{card(Y_2 \cap Z_2)}{card(Y_2)}\right].$$

For example, referring to Table 4, let us assume that our decision attribute is d, and let us also assume that we are interested in shifting our decision state from d_1 to d_2; a candidate action rule would be the following: $r_1 = (e, e_1 \rightarrow e_2) \wedge (f, f_1 \rightarrow f_2) \Rightarrow (d, d_1 \rightarrow d_2)$, which means that by shifting the state of attribute e from e_1 to e_2, and the state of attribute f from f_1 to f_2, we should observe a desired shift in our decision attribute d from d_1 to d_2. Using our previous definitions, we calculate the support and confidence of this action rule r_1; we start by calculating the standard interpretation for both the condition and decision side of r_1; Y_1, Y_2, Z_1, and Z_2:

$$N_s((e, e_1 \rightarrow e_2) \wedge (f, f_1 \rightarrow f_2)) = [Y_1, Y_2] = [\{x_0, x_1\}, \{x_2, x_6\}],$$

$$N_s(d, d_1 \rightarrow d_2) = [Z_1, Z_2] = [\{x_0, x_1, x_3, x_6\}, \{x_2, x_4, x_5\}],$$

$$supp(r_1) = min\{card(\{x_0, x_1\}), card(\{x_2\})\} = 1,$$

$$conf(r_1) = \left[\frac{card(\{x_0, x_1\})}{card(\{x_0, x_1\})}\right] * \left[\frac{card(\{x_2\})}{card(\{x_2, x_6\})}\right] = 1 * \frac{1}{2} = .5.$$

2.3 Extraction of Action Rules

There has been considerable research on the varied methodologies for extracting action rules from information systems [8], [9], [10], [11]. In general however, we can categorize all methodologies into two groups; the first one being when classification rules are required for the construction of action rules [20], [1], [21], [22], and the

second, more recent approach, being when action rules are directly extracted from an information system [23]. To extract action rules, we used the algorithm described in [23]. The idea of the algorithm is to start by constructing all possible action sets that have occurred more than a pre-defined number, called the minimum support. Then, in accordance to our desired change in the decision attribute, action rules are formed.

Let t_a be an action set, where $N_s(t_a) = [Y_1, Y_2]$ and $a \in A$. We say that t_a is a *frequent action set* [23] if $card(Y_1) \geq \lambda_1$ and $card(Y_2) \geq \lambda_1$, where λ_1 is the minimum support. Another way of interpreting the frequent action sets would be that all frequent action sets have support greater than or equal to the minimum support λ_1. By specifying λ_1, we make sure that the extracted action rules have support greater than or equal to the minimum support λ_1. The algorithm presented below is similar to [28]. To extract action rules, we start by generating atomic action sets that have support greater than or equal to the minimum support value λ_1 pre-defined by the user; we will refer to this set as *1-element frequent action set*. The term *frequent* will be used to indicate that an action set has support greater than or equal to the minimum support, and the term *k-element* will be used to indicate the number of elements (or atomic action terms) in an action set. Both *frequent atomic action sets* and *1-element frequent action set* refer to exactly the same set, since from the definition of atomic action sets, they consist of only one element. After generating all frequent atomic action sets, we undertake the following two-step process initially for $k = 1$:

1. *Merge step:* Merge pairs (t_1, t_2) of k-element action sets into all $(k+1)$-element

candidate action sets.

2. *Delete step:* Delete all $(k+1)$-element candidate action sets that are either not action sets, or contain a non-frequent k-element action set, or that have support less than the minimum support λ_1.

We keep iterating the above two steps until we cannot generate new frequent action sets anymore. At this point, we have generated all $(k+1)$-element frequent action sets, which will allow us to generate action rules that are guaranteed to have support greater than or equal to the minimum support λ_1. Last step is to further filter the desired action rules based on their confidence, where we only consider action rules with confidence greater than or equal to a pre-defined minimum confidence λ_2. For example, from the frequent action set $t_1 = (a, a_1 \rightarrow a_2) \wedge (d, d_1 \rightarrow d_2)$, we can generate the following two action rules:

1. $r_1 = [(a, a_1 \rightarrow a_2) \Rightarrow (d, d_1 \rightarrow d_2)]$.

2. $r_2 = [(d, d_1 \rightarrow d_2) \Rightarrow (a, a_1 \rightarrow a_2)]$.

where both r_1 and r_2 have support greater than or equal to the minimum support λ_1. However, we will only be interested in specific changes of the decision attribute, e.g. in changing the decision attribute d from state d_1 to d_2. Therefore, we will only consider r_1.

Action rule extraction example: using the previously described approach, we will extract action rules from information system $S_3 = (\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}, \{b, c\} \cup \{a\} \cup \{d\}, V), V = V_b \cup V_c \cup V_a \cup V_d$, shown below. The set {b, c} lists stable attributes,

a is a flexible attribute, and d is a decision attribute. Also, we assume that we are interested in the transition of decision attribute from state L to H, referring to *low profit*, and *high profit*, respectively; we assume λ_1 (support) = 3.

Table 5: Information system S_3

	a	b	c	d
x_1	0	S	0	L
x_2	0	R	1	L
x_3	0	S	1	L
x_4	0	R	1	L
x_5	2	P	2	H
x_6	2	P	2	H
x_7	2	S	2	H
x_8	2	S	2	H

We then start generating *1-element frequent action sets*:

$(a, 0)$*; support 4 (frequent)*

$(a, 0 \rightarrow 2)$*; support 4 (frequent)*

$(a, 2)$*; support 4 (frequent)*

$(a, 2 \rightarrow 0)$*; support 4 (frequent)*

(b, S)*; support 4 (frequent)*

(b, R); support 2 (not frequent)

(b, P); support 2 (not frequent)

$(c, 0)$; support 1 (not frequent)

$(c, 1)$; support 3 (not frequent)

$(c, 2)$*; support 4 (frequent)*

(d, L)*; support 4 (frequent)*

$(d, L \rightarrow H)$*; support 4 (frequent)*

(d, H)*; support 4 (frequent)*

$(d, H \rightarrow L)$*; support 4 (frequent)*

Next, we generate (k+1)-element candidate action sets (where k = 1):

$(a, 0) \wedge (b, S)$; support 2 (not frequent)

$(a, 0) \wedge (c, 2)$; support 0 (not frequent)

$(a, 0) \wedge (d, L)$*; support 4 (frequent)*

$(a, 0) \wedge (d, H)$; support 0 (not frequent)

$(a, 0) \wedge (d, L \rightarrow H)$; support 0 (not frequent)

$(a, 0) \wedge (d, H \rightarrow L)$; support 0 (not frequent)

$(a, 0 \rightarrow 2) \wedge (b, S)$; support 2 (not frequent)

$(a, 0 \rightarrow 2) \wedge (c, 2)$; support 0 (not frequent)

$(a, 0 \rightarrow 2) \wedge (d, L)$; support 0 (not frequent)

$(a, 0 \rightarrow 2) \wedge (d, H)$; support 0 (not frequent)

$(a, 0 \rightarrow 2) \wedge (d, L \rightarrow H)$*; support 4 (frequent)*

$(a, 0 \rightarrow 2) \wedge (d, H \rightarrow L)$; support 0 (not frequent)

$(a, 2) \wedge (b, S)$; support 2 (not frequent)

$(a, 2) \wedge (c, 2)$*; support 4 (frequent)*

$(a, 2) \wedge (d, L)$; support 0 (not frequent)

$(a, 2) \wedge (d, H)$*; support 4 (frequent)*

$(a, 2) \wedge (d, L \rightarrow H)$; support 0 (not frequent)

$(a, 2) \wedge (d, H \rightarrow L)$; support 0 (not frequent)

$(b, S) \wedge (c, 2)$; support 2 (not frequent)

$(b, S) \wedge (d, L)$; support 2 (not frequent)

$(b, S) \wedge (d, H)$; support 2 (not frequent)

$(b, S) \wedge (d, L \rightarrow H)$; support 2 (not frequent)

$(b, S) \wedge (d, H \rightarrow L)$; support 2 (not frequent)

$(c, 2) \wedge (d, L)$; support 0 (not frequent)

$(c, 2) \wedge (d, H)$*; support 4 (frequent)*

$(c, 2) \wedge (d, L \rightarrow H)$; support 0 (not frequent)

$(c, 2) \wedge (d, H \rightarrow L)$; support 0 (not frequent)

We continue to generate the next (k+1)-element candidate action action set (where k = 2):

$(a, 2) \wedge (c, 2) \wedge (d, H)$*; support 4 (frequent)*

No more frequent action sets can be further generated. The only frequent action set that we are interested in is $(a, 0 \rightarrow 2) \wedge (d, L \rightarrow H)$*; support 4 (frequent)*, since we desire decision state transition from L (Low profit) to H (High profit). To construct action rules, we move the decision attribute atomic action set to the right side, which in this case will give us the following action rule: $(a, 0 \rightarrow 2) \Rightarrow (d, L \rightarrow H)$. Lastly, we check the confidence of this action rule to make sure it is greater than or equal to the minimum confidence λ_2.

CHAPTER 3: OUR DATASET: HYPERNASALITY TREATMENT

3.1 Overview of Hypernasal Speech Disorder

Distortions of the velopharyngeal closure, resulting in speech hypernasality or hyponasality, may cause speech disorders in children [26]. The patient's nasopharynx disorders have been examined in the Children's Memorial Health Institute in Warsaw for many years. The gathered data also include general information on the patient's condition if it can be of importance, e.g. cerebral palsy, neurology, or myopathy. This way a rich collection of complex data describing hypernasality was gathered, in close cooperation with one of our collaborators, Prof. Ryszard Gubrynowicz, who is a speech scientist and expert in this area; the data were collected when he was working in the Children's Memorial Health Institute.

Hypernasality can be examined by means of Czermak's mirror test of nasal air escape, see Figure 1. The child is asked to repeat several times a syllable composed of a plosive consonant and an open vowel, e.g. /pa/-/pa/-/pa/, and the sizes of the fogging circles appearing on the mirror are rated on 4–point scale, from 0 (no hypernasality) to 3 (most severe hypernasality). Therefore, *Czermak's mirror test* was used as a decision attribute in the nasality data set. All attributes, representing various medical conditions in the examined children, are listed in Table 3.2. More explanations about these attributes are given below.

Figure 1: Czermak's mirror fogging test, rating the degree of the patient's nasal air escape on a 4-point scale: none = 0; small = 1, medium = 2, large = 3 [26].

3.2 Description of Original Attributes

Our dataset is composed of 225 patients; each patient was examined several times where each examination was performed on a separate visit, ranging from 2 to 11 visits for each patient. Personal data were recorded (first name and last name, sex), and for each examination the age of the child was marked. Personal data were removed before further processing, and replaced with ID data, representing the patient's ID combined with the sequential number of this patient's visit.

During each visit, the articulation of selected vowels and consonants was recorded, and the recording date was marked (*recording date* attribute). The data stored in columns marked as *diagnosis* and *diagnosis*2 describe patient's condition related to nasality; only one diagnosis is stored in each of these columns, so *diagnosis*2 represents additional diagnosis, if there is more than one. The following diagnoses are described in these columns: R - cleft, RP - cleft palate, OR - after cleft palate surgery, WKP - congenital short velum, NO - hypernasality, NZ - hyponasality, BR - no diagnosis, PRP - submucous cleft palate, AT - after tonsillectomy, DKP - quite short palate, RJ - cleft uvula, III - hypertrophy of adenoids and possibly palatine tonsils, MP - hypertrophy of palatine tonsils, MPDz - cerebral palsy, AD - after

adenotomy, ADT - after adenotonsillectomy, UK - larynx after injury/trauma, NS - hypoacusis, ORM - retarded speech development, NEU - neurology, ONR - after neurological surgery. If NO (hypernasality) is diagnosed and marked in the column *diagnosis*, it represents the most severe case of hypernasality. The numbers 0–3 in *diagnosis*2 refer to sleep apnoea, i.e. temporary cessation of respiration during sleep. 0 means no apnoea, 3 - very often. Sleep apnoea is also represented as a separate attribute, but the values assessed for the same patient may differ significantly, so they were kept in both columns. Generally, physicians may differ in their opinions, this is why we must be prepared to deal with some inconsistencies in the data. More of diagnostic details are given in the column *comments*, but these comments are not taken into account in the current version of our action rule software.

Other physical conditions recorded in the database include the degree of hypertrophy of adenoids and possibly palatine tonsils, and the degree of motility of the soft palate, represented as *tonsils* and *motility* attributes. The assessment of the patient's recorded speech is represented in the following attributes: *yeaoui* (vowels /I, e, a, o, u, i/ - a sequence of short vowel sounds spoken in isolation), $i - long$ (long vowel /i/ - vowel of sustained phonation), and *bdg* (high pressure consonants /b, d, g/); SAMPA coding of phonetic alphabet is used [27]. These attributes describe the measure of nasalization (coefficient of nasalization), calculated from the analysis of mouth and nose signals (separately recorded), as the ratio of the nose signal level to the sum of the level of the nose and mouth signals for the phonemes indicated in each attribute. *difference level* $F1 - F2$ describes the vocal tract's first 2 resonances as the difference level of the 1st and the 2nd formant, measured for /i/-long.

Table 6: Attributes in the hypernasality dataset. Expansions of acronyms are given in the text.

Attribute	Description
ID	Patient's ID, with the sequential number of his/her visit
age	Age [years, months]
sex	Sex {M, F}
recording date	Recording Date [yyyy.mm.dd]
diagnosis	Diagnosis {AD, ADT, AT, BR, III, myopathy, MPDz, NEU, NO, ONR, OR, ORM, RJ, RP, UK, WKP}
comments	Comments, details of the diagnosis
*diagnosis*2	Diagnosis {0, 1, 2, 3, DKP, RJ, WKP}
sleep apnoea	Sleep apnoea {0, 1, 2, 3}
tonsils	Hypertrophy of adenoids and possibly palatine tonsils {0, 1, 2, 3}
$Czermak's\ mirror\ test$ - decision attribute	Mirror-fogging test {0, 1, 2, 3}
yeaoui	Measure of nasalization for vowels /I, e, a, o, u, i/ [0, 100]
$i - long$	Measure of nasalization for vowel /i/-long [0, 100]
bdg	Measure of nasalization for high pressure consonants /b, d, g/ [0, 100]
motility	Motility of the soft palate [0, 12]
difference level $F1 - F2$	The difference level of 1st & 2nd formant measured for /i/-long [-14, 26]

The best diagnosis we are interested in is when the parameters' values are in normal ranges. Our decision attribute is Czermak's mirror test, so its values are most important in our research. The most desired value of our decision attribute is when it is equal to 0. The diagnosis is worse when Czermak's test value equals 2, next worse case is when Czermak's test value equals 3, and this is the most severe case. The lower the Czermak's test value, the better the diagnosis is. Therefore, we are interested in action rules indicating how to decrease the Czermak's test value. The goal of our system is to find action rules which purpose is to provide hints referring to doctor's interventions. They show how values of certain attributes need to be changed (through various medical procedures, according to the physician's order), so the patient's condition will get improved.

3.3 Description of the Derived Attributes

In this work, we derived a new set of attributes in accordance to [3]. In addition to our attributes shown in Table 2, for each of the following four attributes: *yeaoui, i - long, bdg*, and *motility*, two new attributes were derived, resulting in eight new attributes. The two derived attributes are the difference, and the rate of change for every two consecutive instances, which we calculated as follows:

1. The difference of values for *yeaoui, i - long, bdg* and *motility* for every two consecutive visits is calculated, thus constituting the following new attributes: $yeaoui_1, i_1 - long, bdg_1$ and $motility_1$. For example, the value of bdg_1 equals to the value of bdg for the $(k+1)^{th}$ visit minus the value for the kth visit.

2. The rate of change a_2 for every two consecutive visits is defined as:

$$a_2 = arctan\left(\frac{a_1}{\text{age difference in months}}\right)$$

where a_1 is the difference of values of the attribute a for the two visits.

After calculating the derived attributes, we used the Rough Set Exploration System [15] to discretize our real-valued attributes wrt. our decision attribute. Next, our temporal object-driven action rule discovery system, presented in Section 2.3, was applied to the discretized data.

Our decision attribute *Czermak's mirror test* was not discretized. Moreover, when a physician could not decide between two neighboring Czermak's test values, an intermediate value was assigned. Therefore, the decision values are {0, .5, 1, 1.5, 2, 2.5, 3}.

Snapshot of our dataset after cleaning (and discretization) is given below:

A	B	C	D	E	F	G	H	I	J	K	L
diagnosis	sleep apnoea	palatine tonsils	yeaou	i	i_2	bdg	bdg_1	motility	difference level F1-F2	Czermak test	
OR	less than 2	less than 2	[2.5, 3.5)	[2.5 7.5)	>= 5.5	>= 8.5	less than 6.5	[3.5, 4.5)	less than 4.5	1	
OR	less than 2	less than 2	[4.5, 5.5)	[7.5 9.5)	>= 5.5	>= 8.5	less than 6.5	[4.5, 5.5)	less than 4.5	2	
OR	less than 2	less than 2	>= 7.5	[7.5 9.5)	>= 5.5	>= 8.5	less than 6.5	[4.5, 5.5)	less than 4.5	2	
OR	less than 2	less than 2	[6.5, 7.5)	>= 9.5	less than 5.5	>= 8.5	less than 6.5	less than 3.5	less than 4.5	2.5	
OR	less than 2	less than 2	[6.5, 7.5)	>= 9.5	>= 5.5	>= 8.5	less than 6.5	less than 3.5	[6.5, 9.5)	2.5	
OR	less than 2	less than 2	[5.5, 6.5)	>= 9.5	less than 5.5	>= 8.5	less than 6.5	less than 3.5	[6.5, 9.5)	2.5	
OR	less than 2	less than 2	[5.5, 6.5)	[7.5 9.5)	less than 5.5	less than 6.5	less than 6.5	less than 3.5	less than 4.5	2.5	
NO	less than 2	less than 2	[5.5, 6.5)	>= 9.5	less than 5.5	>= 8.5	less than 6.5	less than 3.5	less than 4.5	1	
NO	less than 2	less than 2	[6.5, 7.5)	>= 9.5	>= 5.5	>= 8.5	less than 6.5	[4.5, 5.5)	less than 4.5	2	
MPDz	less than 2	less than 2	[5.5, 6.5)	>= 9.5	>= 5.5	>= 8.5	less than 6.5	[4.5, 5.5)	less than 4.5	2	
MPDz	less than 2	less than 2	[6.5, 7.5)	>= 9.5	less than 5.5	>= 8.5	less than 6.5	[4.5, 5.5)	[6.5, 9.5)	1.5	
OR	less than 2	less than 2	less than 2.5	less than 1.5	less than 5.5	less than 6.5	less than 6.5	less than 3.5	less than 4.5	1	
OR	less than 2	less than 2	[3.5, 4.5)	[2.5 7.5)	less than 5.5	>= 8.5	less than 6.5	[4.5, 5.5)	[6.5, 9.5)	0	
OR	less than 2	less than 2	less than 2.5	less than 1.5	>= 5.5	less than 6.5	>= 6.5	[4.5, 5.5)	less than 4.5	0	
OR	less than 2	less than 2	[3.5, 4.5)	[2.5 7.5)	less than 5.5	>= 8.5	less than 6.5	[4.5, 5.5)	less than 4.5	0	
III	less than 2	less than 2	less than 2.5	less than 1.5	less than 5.5	less than 6.5	less than 6.5	[4.5, 5.5)	less than 4.5	0	
III	less than 2	less than 2	[3.5, 4.5)	less than 1.5	>= 5.5	less than 6.5	>= 6.5	[3.5, 4.5)	less than 4.5	0	
III	less than 2	less than 2	less than 2.5	[1.5, 2.5)	less than 5.5	[6.5, 8.6)	less than 6.5	[4.5, 5.5)	less than 4.5	0	
III	#>= 2	less than 2	less than 2.5	less than 1.5	>= 5.5	less than 6.5	>= 6.5	[4.5, 5.5)	less than 4.5	0	
III	less than 2	>= 2	less than 2.5	[1.5, 2.5)	less than 5.5	[6.5, 8.6)	less than 6.5	>= 5.5	less than 4.5	0	
RJ	less than 2	>= 2	less than 2.5	less than 1.5	>= 5.5	less than 6.5	less than 6.5	>= 5.5	less than 4.5	0	
AD	less than 2	less than 2	[3.5, 4.5)	[2.5 7.5)	>= 5.5	[6.5, 8.6)	>= 6.5	>= 5.5	[5.5, 6.5)	0	
AD	less than 2	>= 2	[2.5, 3.5)	[7.5 9.5)	less than 5.5	>= 8.5	less than 6.5	>= 5.5	less than 4.5	0.5	
AD	less than 2	less than 2	[4.5, 5.5)	[2.5 7.5)	less than 5.5	>= 8.5	less than 6.5	less than 3.5	less than 4.5	0	

Figure 2: Snapshop of hypernasality dataset.

CHAPTER 4: OBJECT-DRIVEN AND TEMPORAL ACTION RULES

4.1 Motivation: From Concepts to Applications

In this section, we start by describing the characteristics of information systems that satisfy the object-driven and temporal nature; we begin by defining each, while providing examples of where the kinds of information systems that satisfy that nature exist, and why it should be examined differently; lastly we discuss the motivation for the kinds of solutions we provide while dealing with those types of datasets, and how to approach them accordingly.

In Section 1.2, we introduced the tabular representation of information systems, in which each row is denoting a complete observation; temporal information systems shall be represented similarly. However, each observation (or row) in our temporal information systems must be coupled with a time state (stamp), with the additional condition that time has a substantial meaning in that particular case. For example, an information system about the stock market or foreign exchange prices is doubtlessly a temporal information system; also a system of electronic medical records containing multiple observations about various visits of patients with time information is clearly a temporal system.

To understand how action rules should be extracted differently in temporal systems, we first provide a brief, but perhaps distinct, overview of the systematic approach

of the classical action rules. The notion of *support* introduced in Chapter 2 has a distinguished semantic meaning in the literature of action rules; the support of an action rules denotes the number of occurrences that have possibly occurred in our information system; assuming that our instances are independent and identically distributed. To clarify this concept, we present a graph:

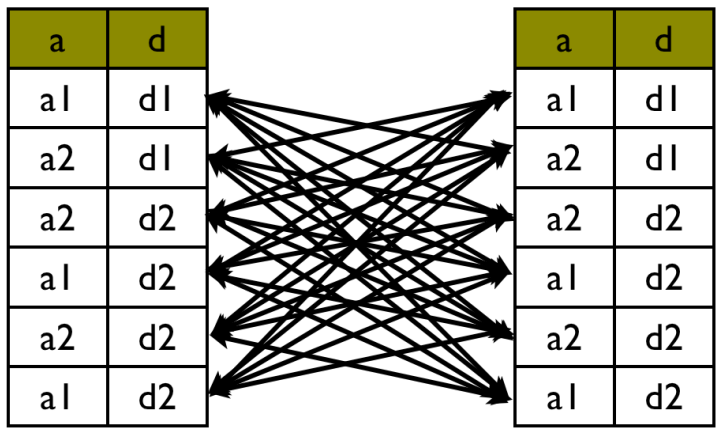

Figure 3: In the classical action rules approach, there is no restriction on the transition direction that could occur within our information system.

In Figure 3, we show a simplified information system composed of the following two attributes; a, a flexible attribute; and d, the decision attribute. The action rule we are interested in is the following: $(a_1 \rightarrow a_2) \Rightarrow (d_1 \rightarrow d_2)$. The arrows appearing in the graph represent possible transitions that could occur at some point of time. Based on our assumption of the independent and identically distributed information system, the instances in this simplified information system are free of any constraints, hence it is perfectly logical for any instance to transition to any other instance. For example, it would be valid to assume that the transition from the first row (a_1, d_1) to the fifth row (a_2, d_2) has actually occurred, also it would be equally valid to assume the

opposite, which denotes that the transition from the fifth row (a_2, d_2) to the first row (a_1, d_1) has occurred as well. In a temporal system however, the time state imposes additional constraints that permits us from making the same earlier assumptions. Again, we illustrate with a graph:

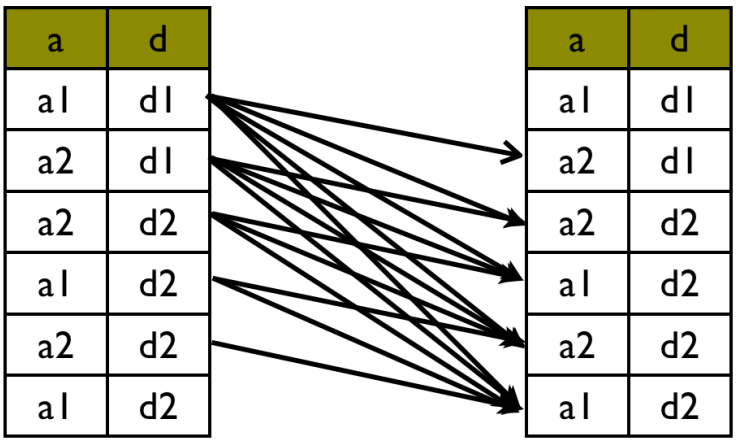

Figure 4: In the temporally-driven action rules approach on the other hand, there is a restriction on the transition direction that could occur within our information system.

As depicted in Figure 4, the direction of instances' transition is compelled with the temporal states attached with them; assuming the observations are ordered in a chronological order. Referring to the example from Figure 4, it would be invalid to assume that the fifth instance (a_2, d_2) transformed to the first instance (a_1, d_1), as for the first instance occurred before the fifth instance; again, for we are assuming our observations are chronologically ordered from earlier time to a later time. To that end, we propose subsequently in this chapter the appropriate changes applied to action rules extraction to adapt to the temporal constraint. By now, our readers should have formed a sufficient grasp of the notion of temporal based information

systems, and the need for the proper modifications to be addressed; as this is one main component of this thesis. Next, we examine the other main component, that is the object-driven nature.

Despite the intrinsic structure that calls for integration between both main components, namely the temporal and the object-driven, we are examining each individually. This will help to get a better understanding, also, cases that have one nature and not the other are equally common, hence equally important.

Information systems with object-driven nature are different than the classical information system, for that they have an attribute which we call the object attribute. Prior to illustrating object-driven information systems with an example, we provide two main conditions that must be satisfied in any object attribute: 1) for each distinct state of an object attribute, there must be multiple observations, 2) the possible values for an object attribute should be arbitrarily assigned; loosely speaking, and 3) all observations of a particular object attribute state should share the same distribution in a certain sense. To clarify these three characteristics of our object-driven information systems, we provide an example from the health field. Say we collect observations about patients in a hospital; for each patient, various observations were collected (in different points of time, though we will discard the timestamps for now to focus on the notion of object-driven); such information system could be depicted in Table 7.

As have probably been guessed, the *Patient ID* attribute is an ideal object attribute. Let us examine the three previously mentioned conditions that must be satisfied in any object attribute; as seen in Table 7, the first condition, which states that for

Table 7: Information system S_4

Patient ID	*Headache*	*Fatigue*	*Sneezing*	*Diagnosis*
1	*Rare*	*Rare*	*Yes*	*Cold*
1	*Rare*	*Rare*	*Yes*	*Cold*
1	*Yes*	*Yes*	*Sometimes*	*Flu*
2	*Yes*	*Yes*	*Sometimes*	*Flu*
2	*Yes*	*Yes*	*Sometimes*	*Flu*
2	*Rare*	*Rare*	*Yes*	*Cold*
2	*Rare*	*Rare*	*Yes*	*Cold*
3	*Yes*	*Yes*	*Sometimes*	*Flu*
3	*Rare*	*Rare*	*Yes*	*Cold*
3	*Yes*	*Yes*	*Sometimes*	*Flu*

each object attribute state there must be multiple observations, is clearly satisfied; (*Patient ID*, 1) has three observations, (*Patient ID*, 2) has four observations, and (*Patient ID*, 3) has three observations; the second condition of the object attribute is the arbitrariness condition, which states that the values of our object attribute should not have any particular meaning, again, it is clearly shown in Table 7 that this condition has been satisfied as well, the numbers from one to three do not have any particular meaning with respect to their corresponding observations; the third constraint states that all observations from the same object attribute need to be from the same distribution, which is also satisfied in our example, in something as personalized as patients, often it is wise to treat each patient (or each subset of patients) independently, as opposed to treating the whole set at once. In following sections, we show how by appropriately using our object attribute, we are able to build multiple subsystems for action rules extraction; object-driven assumption is the second main component of this work.

Lastly, it is worth mentioning that in general however, the two natures, namely the

temporal and the object-driven, are often found coupled together. In future sections, we provide a case study in which we combine the two assumptions to provide a complete system.

4.2 Object-Driven Assumption

The drive behind the introduction of object-driven action rules in [3] was to bring forth an adapted approach to extract action rules from systems of temporal and object-driven nature. In [3], we proposed an object-independency assumption that suggests extracting patterns from subsystems defined by unique objects, and then aggregating similar patterns amongst all objects. The motivation behind this approach is based on the fact that same-object observations share similar features that are not shared with other objects, and these features are possibly not explicitly included in our dataset. Therefore, by individualizing objects prior to calculating action rules, variance is reduced, and over-fitting is potentially avoided. In addition to the object-independency assumption, temporal information is exploited by taking into account only the state transitions that occurred in the valid direction. In this section, we limit our discussion to the object-independency assumption, and in later sections we discuss the temporal assumption.

Let $\mathbf{O}$ be the set of object ID's which instances belong to X. We define *object-driven action rules* to be rules that are extracted from a subsystem $S_p = (X_p, A, V)$ of S, where X_p contains all instances in X of the object p, $p \in \mathbf{O}$.

Here we provide an example to demonstrate the advantages of object-driven action rules extraction, using the information system S_5 shown in Table 8. As a company,

Table 8: Information system S_5. Each one of the two employees was observed four different times.

	ObjectID	*Observation*	*Loyalty*	*Income*	*Children*
x_0	1	1	High	High	More than 3
x_1	1	2	High	High	More than 3
x_2	1	3	Low	Medium	More than 3
x_3	1	4	Low	Medium	More than 3
x_4	2	1	High	Medium	Less than or equal to 3
x_5	2	2	High	Medium	Less than or equal to 3
x_6	2	3	Low	Low	Less than or equal to 3
x_7	2	4	Low	Low	Less than or equal to 3

we are interested in extracting action rules that change the state of attribute *Loyalty* from *Low* to *High*. We have two objects with 4 instances each.

Using the classical action rules extraction method, two action rules will be extracted from the system:

1. $r_1 = [[(Income, Low \rightarrow Medium) \wedge (Children, \leq 3)] \Rightarrow (Loyalty, Low \rightarrow High)]$; $conf(r_1) = 100\%$.

2. $r_2 = [[(Income, Medium \rightarrow High) \wedge (Children, > 3)] \Rightarrow (Loyalty, Low \rightarrow High)]$; $conf(r_2) = 100\%$.

Now let us assume that the attribute *Children* is missing in S_5. The following action rules will be extracted instead:

1. $\wp_1 = [(Income, Low \rightarrow Medium) \Rightarrow (Loyalty, Low \rightarrow High)]$;

   $conf(\wp_1) = 50\%$.

2. $\wp_2 = [(Income, Low \rightarrow High) \Rightarrow (Loyalty, Low \rightarrow High)]$;

   $conf(\wp_2) = 100\%$.

3. $\wp_3 = (Income, Medium) \Rightarrow (Loyalty, Low \rightarrow High)$;

   $conf(\wp_3) = 25\%$.

4. $\wp_4 = [(Income, Medium \rightarrow High) \Rightarrow (Loyalty, Low \rightarrow High)]$;

   $conf(\wp_4) = 50\%$.

We can observe that the rules $\wp_1, \wp_4$ are weaker than r_1, r_2 and also the condition and decision part in both rules $\wp_2$ and $\wp_3$ are referring to different objects, so we should not consider them valid. By using object-driven action rules extraction, we would not get $\wp_2$ and $\wp_3$.

Coming back to the action sets, we define the pth standard interpretation $N_{s(p)}$, where p is the object's unique ID, of action sets in $S = (X, A, V)$ as follows:

1. If $(a, a_1 \rightarrow a_2)$ is an atomic action set, then

$$N_{s(p)}((a, a_1 \rightarrow a_2)) = [\{x \in X_p : a(x) = a_1\}, \{x \in X_p : a(x) = a_2\}] \ ,$$

   where X_p is the set of all instances of the pth object.

2. If $t_1 = (a, a_1 \rightarrow a_2) \wedge t$ and $N_{s(p)}(t) = [Y_1, Y_2]$, then

$$N_{s(p)}(t_1) = [Y_1 \cap \{x \in X_p : a(x) = a_1\}, Y_2 \cap \{x \in X_p : a(x) = a_2\}] \ ,$$

   where X_p is the set of all instances of the pth object.

4.3 Temporal Assumption

In this section, we explore the notion of temporal datasets, and how to apply action rules extraction to them. As shown in the previous chapter, our dataset (hypernasal

speech disorder) for our case study is of temporal type and it contains information about patients' visits to the Children's Memorial Health Institute in Warsaw, Poland. The number of visits for each patient is ranging from 2 to 11. Patients are seen as objects represented in our dataset by minimum two and maximum eleven instances.

In a typical scenario of action rules extraction, we have no additional information about instances in a dataset besides values of their attributes. However, in the case of our medical dataset, we also assume that:

1. For each instance, we have a unique patient ID, which is utilized to extract *object-driven action rules*,

2. We also know that the visits are ordered for each patient, so for example, the $(y)^{th}$ visit for a specific patient, where $y > 1$, has occurred immediately after the $(y-1)^{th}$ visit; this information will allow us to add an ordered pairing restriction which we will call the *temporal constraint*.

The strategy of action rules construction, presented in Section 2.3 will be slightly modified to take into account the temporal nature of our dataset. This way, we believe, more refined action rules will be built leading to their higher accuracy and better generalization property.

4.3.1 Classical Object-Driven Approach

Here, we make the assumption that the only valid change of attribute value, is the change that happens between two instances of the same object. Accordingly, the standard interpretation N_s^{TC} that complies with the temporal constraint of an action set in $S = (X, A, V)$ is redefined as follows:

1. If $(a, a_1 \rightarrow a_2)$ is an atomic action set, then

$$N_s^{TC}((a, a_1 \rightarrow a_2)) = [\{x_1 \in X : a(x_1) = a_1\}, \{x_2 \in X : a(x_2) = a_2\}] = [X_1, X_2],$$

where for each $x_1 \in X_1$ there exist $x_2 \in X_2$ such that instance x_2 occurred after x_1, and there are no other objects in X_2.

2. If $t_1 = (a, a_1 \rightarrow a_2) \wedge t$ and $N_s^{TC}(t) = [Y_1, Y_2]$, then

$$N_s^{TC}(t_1) = [Y_1 \cap \{x_1 \in X : a(x_1) = a_1\}, Y_2 \cap \{x_2 \in X : a(x_2) = a_2\}] = [X_1, X_2],$$

where for each $x_1 \in X_1$ there exist $x_2 \in X_2$ such that instance x_2 occurred after x_1, and there are no other objects in X_2.

The definition of support of an action set and the definitions of support and confidence of an action rule are all the same as in the previous subsection.

Following both the definition of the **T**emporal **C**onstraint standard interpretation N_s^{TC} and the pth standard interpretation $N_{s(p)}$, it becomes apparent that the definition of the pth standard interpretation that complies with the Temporal Constraint, $N_{s(p)}^{TC}$, where p is the object's unique ID (in the following sections, p will be called an object), of action sets in $S = (X, A, V)$ can be defined as follows:

1. If $(a, a_1 \rightarrow a_2)$ is an atomic action set, then

$$N_{s(p)}^{TC}((a, a_1 \rightarrow a_2)) = [\{x_1 \in I_{s(p)} : a(x_1) = a_1\}, \{x_2 \in I_{s(p)} : a(x_2) = a_2\}] = [X_p^1, X_p^2],$$

where $\forall\, x_1\, \exists\, x_2$ such that x_2 is after x_1, and $I_{s(p)}$ is the set of all instances for the pth object. Additionally, we assume that there are no other instances in X_p^2.

2. If $t_1 = (a, a_1 \rightarrow a_2) \wedge t$ and $N^{TC}_{s(p)}(t) = [Y_1, Y_2]$, then

$$N^{TC}_{s(p)}(t_1) = [Y_1 \cap \{x_1 \in I_{s(p)} : a(x_1) = a_a\}, Y_2 \cap \{x_2 \in I_{s(p)} : a(x_2) = a_2\}] = [X^1_p, X^2_p],$$

where $\forall\, x_1\, \exists\, x_2$ such that x_2 is after x_1, and $I_{s(p)}$ is the set of all instances for the pth object. Additionally, we assume that there are no other instances in X^2_p.

If t is an action set and $N^{TC}_{s(p)}(t) = \{Y_1, Y_2\}$, then the support of t in S is defined as: $supp^{TC}_p(t) = min\{card(Y_1), card(Y_2)\}$.

Let $r = [t_1 \Rightarrow t_2]$ be an action rule, where $N^{TC}_{s(p)}(t_1) = [Y_1, Y_2], N^{TC}_{s(p)}(t_2) = [Z_1, Z_2]$. The pth support $supp^{TC}_p(r)$ and the pth confidence $conf^{TC}_p(r)$ of r are defined as follows:

$$supp^{TC}_p(r) = min\{card(Y_1 \cap Z_1), card(Y_2 \cap Z_2)\},$$

$$conf^{TC}_p(r) = \left[\frac{card(Y_1 \cap Z_1)}{card(Y_1)}\right] \cdot \left[\frac{card(Y_2 \cap Z_2)}{card(Y_2)}\right].$$

Now, assume that by O we mean the set of all objects' unique IDs. After all objects-driven action rules are extracted and their p^{th} support and p^{th} confidence are computed for each $p \in \mathbf{O}$, we then calculate their total support $supp^{TC}_{\mathbf{O}}(r)$ (called support) and total confidence $conf^{TC}_{\mathbf{O}}(r)$ (called confidence) following the definitions below:

$$supp^{TC}_{\mathbf{O}}(r) = \sum_{p \in \mathbf{O}} supp^{TC}_p(r),$$

$$conf_{\mathbf{O}}^{TC}(r) = \sum_{p \in \mathbf{0}} \left(\frac{supp_p^{TC}(r) \cdot conf_p^{TC}(r)}{supp_{\mathbf{O}}^{TC}(r)} \right).$$

Example to demonstrate the classical object-driven approach: Here, we provide an example to demonstrate how we calculate the support and the confidence for the whole system S_6 shown in Table 9 using the proposed classical object-driven approach.

Table 9: Information system S_6

	Patient ID	a	b	c	d
x_0	1	a_1	b_1	c_1	d_1
x_1	1	a_2	b_1	c_1	d_1
x_2	1	a_2	b_2	c_2	d_2
x_3	1	a_1	b_2	c_1	d_1
x_4	1	a_2	b_1	c_1	d_2
x_5	2	a_1	b_2	c_1	d_2
x_6	2	a_2	b_1	c_1	d_1
x_7	3	a_1	b_2	c_1	d_1
x_8	3	a_2	b_2	c_1	d_1
x_9	3	a_1	b_1	c_1	d_1
x_{10}	3	a_2	b_1	c_1	d_1

We assume that for each patient in Table 9, the instances are ordered in chronological order. For example, for the first patient (*Patient ID, 1*), we know that:

- x_4 has occurred after x_0, x_1, x_2, x_3
- x_3 has occurred after x_1, x_1, x_2, and before x_4
- x_2 has occurred after x_0, x_1, and before x_3, x_4
- x_1 has occurred after x_0, and before x_2, x_3, x_4
- x_0 has occurred before x_1, x_2, x_3, x_4

Referring to our information system S_6 shown in Table 9, we calculate the support $\sup_O^{TC}(r)$ and the confidence $\text{conf}_O^{TC}(r)$ for the following rule:

$$r = [(a, a_1 \rightarrow a_2) \wedge (c, c_1)] \Rightarrow (d, d_1 \rightarrow d_2)]$$

We first calculate the pth standard interpretation for each patient p for both the condition and decision parts in action rule r.

$N_{s(1)}^{TC}((a, a_1 \rightarrow a_2) \wedge (c, c_1)) = [\{x_0, x_3\}, \{x_1, x_4\}]$

$N_{s(1)}^{TC}(d, d_1 \rightarrow d_2) = [\{x_0, x_1, x_3\}, \{x_2, x_4\}]$

$N_{s(2)}^{TC}((a, a_1 \rightarrow a_2) \wedge (c, c_1)) = [\{x_5\}, \{x_6\}]$

$N_{s(2)}^{TC}(d, d_1 \rightarrow d_2) = [\phi, \phi]$

$N_{s(3)}^{TC}((a, a_1 \rightarrow a_2) \wedge (c, c_1)) = [\{x_7, x_9\}, \{x_8, x_{10}\}]$

$N_{s(3)}^{TC}(d, d_1 \rightarrow d_2) = [\{x_7, x_9\}, \{x_8, x_{10}\}]$

Using the temporal constraint and the object-driven assumption explained above, the support and confidence for each patient are calculated as follows:

$$\sup_1^{TC}(r) = min\{card(\{x_0, x_3\} \cap \{x_0, x_1, x_3\}), card(\{x_1, x_4\} \cap \{x_2, x_4\})\} = 1,$$

$$\text{conf}_1^{TC}(r) = \left[\frac{card(\{x_0, x_3\} \cap \{x_0, x_1, x_3\})}{card(\{x_0, x_3\})}\right] * \left[\frac{card(\{x_1, x_4\} \cap \{x_2, x_4\})}{card(\{x_1, x_4\})}\right] = 1 * \frac{1}{2} = .5$$

$$\sup_2^{TC}(r) = min\{card(\{x_5\} \cap \phi), card(\{x_6\} \cap \phi)\} = 0,$$

$$\text{conf}_2^{TC}(r) = \left[\frac{card(\{x_5\} \cap \phi)}{card(\{x_5\})}\right] * \left[\frac{card(\{x_6\} \cap \phi)}{card(\{x_6)}\right] = 0$$

$$\sup_3^{TC}(r) = min\{card(\{x_7, x_9\} \cap \{x_7, x_9\}), card(\{x_8, x_{10}\} \cap \{x_8, x_{10}\})\} = 2,$$

$$\text{conf}_3^{TC}(r) = \left[\frac{card(\{x_7, x_9\} \cap \{x_7, x_9\})}{card(\{x_7, x_9\})}\right] * \left[\frac{card(\{x_8, x_{10}\} \cap \{x_8, x_{10}\})}{card(\{x_8, x_{10}\})}\right] = 1$$

Now we calculate the overall support and overall confidence for the entire system:

$$\sup_O^{TC}(r) = 3,\ \text{conf}_O^{TC}(r) = \left(\frac{1 * .5}{3}\right) * \left(\frac{0 * 0}{3}\right) * \left(\frac{2 * 1}{3}\right) = .83$$

It is important to keep in mind that regardless of the methodology used to extract action rules, the way the support and confidence are calculated for object-driven action rules will stay the same. In the following case study, we use the approach proposed in [23]; explained through an example in previous section. However, in future studies, additional methodologies could be used to extract object-driven action rules, as long as they comply with both the temporal constraint, and the object-driven assumption.

4.3.2 Results of Applying Classical Object-Driven Approach to Hypernasality

In this section, we show a sample of the results after running our classical object-driven approach. Here are few results and their explanations:

Rule 1. $r_1 = [(palatine\ tonsils, 0) \wedge (i - long, \geq 9.5) \wedge (i_2 - long, \geq 5.5 \rightarrow < 5.5)] \Rightarrow (Czermak's\ mirror\ test, 2 \rightarrow 1.5);\quad supp(r_1) = 2,\ conf(r_1) = 66.7\%.$

Rule 1 means that if our patient has no hypertrophied palatine tonsils (value 0), and $i - long$ (nasalization for /i/-long) is more than 9.5, then decreasing the rate of change of $i - long$ ($i_2 - long$) from more than or equal to 5.5, to less than 5.5, would

improve patient's *Czermak's mirror test* from 2 to 1.5.

Rule 2. $r_2 = [(palatine\ tonsils, 1) \wedge (i - long, \geq 9.5) \wedge (i_2 - long, \geq 5.5 \rightarrow < 5.5)] \Rightarrow (Czermak's\ mirror\ test, 2 \rightarrow 1.5);\quad supp(r_2) = 2,\ conf(r_2) = 66.7\%.$

Rule 2 means that if our patient has a bit hypertrophied palatine tonsils (value 1), and $i - long$ is more than 9.5, then decreasing the rate of change of $i - long$ ($i_2 - long$) from more than or equal to 5.5, to less than 5.5, would improve patients *Czermak's mirror test* from 2 to 1.5.

Rule 3. $r_3 = (i - long, \geq 9.5 \rightarrow [2.5, 7.5)) \Rightarrow (Czermak's\ mirror\ test, 1 \rightarrow .5);$ $supp(r_3) = 2,\ conf(r_3) = 66.7\%.$

Rule 3 means that if we decrease the value of $i - long$ from greater than 9.5 to [2.5, 7.5), we would improve the patient's *Czermak's mirror test* from 1 to .5. Decreasing of $i - long$, i.e. the decrease of nasalization for /i/-long can be achieved, to some extend, through speech therapy, or (eventually) surgically through a nasopharynx surgery, moving posterior pharyngeal wall forward, towards soft palate. However, none of the examined patients underwent this surgery.

These rules suggest decreasing palatine tonsils, and decreasing the nasalization coefficient for /i/-long. This confirms the importance of these attributes, which was expected, but not so obvious in the case of $i - long$. However, decreasing palatine tonsils causes only slight change of the Czermak's test, and this is also a confirmation for physicians that surgery on palatine tonsils might not be particularly beneficiary for patients. Palatine tonsils tend to regrow after surgery, and their influence on nasality is not so high, because of their location. At the same time, hypernasality can

be treated, to some extent, through speech therapy aiming at decreasing nasalization for /i/-long.

Another interesting observation in our resulted action rules is that we may desire different transitions for the same attribute, depending on the current value of the *Czermak's mirror test*. For example, if our patient's *Czermak's mirror test* is 2, we would want for $i - long$ to stay greater than 9.5 while changing $i_2 - long$. However, if our patient's *Czermak's mirror test* is 1, we would want to decrease the value of $i - long$ from greater than 9.5 to [2.5 7.5).

Rule 4. $r_4 = [(palatine\ tonsils, < 2) \wedge (i_2 - long, < 5.5) \wedge (motility, [4.5, 5.5)) \wedge (i - long,\ [1.5, 2.5) \rightarrow < 1.5)] \Rightarrow (Czermak's\ mirror\ test, 0.5 \rightarrow 0)$; $supp(r_4) = 3$, $conf(r_4) = 60\%$.

Rule 5. $r_5 = [(palatine\ tonsils, < 2) \wedge (bdg_1, < 6.5) \wedge (motility, [4.5, 5.5)) \wedge (i - long, [1.5, 2.5) \rightarrow < 1.5)] \Rightarrow (Czermak's\ mirror\ test, 0.5 \rightarrow 0)$; $supp(r_5) = 3$, $conf(r_5) = 60\%$.

The above rules mean that very slight hypernasality might be completely removed by decreasing the nasalization coefficient for /i/-long, provided the indicated accompanying conditions are fulfilled.

Rule 6. $r_6 = [(sleep\ apnoea, < 2 \rightarrow > 2) \wedge (bdg, > 8.5 \rightarrow [6.5, 8.5)] \Rightarrow (Czermak's\ mirror\ test, 1.5 \rightarrow 0)$; $supp(r_6) = 2$, $conf(r_6) = 100\%$.

This rule is very interesting because it shows that the increase of sleep apnoea (along with decreasing the nasality of /bdg/) cures light-medium hypernasality.

Rule 7. $r_7 = [(palatine\ tonsils, < 2) \wedge (motility, < 3.5 \rightarrow [4.5, 5.5)) \wedge (difference$

$level\ F1 - F2, [5.5, 6.5) \rightarrow < 4.5)] \Rightarrow (Czermak's\ mirror\ test, 1.5 \rightarrow 0); \quad supp(r_7) = 2,\ conf(r_7) = 100\%$.

Rules 6 and 7 are especially interesting because their confidence is 100%, and they both induce a significant shift in Czermak's test, from 1.5 to 0, i.e. curing light-medium hypernasality completely, through procedures increasing the motility of the soft palate, and decreasing the difference for the first two formants of the vocal tract for /i/-long.

4.3.3 Pair-Based Object-Driven Approach

As defined previously, temporal object-driven datasets consist of numerous unique objects, where each object is comprised of multiple instances that have assigned corresponding timestamps. Previously in [3]; as explained in section 4.3.1, the object p based standard interpretation of an action set $t = (a, a_1 \rightarrow a_2)$ was defined as the pair of two sets $[Y_1, Y_2]$ where Y_1 is the set of instances of the object p that satisfy the left side, or condition side, of the action set, and Y_2 is the set of instances of the object p that satisfy the right side, or decision side, of the action set, with the addition that for every instance in Y_1, there exist a matching instance in Y_2 that occurred after it. This definition resembles the definition of standard interpretation for classical action rules while restricting valid transitions to only one direction. In this section however, we present an approach in which the nature of the object-driven temporal dataset allows us to redefine the standard interpretation into a more intuitive pair-based structure which we believe is more appropriate for object-driven temporal systems.

Let us first assume that $I_s(p)$ denotes the set of all instances of the object p in an

information system S. Also, the relation $\angle \subseteq I_s(p)$ is defined as:

$(x_1, x_2) \in \angle$ iff x_2 has occurred after x_1.

The pair-based standard interpretation $N^{TC}_{s(p)}$ in $S = (X, A, V)$ for an object p is redefined as:

1. If $(a, a_1 \rightarrow a_2)$ is an atomic action set, then

   $N^{TC}_{s(p)}((a, a_1 \rightarrow a_2)) = \{(x_1, x_2) \in \angle : a(x_1) = a_1, a(x_2) = a_2\}$

   where $\angle \subset I_s(p)$.

2. If $t_1 = (a, a_1 \rightarrow a_2) \wedge$ and $N^{TC}_{s(p)}(t) = Y_1$, then

   $N^{TC}_{s(p)}(t_1) = Y_1 \cap \{(x_1, x_2) \in \angle : a(x_1) = a_1, a(x_2) = a_2\}$

   where $\angle \subset I_s(p)$.

In other words, our standard interpretation will consist of all valid transitions from the left side of an action set to the right side, represented as pairs. The motivation behind this new interpretation is due to the fact that the instances within one object are not observed independently, which will allow us to relax the minimum assumption previously used. Our object-independency assumption states that the whole system consists of multiple independent subsystems, each one marked by a unique object. Although it confines the system to extract action rules only from instances of the same object, it provides more flexibility to be applied within unique objects.

If t is an action set and $N^{TC}_{s(p)}(t) = Y_1$, then the support of t in S is defined as: $supp^{TC}_p = card(Y_1)$.

Let $r = [t_1 \Rightarrow t_2]$ be an action rule, where $N^{TC}_{s(p)}(t_1) = Y_1, N^{TC}_{s(p)}(t_2) = Y_2$. The pth

support $supp_p^{TC}(r)$ and the pth confidence $conf_p^{TC}(r)$ of r are defined as follows:

$$supp_p^{TC}(r) = card(Y_1 \cap Y_2),$$

$$conf_p^{TC}(r) = \left[\frac{card(Y_1 \cap Y_2)}{card(Y_d)}\right].$$

To define Y_d, let us first assume that $\zeta(Y)$ denotes the set of first elements of the set of pairs Y. For instance, if $Y = \{(x_1, x_3), (x_3, x_4), (x_1, x_2)\}$, then $\zeta(Y) = \{x_1, x_3\}$. We define $Y_d = \{(x_1, x_2) \in Y_1 : x_1 \in \zeta(Y_2)\}$. The interpretation of this definition means that to calculate the confidence of the action rule $r = (a, a_1 \rightarrow a_2) \Rightarrow (d, d_1 \rightarrow d_2)$, the pairs that we are considering are the ones that have first elements that satisfy $a = a_1$ and $d = d_1$. Since the transition from a_1 to a_2 could possibly trigger other states of decision attribute d, we are only interested in the states of our action rule.

After all object-driven action rules are extracted and their p^{th} support and p^{th} confidence are computed for all $p \in O$, we then calculate their total support $supp_O^{TC}(r)$ (called support) and total confidence $conf_O^{TC}(r)$ (called confidence) following the definition below:

$$supp_O^{TC}(r) = \sum_{p \in O} supp_p^{TC}(r)\ ,$$

$$conf_O^{TC}(r) = \sum_{p \in O} \left(\frac{supp_p^{TC}(r) * conf_p^{TC}(r)}{supp_O^{TC}(r)}\right).$$

If the denominator in the formula for calculating confidence is equal to zero, then the confidence is equal to zero by definition.

Example to demonstrate pair-based object driven support and confidence: here, we provide an example to demonstrate how we calculate the support and the confidence for the whole system S_7 shown in Table 10. We assume that for all 3 objects in X

their instances x_i, where $1 \leq i \leq 10$, have chronological order.

Table 10: Information system S_7

	$objectID$	a	b	c	d
x_0	1	a_1	b_1	c_1	d_1
x_1	1	a_2	b_1	c_1	d_1
x_2	1	a_2	b_2	c_2	d_2
x_3	1	a_1	b_2	c_1	d_1
x_4	1	a_2	b_1	c_1	d_2
x_5	2	a_1	b_2	c_1	d_2
x_6	2	a_2	b_1	c_1	d_1
x_7	3	a_1	b_2	c_1	d_1
x_8	3	a_2	b_2	c_1	d_2
x_9	3	a_1	b_1	c_1	d_1
x_{10}	3	a_2	b_1	c_1	d_2

Referring to our information system S shown in Table 1, we calculate the support $supp_O^{TC}(r)$ and the confidence $conf_O^{TC}(r)$ for the following rule:

$$r = [(a_1 \rightarrow a_2) \wedge (c, c_1) \Rightarrow (d, d_1 \rightarrow d_2)].$$

We first calculate the pth standard interpretation for each object p (e.g. for a patient) for both the condition and the decision parts in the action rule r:

$N_{s(1)}^{TC}((a, a_1 \rightarrow a_2) \wedge (c, c_1)) = \{(x_0, x_1), (x_0, x_2), (x_0, x_4), (x_3, x_4)\} \cap$

$\{(x_0, x_1), (x_0, x_3), (x_0, x_4), (x_1, x_3), (x_1, x_4), (x_3, x_4)\}$

$= \{(x_0, x_1), (x_0, x_4), (x_3, x_4)\}$,

$N_{s(1)}^{TC}(d, d_1 \rightarrow d_2) = \{(x_0, x_2), (x_0, x_4), (x_1, x_2), (x_1, x_4), (x_3, x_4)\}$,

$N_{s(2)}^{TC}((a, a_1 \rightarrow a_2) \wedge (c, c_1)) = \{(x_5, x_6)\}$,

$N_{s(2)}^{TC}(d, d_1 \rightarrow d_2) = \phi$,

$N^{TC}_{s(3)}((a, a_1, \rightarrow a_2) \wedge (c, c_1)) = \{(x_7, x_8), (x_7, x_{10}), (x_9, x_{10})\} \cap$

$\{(x_7, x_8), (x_7, x_9), (x_7, x_{10}), (x_8, x_9), (x_8, x_{10}), (x_9, x_{10})\}$

$= \{(x_7, x_8), (x_7, x_{10}), (x_9, x_{10})\}$,

$N^{TC}_{s(3)}(d, d_1 \rightarrow d_2) = \{(x_7, x_8), (x_7, x_{10}), (x_9, x_{10})\}$.

Using the temporal constraint and the object-driven assumptions, the pair-based support and confidence for each object is calculated as follows:

$$sup^{TC}_1(r) = card(\{(x_0, x_4), (x_3, x_4)\}) = 2 \ ,$$

$$conf^{TC}_1(r) = \left[\frac{card(\{(x_0, x_4), (x_3, x_4)\})}{card(\{(x_0, x_1), (x_0, x_4), (x_3, x_4)\})}\right] = \frac{2}{3} \ ,$$

$$sup^{TC}_2(r) = card(\phi) = 0 \ ,$$

$$conf^{TC}_2(r) = 0 \ ,$$

$$sup^{TC}_3(r) = card(\{(x_7, x_8), (x_7, x_{10}), (x_9, x_{10})\}) = 3 \ ,$$

$$conf^{TC}_3(r) = \left[\frac{card(\{(x_7, x_8), (x_7, x_{10}), (x_9, x_{10})\})}{card(\{(x_7, x_8), (x_7, x_{10}), (x_9, x_{10})\})}\right] = \frac{3}{3} = 1 \ .$$

Now we calculate the overall support and confidence for the whole system:

$$sup^{TC}_O(r) = 5, \ conf^{TC}_O(r) = \left(\frac{2 * \frac{2}{3}}{5}\right) + \left(\frac{3 * 1}{5}\right) = \frac{4.33}{5} = .87 \ .$$

4.3.4 Results of Applying Pair-Based Object-Driven Approach to Hypernasality

In this section, we show a sample of results after running our pair-based approach to extract object-driven action rules from temporal systems. We show that by using the pair-based approach, not only we were able to extract a larger set of action rules,

but also we were able to extract action rules that provide more dramatic decrease of patient severity than the rules extracted in [3]. For an action rule to be eligibly used on a patient, the pre-conditions of the action rule and the patient's current condition have to match, meaning that only a subset of our patients will benefit from each particular action rule. Having said that, using our pair-based approach to extract action rules we will generate a significant amount of action rules that can be appropriately used for various sets of patients.

Rule 1. $r_1 = (\textit{difference level F1-F2}, \geq 9.5 \rightarrow [6.5, 9.5))$
$\Rightarrow (\textit{Czermak's mirror test}, 3 \rightarrow 2); \quad supp(r_1) = 2, \quad conf(r_1) = 100\%$.

This rule means that by decreasing the difference between the first two formants of the vocal tract for /i/ - long, we would notice a decent shift of the Czermak's mirror test, decreasing from 3 to 2. In [3], we extracted a similar action rule that also indicated the importance of *difference level F1-F2* attribute. However, this action rule is exclusive to the work described in this chapter.

Rule 2. $r_2 = (i_2 - long, \geq 5.5 \rightarrow < 5.5) \Rightarrow (\textit{Czermak's mirror test}, 3 \rightarrow 2)$; $supp(r_2) = 3, \quad conf(r_2) = 66.7\%$.

This rule means that decreasing the value of $i - long$ in a short period of time, since $i_2 - long$ is defined as the rate of change, will result in a similar decrease of the Czermak's mirror test from 3 to 2. Again, this rule affirms the importance of the attribute $i - long$.

Rule 3. $r_3 = (i_2 - long, \geq 5.5 \rightarrow < 5.5) \wedge (bdg, \geq 8.5)$
$\Rightarrow (\textit{Czermak's mirror test}, 2.5 \rightarrow 2); \quad supp(r_3) = 2, \quad conf(r_3) = 100\%$.

This rule is similar to Rule 1. It confirms the effect of decreasing the rate of change of the nasalization measured for /i/ - long, but also adds an additional condition concerning the nasality of /bdg/, that is, this rule only applies to patients suffering from high nasality for /bdg/ (≥ 8.5).

Rule 4. $r_4 = (tonsils, < 2) \wedge (i_2 - long, \geq 5.5 \rightarrow < 5.5) \wedge (motility, [4.5, 5.5)) \Rightarrow$ $(Czermak's\ mirror\ test, 2 \rightarrow 1.5)$; $supp(r_4) = 2$, $conf(r_4) = 100\%$.

This rule states that when a patient is experiencing a little hypertrophied adenoids and possibly palatine tonsils ($tonsils < 2$), we can slightly improve his condition from Czermak's mirror test 2 to 1.5 by decreasing the rate of change in /i/ - long, and if the motility of the soft palate does not change.

Rule 5. $r_5 = (bdg, \geq 8.5 \rightarrow [6.5, 8.5)) \Rightarrow (Czermak's\ mirror\ test, 1 \rightarrow .5)$; $supp(r_5) = 3$, $conf(r_5) = 66.7\%$.

This rule states that by only decreasing the nasality of /bdg/, we would be able to shift the patients' Czermak's mirror test state from 1 to .5.

Rule 6. $r_6 = (i - long, \geq 9.5 \rightarrow [2.5, 7.5)) \Rightarrow (Czermak's\ mirror\ test, 1 \rightarrow .5)$; $supp(r_6) = 2$, $conf(r_6) = 100\%$.

Although the support of this action rule is not high, the rule is rather interesting. It states that by decreasing only one attribute; /i/ - long, there is a 100% chance that the Czermak's mirror test will shift from 1 to .5.

Rule 7. $r_7 = (motility, < 3.5 \rightarrow [4.5, 5.5)) \wedge (diagnosis, OR)$
$\Rightarrow (Czermak's\ mirror\ test, 1 \rightarrow 0)$; $supp(r_7) = 3$, $conf(r_7) = 100\%$.

This rule states that if a patient has gone through a cleft palate surgery (OR), then

increasing the motility of the soft palate would significantly improve the patient's condition, to the level where the patient is entirely cured, which will result in shifting the Czermak's mirror test value from 1 to 0.

Rule 8. $r_8 = (i_2 - long, \geq 5.5 \rightarrow < 5.5) \Rightarrow (Czermak's\ mirror\ test, .5 \rightarrow 0)$; $supp(r_8) = 7, \quad conf(r_8) = 71\%$.

This rule has a relatively high support. It states that decreasing the rate of change of $i - long$ from greater than or equal to 5.5, to less than 5.5, will result in curing a light hypernasality.

Rule 9. $r_9 = (i_2 - long, \geq 5.5 \rightarrow < 5.5) \wedge (sleep\ apnoea, < 2)$
$\Rightarrow (Czermak's\ mirror\ test, .5 \rightarrow 0); \quad supp(r_9) = 6, \quad conf(r_9) = 83\%$.

This rule is a similar, but more specific than rule 8. By expanding the condition side of action rules, we are able to generate action rules with higher confidence. Rule 8 states that by only decreasing the rate of change of $i - long$, we would have a 71% chance of shifting the Czermak's mirror test from .5 to 0. However, rule 9 states that by decreasing the rate of change of $i - long$ and maintaining a low value of sleep apnoea, we would have an 83% chance of shifting Czermak's mirror test from .5 to 0.

Rule 10. $r_{10} = (tonsils, \geq 2 \rightarrow < 2) \Rightarrow (Czermak's\ mirror\ test, .5 \rightarrow 0)$; $supp(r_9) = 5, \quad conf(r_9) = 100\%$.

This rule states, with absolute certainty (confidence 100%), that by decreasing the hypertrophied adenoids and possibly palatine tonsils that the patient is experiencing, the Czermak's mirror test will shift from .5 to 0. Although the improvement does not appear to be significant, the high support and high confidence make this rule highly

valuable.

In our hypernasality dataset, most of the patients were experiencing slight to no hypernasality speech (Czermak's mirror test .5 or 0). As a consequence, the last three action rules had a much higher support compared to the others.

CHAPTER 5: ADDITIONAL DATASET: STATE INPATIENT DATASET

5.1 Overview of State Inpatient Dataset

The Healthcare Cost and Utilization Project (HCUP) State Inpatient Databases (SID) consist of records collected from data organizations (essentially, hospitals) from across 47 participating States. In this work, we explore records collected (spanning multiple years) from institutions in Florida. Although we explore data collected from only one state, the number of records (and attributes) is still considerably large. Most records are collected from acute care institutions, which generally means that patients receive active but short-term treatment for a severe injury; Table 11 shows few of the most common diagnoses observed in patients.

Table 11: Most common diagnoses in the state inpatient dataset

Diagnosis	*Description*
Hypertension	Abnormally high blood pressure.
Hyperlipidemia	Abnormally high concentration of fats in the blood.
Diabetes	Diabetes mellitus without mention of complication.
Coronary atherosclerosis	Coronary atherosclerosis of native coronary artery.
Esophageal reflux	Condition in which the stomach contents leak backwards from the stomach into the esophagus.
Anemia	Condition marked by a deficiency of red blood cells or of hemoglobin in the blood.
Congestive heart failure	Condition occurs when your heart muscle does not pump blood as well as it should.
Atrial fibrillation	Irregular and often rapid heart rate that commonly causes poor blood flow to the body.
Hypothyroidism	Abnormally low activity of the thyroid gland, resulting in retardation of growth.
Urinary tract infection	Infection in any part of your urinary system.

The State Inpatient Dataset is both temporally-based and object-driven, which means that there were multiple visits recorded for each patient (at least the majority of them); and for each visit, a timestamp was registered. In each visit, the patient was given from 1 to 31 different diagnoses; some patients have as little as one diagnosis, while others have as many as thirty-one diagnoses. Note here that this does not mean that in this dataset we only have 31 different diagnoses; in fact, this dataset contains a tremendous number of different diagnoses (roughly 10 thousand unique diagnoses); by 31, we only mean that for one particular visit, the patient was given at most 31 different diagnoses. Clearly, this dataset is drastically more general than our Hypernasality dataset, in the sense that patients are diagnosed with many (and often different) diagnoses, which will require us to build a rather more elaborate rule extraction system, as will be shown in the next section. Table 12 shows all attributes that are considered. The number of unique patients in the dataset is almost 1.5 million; for each patient, there were 2 to 39 visits recorded; although we do not have access to the exact date of each visit, the difference (in days) between any two visits (for a unique patient) is recorded.

Table 12: Attributes that we used in the state inpatient dataset

Attribute	*Description*
Patient ID	Each patient has a unique ID
Age (year)	Age of the patient, time of visit
Admission Type	{Emergency, Urgent, Elective, Newborn}
Diagnosis	From 1 to 31 diagnosis for each visit
Sex	{Male, Female}
Race	{White, Black, Hispanic, Asian, Native American, Other}
Procedure	From 1 to 30 procedures for each visit
Days to Event	This value indicates the difference in days between visits

Both the diagnoses of the patient and the procedure(s) operated on the patient, are extremely essential for our action rule extraction system; similar to the diagnosis attribute, there were often multiple procedures operated on the patient in each visit, ranging from 1 to 30 different procedures, while the number of different procedures recorded in this dataset is roughly 3 thousand.

In the next section, we will explain how to extract atomic action sets from such complex dataset; surely, we would still exploit our temporal-based (and object-driven) approach, but the main challenge will be on how to transition from one diagnoses to an entirely different one, since as we mentioned earlier, our dataset deals with all various kinds of diseases (and diagnoses). Table 13 shows the most common procedures operated on patients.

Table 13: Most common procedures in the state inpatient dataset

Procedure	*Description*
Packed cell transfusion	Transfusion of packed cells (or blood).
Venous catheterization	Long, thin, flexible tube used to give medicines, fluids, nutrients, or blood products over a long period of time.
Coronary arteriography	Coronary arteriography using two catheters
Cont inv mec ven $<$ 96 hrs	Continuous invasive mechanical ventilation for less than 96 consecutive hours.
Insert endotracheal tube	the placement of a flexible plastic tube into the trachea to maintain an open airway or to serve as a conduit through which to administer certain drugs.
Hemodialysis	Kidney dialysis.
Left heart cardiac cath	Left heart cardiac catheterization.
Repair ob laceration	Repair of other current obstetric laceration.

The key here is to extract atomic action sets from each diagnosis to each other diagnosis; similarly, atomic action rules are extracted from each procedure to each

other procedure, for each two visits. More details are provided in the next section.

5.2 Real-Time Action Rules Extraction Approach

There are three main characteristics that distinguishes the State Inpatient Dataset from other more traditional datasets (such as the Hypernasality treatment):

1. *Some columns are inconsistent*; by that, we mean that (for some attributes) the same column may refer to states that are entirely different (and unrelated to each other), this applies to both the diagnosis attributes and the procedure attributes; since each record (or visit) can contain a maximum of 31 diagnoses, this means that there are 31 attributes designated to diagnoses; having said that, the diagnoses are not semantically ordered. For example, if we observe the states (values) of one particular diagnosis attribute (or column), we may observe the diagnoses is *Hypertension* in the first visit, and in the same column the diagnosis is *cogestive heart failure* in the second visit. This therefore means that transitions from states within the same column for the diagnoses should not be performed, instead we take into consideration all possible transitions from each diagnosis in the first visit, to each other diagnosis in the second visit. For example, if we observe the following diagnoses for the first visit: $\{diag1, diag2, diag3\}$, and we observe the following diagnoses for the second visit {diag4, diag5}, then the atomic action sets that are generated are the following: $\{(diag1 \rightarrow diag4), (diag1 \rightarrow diag5), (diag2 \rightarrow diag4), (diag2 \rightarrow diag5), (diag3 \rightarrow diag4), (diag3 \rightarrow diag5)\}$, which is essentially the set of all possible pairs; we use the same methodology to generate atomic action sets for

the procedure attributes.

2. *Extreme number of attribute states*: As mentioned earlier, the number of valid diagnoses and the number of valid procedures is extreme; in this dataset, we have more than 10 thousand unique diagnoses, and more than 3 thousand procedures, which means that the number of unique atomic action sets will be roughly 100 million (for only the diagnosis attributes), and almost 1 million (for only the procedure attributes). Note here that in the association action rule extraction methodology, we build frequent action sets from frequent atomic action sets; which means that this number will grow quickly (and drastically). To solve this problem, we propose an active (or real-time) action rule generation approach; this novel approach is called real-time (or active) because it will allow action rules to get generated only when needed (in our case, when a new patient comes to the hospital). In other words, instead of constructing all (complete) action rules before we start applying (or testing) our models to new instances, we only construct atomic action sets, and when testing is required (new patient comes to the hospital), we use the set of already constructed atomic action sets to build complete action rules. By using this approach, we will only need to keep track of atomic action sets, which would save us tremendous resources (both speed and memory) in cases such as this one; when we are dealing with large datasets.

3. *Unknown desired/undesired states:* Since the set of all possible diagnoses is terribly large (and diverse), it would be impossible to sort them in an order

that implies their desire levels. For example, it would not make any logical sense to ask (doctors) the question of whether *diabetes* is more (or less) desired than *Hyperlipidemia*, since the two are basically distinct diagnoses; hence (for the most part) unrelated. To solve this problem, we propose two approaches, the first one is to generate essentially all action rules that transforms diagnoses with respect to procedures; so when a new patient gets admitted to the hospital, we observe his/her current diagnoses (and possibly the last procedure operated on him/her), then we extract a set of candidate procedures that were essentially operated on similar patients in the past, and from each procedure we generate complete action rules that show the transitions in diagnoses, by observing the set of anticipated diagnoses (for each procedure), the doctor then proceeds with the most appropriate procedure. The second approach is to further filter action rules by allowing the doctor to specify the procedure that he/she would like to perform on the patient, and then the set of diagnoses transitions are presented to the doctor. Note here that the system allows users (essentially doctors) to specify more than one operation to be performed on the patient.

Next, we show a complete example of how to build our *real-time action rule extraction system*. Shown in Table 17 is a simplified object-driven (and temporally based) information system; we assume that for each patient, the instances are presented in chronological order.

Using the pair-based approach, we extract the following diagnosis atomic action sets:

Table 14: Information system S_8

	Patient ID	*diagnosis1*	*diagnosis2*	*diagnosis3*	*procedure1*	*procedure2*
x_0	1	d_1	d_3	d_2	p_1	p_0
x_1	1	d_5	d_6	d_0	p_3	p_4
x_2	2	d_5	$-$	$-$	p_2	$-$
x_3	2	d_1	d_2	$-$	p_1	p_3
x_4	2	d_1	d_3	d_4	p_1	$-$
x_5	3	d_8	d_7	$-$	p_4	p_3
x_6	3	d_2	$-$	$-$	p_3	p_0
x_7	4	d_8	$-$	$-$	p_1	$-$
x_8	4	d_2	d_3	$-$	p_2	p_5
x_9	4	d_4	$-$	$-$	p_1	$-$
x_{10}	4	d_4	d_3	$-$	p_2	p_5

- From (Patient ID, 1): $\{(d_1 \rightarrow d_5, support : 1), (d_1 \rightarrow d_6, support : 1), (d_1 \rightarrow d_0, support : 1), (d_3 \rightarrow d_5, support : 1), (d_3 \rightarrow d_6, support : 1), (d_3 \rightarrow d_0, support : 1), (d_2 \rightarrow d_5, support : 1), (d_2 \rightarrow d_6, support : 1), (d_2 \rightarrow d_0, support : 1)\}$

- From (Patient ID, 2): $\{(d_5 \rightarrow d_1, support : 2), (d_5 \rightarrow d_2, support : 1), (d_1 \rightarrow d_1, support : 1), (d_1 \rightarrow d_3, support : 1), (d_1 \rightarrow d_4, support : 1), (d_2 \rightarrow d_1, support : 1), (d_2 \rightarrow d_3, support : 1), (d_2 \rightarrow d_4, support : 1)\}$

- From (Patient ID, 3): $\{(d_8 \rightarrow d_2, support : 2), (d_7 \rightarrow d_2, support : 1)\}$

- From (Patient ID, 4): $\{(d_8 \rightarrow d_2, support : 2), (d_8 \rightarrow d_3, support : 2), (d_8 \rightarrow d_4, support : 2), (d_2 \rightarrow d_4, support : 2), (d_2 \rightarrow d_3, support : 1), (d_3 \rightarrow d_4, support : 2), (d_3 \rightarrow d_3, support : 1), (d_4 \rightarrow d_4, support : 1), (d_4 \rightarrow d_3, support : 1)\}$

Note that when we transform from one diagnosis to the same, we still consider the atomic action set e.g. (Patient ID, 2): $(d_1 \rightarrow d_1, support : 1)$; also note here that for each atomic action set, we keep track of the instances that satisfy the left hand side, and the instances that satisfy the right hand side; this would be the only way

to build action rules from atomic action sets. This should become more clear as we continue with our example. To keep track of all the atomic action sets (and their support), one efficient way would be to construct a matrix of diagnoses codes; this will allow O(1) for accessing atomic action sets. Table 15 shows such matrix for our diagnosis attributes (for this example); note that this is not a symmetric matrix, since the atomic action set $(d_x \rightarrow d_y)$ does not indicate $(d_y \rightarrow d_x)$.

Table 15: Matrix of atomic action sets for the *diagnosis* attribute

	d_0	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
d_0	0	0	0	0	0	0	0	0	0
d_1	1	1	0	1	1	1	1	0	0
d_2	1	1	0	2	3	1	1	0	0
d_3	1	0	0	1	2	1	1	0	0
d_4	0	0	0	1	1	0	0	0	0
d_5	0	2	1	1	1	0	0	0	0
d_6	0	0	0	0	0	0	0	0	0
d_7	0	0	1	0	0	0	0	0	0
d_8	0	0	2	2	2	0	0	0	0

Similarly, we do the same for our procedures; using the pair-based approach, we extract the following procedure atomic action sets:

- From (Patient ID, 1): $\{(p_1 \rightarrow p_3, support : 1), (p_1 \rightarrow p_4, support : 1), (p_0 \rightarrow p_3, support : 1), (p_0 \rightarrow p_4, support : 1)\}$
- From (Patient ID, 2): $\{(p_2 \rightarrow p_1, support : 2), (p_2 \rightarrow p_3, support : 1), (p_1 \rightarrow p_1, support : 1), (p_3 \rightarrow p_1, support : 1)\}$
- From (Patient ID, 3): $\{(p_4 \rightarrow p_3, support : 1), (p_4 \rightarrow p_0, support : 1), (p_3 \rightarrow p_3, support : 1), (p_3 \rightarrow p_0, support : 1)\}$

- From (Patient ID, 4): $\{(p_1 \rightarrow p_2, support : 2), (p_1 \rightarrow p_5, support : 2), (p_1 \rightarrow p_1, support : 1), (p_2 \rightarrow p_1, support : 1), (p_2 \rightarrow p_2, support : 1), (p_2 \rightarrow p_5, support : 1), (p_5 \rightarrow p_1, support : 1), (p_5 \rightarrow p_2, support : 1), (p_5 \rightarrow p_5, support : 1)\}$

Table 16 shows such matrix for our diagnosis attributes:

Table 16: Matrix of atomic action sets for the *procedure* attribute

	p_0	p_1	p_2	p_3	p_4	p_5
p_0	0	0	0	1	0	0
p_1	0	2	2	1	1	2
p_2	0	3	1	1	0	1
p_3	1	1	0	1	0	0
p_4	1	0	0	1	0	0
p_5	0	1	1	0	0	1

When a new patient (or visit) enters the system, we use the atomic action rules extracted (from the diagnosis attributes and procedure attributes) to extract complete action rules. Let us assume that a patient comes in for the second visit, and we observe his/her states (or diagnoses) and from that information, we would like to decide on which procedure(s) to perform to improve his/her condition; using our system, the doctor will be able to anticipate the outcome for any specified operation that he/she can think of, to make sure it matches his/her desired outcome. For example, if the following is the information about the new patient:

Table 17: New patient data

	Patient ID	*diagnosis1*	*diagnosis2*	*diagnosis3*	*procedure1*	*procedure2*
x_0	5	d_0	d_3	d_2	p_1	p_0
x_1	5	d_5	d_6	d_4	?	?

The doctor may want to check what the condition of the patient would be after applying p_2 for example. We start by checking whether there exist (in our learned

system) atomic action sets that transform from last operation to p_2; the current patient's last operations were p_0 and p_1, by examining Table 16, we can observe that $(p_0 \rightarrow p_2)$ does not exist; however, $(p_1 \rightarrow p_2)$ does, so we further examine the instances in which the atomic action set $(p_1 \rightarrow p_2)$ was satisfied; which would be for the left hand side: $\{x_7, x_9\}$, and for the right hand side: $\{x_8, x_{10}\}$; next, we check the atomic action sets for each of the diagnoses to try and find transitions that occurred within the same instances. Let us check one diagnosis at a time, first we start with d_5 - this is the value for *diagnosis1* for the current patient (Patient ID, 5) - by looking at the row d_5 in Table 15:

- $(d_5 \rightarrow d_1)$; left hand side: $\{x_2\}$, right hand side: $\{x_3, x_4\}$;

  $\{x_2\} \cap \{x_7, x_9\} = \phi$ and $\{x_3, x_4\} \cap \{x_8, x_{10}\} = \phi$

- $(d_5 \rightarrow d_2)$; left hand side: $\{x_2\}$, right hand side: $\{x_3\}$;

  $\{x_2\} \cap \{x_7, x_9\} = \phi$ and $\{x_3\} \cap \{x_8, x_{10}\} = \phi$

- $(d_5 \rightarrow d_3)$; left hand side: $\{x_2\}$, right hand side: $\{x_4\}$;

  $\{x_2\} \cap \{x_7, x_9\} = \phi$ and $\{x_4\} \cap \{x_8, x_{10}\} = \phi$

- $(d_5 \rightarrow d_4)$; left hand side: $\{x_2\}$, right hand side: $\{x_4\}$;

  $\{x_2\} \cap \{x_7, x_9\} = \phi$ and $\{x_4\} \cap \{x_8, x_{10}\} = \phi$

Note that none of the transitions from d_5 resulted in a nonempty set when intersected with the sets that transition the specified procedure. We now check the second diagnosis value d_6; by looking at d_6 row in Table 15, we can observe that d_6 does not

transition to any other state, so we discard it. Next we look at the third diagnosis value d_4; by looking at d_4 row in Table 15, we observe the following transitions:

- $(d_4 \rightarrow d_3)$; left hand side: $\{x_9\}$, right hand side: $\{x_{10}\}$;

  $\{x_9\} \cap \{x_7, x_9\} = \{x_9\}$ and $\{x_{10}\} \cap \{x_8, x_{10}\} = \{x_{10}\}$ ✓

- $(d_4 \rightarrow d_4)$; left hand side: $\{x_9\}$, right hand side: $\{x_{10}\}$;

  $\{x_9\} \cap \{x_7, x_9\} = \{x_9\}$ and $\{x_{10}\} \cap \{x_8, x_{10}\} = \{x_{10}\}$ ✓

Since the two transitions: $(d_4 \rightarrow d_3)$ and $(d_4 \rightarrow d_4)$ are both valid, and since there are no more diagnosis transitions that are valid, we can construct the two following action rules:

$$(p_1 \rightarrow p_2) \Rightarrow (d_4 \rightarrow d_3)$$

$$(p_1 \rightarrow p_2) \Rightarrow (d_4 \rightarrow d_4)$$

The above two rules mean that by performing the procedure p_2 on the new patient, we are likely to observe a new condition (or diagnosis): d_3, and the condition (or diagnosis): d_4 is likely to stay. Next, we further calculate the confidence and support for that action rule. Note here that this approach will not affect the way we calculate the confidence and support, it will only extract action rules (in real-time) that are relevant, without extracting all action rules; and this is the main contribution here.

CHAPTER 6: OBJECT-DRIVEN ACTION REDUCTS

The concluding objective of both action reducts and action rules is the same; that is, to provide actionable tasks that specify necessary changes (actions) that will yield to desired transitions in an information system. In spite of their similarities, action reducts and classical action rules substantially differ in their properties, and in some situations, they tackle entirely different challenges. The extraction methodology for each approach is fundamentally distinct from the other, and therefore will result in virtually distinct actionable patterns. In this section, we provide a complete description of action reducts, how they differ from action rules, and how they are extracted from information systems. After that we introduce *Object-driven Action Reducts*; a new object-driven approach for extracting action reducts from information systems. We apply both the generalized action reducts, and the specialized object-driven action reducts to our hypernasality dataset and present some of the resulting action reducts extracted.

6.1 Reducts

The idea of exploring relations between attributes is the heart and soul of data mining and pattern extraction. In a certain sense, data mining could be regarded as a collection of algorithms for extracting relations between attributes, with distinct properties allocated to them.

Reducts answer one vital question about the relations of attributes; that is, can we reduce the total number of attributes used, by disregarding some, without losing any valued information; e.g. potential actionable patterns. Our goal from action rules extraction is to extract the most knowledge that can be extracted, in the form of actionable patterns, from any information system. In many situations, action rules extraction methodologies result in redundant action rules. Hence, the concept of representative action rules was proposed in [23] (explained in Section 2.3) to refer to the subset containing all essential action rules that can be later extended if needed. Extracting action rules from a compact and concise, but complete, system will generally result in overall rules with less redundant attributes; rules which are better to understand, better to extract, and better to apply. In other words, reducts seek to find a subset of the original set of attributes that we can use and still preserve all information needed to extract the same representative action rules.

Reducts have been used extensively in the field of rough set theory as a technique for information system reduction in general [17, 18], and as a tool to be used in extracting action reducts in particular [16]. In this section we provide a detailed explanation of two different variations of reducts, the first, formerly introduced one, is a reduct of the whole set [17, 18, 19], and the second, more recent variation, is the one proposed in [16] which defines α and β reducts.

To establish common background, we start by providing preliminary definitions. A classification, loosely speaking, is an operation that partitions the set of elements in the universe U into subsets that share similar characteristics. Every element will be

Table 18: Information system S_9

	a	b	c	d
x_1	a_1	b_2	c_2	d_1
x_2	a_2	b_2	c_1	d_2
x_3	a_2	b_1	c_1	d_2

classified into one and only one subset (or class). Meaning that a classification will partition our information system into a set of all subsets $C = \{X_1, X_2, \ldots, X_n\}$ such that $X_i \subseteq U, X_i \neq \phi, X_i \cap X_j = \phi$ for $i \neq j,\ i, j = 1, \ldots n$ and $\bigcup X_i = U$ [17].

According to the classification, also referred to as the equivalence relation, elements in any subset X_i are indiscernible (or indistinguishable) from each other. To denote a classification applied to a set, we use the notation U/P, where U is the universe and P is an equivalence relation.

For example, referring to information system S_9 in Table 11,

$$U/P = \{\{x_1\}, \{x_2, x_3\}\}, \text{where } P = \{c\},$$

which means that by using attribute c, our information system is partitioned into the two following subsets: $\{x_1\}$ and $\{x_2, x_3\}$.

The following are few more examples, using S_9 as our information system:

$$U/P = \{\{x_1\}, \{x_2, x_3\}\}, \text{where } P = \{a\},$$

$$U/P = \{\{x_1, x_2\}, \{x_3\}\}, \text{where } P = \{b\},$$

$$U/P = \{\{x_1\}, \{x_2\}, \{x_3\}\}, \text{where } P = \{a, b\}.$$

Note that in the last example, the equivalence relation defined by P is the in-

tersection of the two single-attribute equivalence relations. In other words, for an observation x_1 to be in the same equivalence class as x_2, it has to match both attributes a and b; which is clearly not the case.

A reduct $P \subseteq A$ in an information system $S = (A, V)$ must satisfy the two following conditions:

1. $U/P = U/A$

2. P is a minimal set of attributes with property 1.

The first property guarantees that no loss of valued information has occurred; it means that using the set of attributes defined by the reduct P should not be any less discernible than using the set of all attributes A. Clearly it cannot be more discernible, as $P \subseteq A$, where A is the set of all attributes. For example, referring to information system S_1, we can remark that by using the set of attributes $\{a, b, c\}$ as our equivalence relation, we would be able to discern every instance from every other instance; $U/\{a, b, c\} = \{\{x_1\}, \{x_2\}, \{x_3\}\}$. However, it can also be remarked that {a, b, c} is not a minimal set, which means that the set {a, b, c} cannot be considered a reduct. By doing further more analysis, it can be observed that the reducts in S_1 are the following: $\{a, b\}, \{b, c\}$, and $\{b, d\}$. Note that it is not uncommon to have multiple reducts. In this work, we will not explore the different ways for extracting reducts from an information system.

Next, we shift our discussion to the more recent variation of reducts proposed by Im et al. in [16]. The core difference between the newly introduced approach in [16], and the former classical reducts approach is essentially in the discernibility domain.

Table 19: X_α for information system S_5

	ObjectID	*Income*	*Number of Children*	*Loyalty*
x_3	1	*High*	More than 3	*High*
x_4	1	*High*	More than 3	*High*
x_7	2	*Medium*	Less than or equal to 3	*High*
x_8	2	*Medium*	Less than or equal to 3	*High*

In the classical reducts, we aim to preserve the same discernibility capacity for every instance against every other instance. On the other hand, in the newly introduced reducts, we aim to preserve the same discernibility capacity for the set of desired instances against the set of undesired instances. As we will see in following sections, this will be of essential importance, and great use, when extracting action reducts.

We start by partitioning our information system into two separate tables. The first one, which is called X_α, will contain all desired instances; desired instances will be determined by one (or more) predefined desired state(s) of the decision attribute. The second table, which is called X_β, will contain undesired instances; similarly, undesired instances will be determined by one (or more) predefined undesired state(s) for our same decision attribute.

For example, by partitioning the information system S_5 shown in Table 8 (add page) into the desired set $X_\alpha = \{x_i \in X : Loyalty(x_i) \in \{High\}\}$, and the undesired set $X_\beta = \{x_j \in X : Loyalty(x_j) \in \{Low\}\}$, we would get as a result the two subsystems shown in Table 19 and Table 20, respectively.

The goal of action reducts explained in [16] is to find a minimal set of attributes states that exists exclusively in our desired table X_α. For example, referring to information system S_5, we can observe from Table 19 and Table 20 that the state

Table 20: X_β for information system S_5

	ObjectID	*Income*	*Number of Children*	*Loyalty*
x_1	1	*Medium*	More than 3	*Low*
x_2	1	*Medium*	More than 3	*Low*
x_5	2	*Low*	Less than or equal to 3	*Low*
x_6	2	*Low*	Less than or equal to 3	*Low*

Income(High) exists exclusively in X_α, also the two states combined *Income(Medium)* and *NumberOfChildren(Less than or equal to 3)* also exist (exclusively) in X_α; as a result, both the single-attribute state, and the double-state attributes are reducts. Note that although the two states *Income(High)* and *NumberOfChildren(More than 3)* are exclusive to X_α, they cannot be considered a reduct since they are not minimal. It is also noteworthy to remark that this variation of reducts will result in attribute states results, instead of the first reduct variation, which would result in attributes reducts.

In the next two sections, we provide two different approaches in which action reducts can be extracted. The first one, being the general approach, is the one on which we make the assumption that all instances come from the same distribution, and therefore action reducts are extracted from the overall system at once. It is important to understand that when extracting action reducts (or action rules) from a set of instances, all instances need to come from the same distribution. In theory, if our information system contains every possible attribute for all instances, instances from different distributions would be distinguishable by their attribute values. However, it is practically impossibly to observe the complete characteristic of every instance.

Although generalization is necessary for real-world applications, we believe that

limiting the degree of generalization has major benefits in some situations. It is often the case that our information system instances come from intrinsically many unique distributions; having said that, instances that belong to one cluster of some distribution do not necessarily need to share similar characteristics, however similar behavior, which makes this process rather non-trivial.

6.2 (Generalized) Action Reducts

In generalized action reducts, we assume that all instances are observed from the same distribution; hence, action rules are extracted from the entire information system at once. We start by building the discernible attribute values for X_α against X_β [16], as shown in Table 21.

Table 21: Discernible attribute values for X_α against X_β

	x_3	x_4	x_7	x_8
x_1	*High Income*	*High Income*	$N\_of\_C \leq 3$	$N\_of\_C \leq 3$
x_2	*High Income*	*High Income*	$N\_of\_C \leq 3$	$N\_of\_C \leq 3$
x_5	*High Income* + $N\_of\_C \geq 3$	*High Income* + $N\_of\_C \geq 3$	*Medium Income*	*Medium Income*
x_6	*High Income* + $N\_of\_C \geq 3$	*High Income* + $N\_of\_C \geq 3$	*Medium Income*	*Medium Income*

Each column shows the necessary attribute states that need to be satisfied, in order for the undesired instance to become desired in a way that matches that corresponding column. We assume here that *N_of_C* means "numbers of children". For example, the first column states one way to shift clients' *loyalty* from low to high; by changing the characteristics of elements in X_β to match those of $x_3 \in X_\alpha$. The conjunction of the states in the first column indicates the required change:

$(High\ Income) \wedge (High\ Income \vee N\_of\_C \geq 3)$.

Note that the required change is described in the form of conjunctions. To extract α-reducts, we need to convert the conjunction normal form (CNF) to a disjunction normal form (DNF); by finding prime implicants. Referring to the same example above (using first column), the result of converting the CNF to DNF would yield the following: $(High\ Income) \vee (High\ Income \wedge N\_of\_C \geq 3)$. Since α-reducts need to be minimal, we dismiss the second term of our previous disjunction. Here, we show a list of all α-reducts acquired from Table 21:

- α-reducts$(x_1) = \{(High\ Income)\}$
- α-reducts$(x_2) = \{(High\ Income)\}$
- α-reducts$(x_3) = \{(Medium\ Income \wedge N_of_C \leq 3)\}$
- α-reducts$(x_4) = \{(Medium\ Income \wedge N_of_C \leq 3)\}$

Each resulting α-reduct, also known as action reduct, can be interpreted as an actionable rule. For example, the α-reduct $(Medium\ Income \wedge N\_of\_C \leq 3)$ can be interpreted as follows: by shifting the *income* of employees who have number of children less than or equal to three from *any state* to *medium*, we would improve their *loyalty* from *low* to *high*. Note that this form of actionable tasks does not specify the source state (state that we should shift from). Classical action rules on the other hand specifies both the source state, and the destination state.

To define action reducts more precisely, three characteristics were introduced in [16]; *frequency*, *hit ratio*, and *weight*. The *frequency* of an α-reduct is the number

of times it appears in X_α. The *hit ratio* on the other hand, reflects the percentage of instances in X_β that can be shifted to an α-reduct; note that although we do not specify the source of the actionable task, the fact that some attributes are stable will limit our source domain. Finally, the *weight* is the normalized value of the frequency multiplied by the hit ratio; by normalizing, we set the sum of weights for all action reducts to be one.

For example, the α-reduct ($Medium\ Income \wedge N\_of\_C \leq 3$) has frequency 2; since it appears two times in X_α, has hit ratio 2; since two instances in X_β can be shifted to it, and has weight of 4/8. Table 22 shows the characteristics of the two action reducts.

Table 22: α-reduct for information system S_5

α-reduct	Frequency	Hit Ratio	Weight
($High\ Income$)	2	1	4/8 (50%)
($Medium\ Income \wedge \leq 3$)	2	.5	4/8 (50%)

6.3 Results of Applying Generalized Action Reducts to Hypernasality

Action Reduct 1. $r_1 = (yeaou, < 2.5) \wedge (i - long, < 1.5)$; Cut configuration: desired:{0, .5, 1, 1.5}; undesired:{2, 2.5, 3}; frequency: 133; hit ratio: 1; weight: .029

This action reduct states that by decreasing the levels of nasalization for vowels /I, e, a, o, u, i/ to less than 2.5, and by decreasing /i/-long to less than 1.5, we are expected to move patients who had one of the following undesired states: {2, 2.5, 3} to one of the following desired states {0, .5, 1, 1.5}. Note that although the weight

appears to be low, this is not due to any weakness of the action reduct, but instead due to the many (341) action reducts generated; in fact, this is the strongest action reduct amongs all 341. Recall that the *weights* of all action reducts are normalized so that all weights sum up to 1.

Action Reduct 2. $r_2 = (diagnosis, III) \wedge (yeaou, < 2.5)$; Cut configuration: desired:{0, .5, 1}; undesired:{1.5, 2, 2.5, 3}; frequency: 191; hit ratio: 1; weight: .035

Although the number of total action reducts generated by this cut configuration is higher than the previous action reduct cut by 30, the weight of this action reduct is still higher than of r_1. This action reduct states that for patients who are having hypertrophy of adenoids (and possibly palatine tonsils), by decreasing their levels of nasalization for vowels /I, e, a, o, u, i/ to less than 2.5, they are expected to improve from one of the following Czermak's mirror tests' undesired states: {1.5, 2, 2.5, 3} to one of the following desired states {0, .5, 1}.

Action Reduct 3. $r_3 = (diagnosis, III) \wedge (i - long, < 1.5)$; Cut configuration: desired:{0, .5}; undesired:{1, 1.5, 2, 2.5, 3}; frequency: 118; hit ratio: 1; weight: .023

This action reduct states that for patients who are having hypertrophy of adenoids (and possibly palatine tonsils), by decreasing their levels of /i/-long, they would be expected to improve from one of the following Czermak's mirror tests' undesired states: {1, 1.5, 2, 2.5, 3} to one of the following desired states {0, .5}.

6.4 Specialized Action Reducts

In this section, we will discuss a less general approach in which action reducts can be extracted. We believe that this specialized action reduct extraction methodology

will be a fit candidate in situations where information systems have intrinsic nature of multi-distribution.

The main difference between the specialized approach and the previously explained general approach is in the way we build the discernible matrix. As seen before, when we build our discernible matrix for the general approach, we are concerned with the discernible attribute values for the entire X_α agains the entire X_β. However, when constructing the discernible matrix for the specialized case, we are concerned with the discernible attribute values for subsystems of X_α and X_β; the subsystems are defined by an object attribute. In Table 23, we show the discernible matrix using the specialized (object-driven) case; *N/A* indicates an intersection of two instances with different *UserID* values, therefore discernibility for the two instances should not be calculated.

Table 23: Specialized discernible attribute values for X_α against X_β

	x_3	x_4	x_7	x_8
x_1	*High Income*	*High Income*	*N/A*	*N/A*
x_2	*High Income*	*High Income*	*N/A*	*N/A*
x_5	*N/A*	*N/A*	*Medium Income*	*Medium Income*
x_6	*N/A*	*N/A*	*Medium Income*	*Medium Income*

The way we calculate the frequency, hit ratio, and weight for each subsystem is similar to the general approach. However, we aggregate the values for same α-reducts from multiple subsystems. In our example, the two different subsystems do not share any α-reducts, therefore no aggregation will take place. In addition, we introduce a fourth property for the specialized version of α-reducts; namely, the *accuracy*. The accuracy of an α-reduct extracted from a subsystem will indicate whether there exist

Table 24: Specialized α-reduct for information system S_5

α-reduct	Frequency	Hit Ratio	Weight	Accuracy
(*High Income*)	2	1	4/8 (50%)	2/2 (100%)
(*Medium Income*)	2	.5	4/8 (50%)	2/4 (50%)

any contradictions with other subsystems or not. We define the accuracy of an α-reduct as the sum of its occurrences in both X_α and X_β for all subsystems, divided by its frequency. Table 24 shows the characteristics of the specialized α-reducts.

Note that the second α-reduct also exists in X_β for the second subsystem (userID: 2), which ultimately affects the accuracy rather unfavorably. In a certain sense, the specialized action reducts and the generalized action reducts provide system users with different outcome. The outcome of the generalized action reducts extraction method is absolute, in a sense that 100% of all α-reducts that are extracted are exclusively observed in X_α (in association with the desired state) and not even once in X_β. The outcome of the specialized action reducts on the other hand is not absolute for the entire system; the accuracy of each action reduct is an indicator of its absoluteness.

Example of specialized action reducts: here, we provide a complete example to demonstrate how to extract specialized action reducts from a sample information system. Table 25 shows Information System S_{10}, which consists of elements observed from three different users; denoted by the object attribute *UserID*. From Table 25, we build the two matrices X_α and X_β shown in Table 26 and Table 27 respectively; where our decision attribute is d, and our desired state is d_2. Next, the object-driven specialized discernibly matrix shown in Table 28 is constructed. Note that it would

Table 25: Information system S_{10}

	UserID	a	b	c	d
x_1	1	a_1	b_1	c_1	d_1
x_2	1	a_1	b_1	c_2	d_1
x_3	1	a_2	b_2	c_2	d_2
x_4	1	a_1	b_2	c_1	d_1
x_5	1	a_1	b_1	c_1	d_2
x_6	2	a_1	b_1	c_1	d_2
x_7	2	a_1	b_1	c_2	d_1
x_8	3	a_2	b_2	c_2	d_1
x_9	3	a_2	b_2	c_2	d_1
x_{10}	3	a_1	b_2	c_2	d_2
x_{11}	3	a_1	b_2	c_2	d_2

Table 26: X_α for information system S_{10}

	UserID	a	b	c	d
x_3	1	a_2	b_2	c_2	d_2
x_5	1	a_1	b_1	c_1	d_2
x_6	2	a_1	b_1	c_1	d_2
x_{10}	3	a_1	b_2	c_2	d_2
x_{11}	3	a_1	b_2	c_2	d_2

invalid to extract discernibility value(s) between two instances from different objects; hence, we denote such case with the value *N/A*.

The first column in Table 12 indicates three distinct units of information; firstly, it indicates that the desired instance x_3 is different from the undesired instance x_1 by the three states a_2, b_2, and c_2; secondly, it indicates that x_3 is different from x_2 by the two states a_2 and b_2; and thirdly, it indicates that x_3 is different from x_4 by the two states a_2 and c_2. This means that for the instance x_3 to be distinct from all other undesired states, it should satisfy at least one state when compared to each other undesired state. The process is known as finding prime implicants; and it is

Table 27: X_β for information system S_{10}

	UserID	a	b	c	d
x_1	1	a_1	b_1	c_1	d_1
x_2	1	a_1	b_1	c_2	d_1
x_4	1	a_1	b_2	c_1	d_1
x_7	2	a_1	b_1	c_2	d_1
x_8	3	a_2	b_2	c_2	d_1
x_9	3	a_2	b_2	c_2	d_1

Table 28: Discernable attribute values for information system S_{10}

	x_3	x_5	x_6	x_{10}	x_{11}
x_1	$a_2 + b_2 + c_2$	ϕ	N/A	N/A	N/A
x_2	$a_2 + b_2$	c_1	N/A	N/A	N/A
x_4	$a_2 + c_2$	b_1	N/A	N/A	N/A
x_7	N/A	N/A	c_1	N/A	N/A
x_8	N/A	N/A	N/A	a_1	a_1
x_9	N/A	N/A	N/A	a_1	a_1

done by converting the conjunction normal form (CNF) to disjunction normal form (DNF). The conjunction normal form (CNF) for the first column is the following:

$$((a, a_2) \vee (b, b_2) \vee (c, c_2)) \wedge ((a, a_2) \vee (b, b_2)) \wedge ((a, a_2) \vee (c, c_2))$$

Which, when converted to the disjunction normal form (DNF), would become:

$$((a, a_2) \wedge (b, b_2) \wedge (c, c_2)) \vee ((a, a_2) \wedge (b, b_2)) \vee ((a, a_2) \wedge (c, c_2)) \vee ((b, b_2) \wedge (c, c_2)) \vee (a, a_2)$$

However, since we are only interested in reducts, we discard any set that contains an existing subset; and since (a, a_2) exists in three other components, and yet it is an element by itself, we will be left with only two sets:

$$\cancel{((a, a_2) \wedge (b, b_2) \wedge (c, c_2))} \vee \cancel{((a, a_2) \wedge (b, b_2))} \vee \cancel{((a, a_2) \wedge (c, c_2))} \vee ((b, b_2) \wedge (c, c_2)) \vee (a, a_2)$$

Following the same procedure, we would end up with the following specialized action reducts:

- α-reducts$(x_3) \ = \{((b, b_2) \wedge (c, c_2)), (a, a_2)\}$
- α-reducts$(x_5) \ = \{(b, b_1), (c, c_1)\}$
- α-reducts$(x_6) \ = \{(c, c_1)\}$
- α-reducts$(x_{10}) = \{(a, a_1)\}$
- α-reducts$(x_{11}) = \{(a, a_1)\}$

Note that two of the resulting reducts appear to be contradicting each other, namely (a, a_1) and (a, a_2), this is however not so uncommon in object-driven action reducts; since we are extracting action reducts from subsystems independently, conflict between subsystems is at times inevitable to occur. This conflict however, will not cause any confusion to which action reduct is to be chosen; the latterly defined characteristic of *accuracy* will help us disambiguate this, and any similar, case; by showing which action reduct has more strength; hence, the one to be chosen. Next, we will demonstrate how to calculate the characteristics of object-driven specialized action reducts for both (a, a_1) and (a, a_2).

The *frequency* of (a, a_1) is 4, since it appeared once to (userID, 1), once to (userID, 2), and twice to (userID, 3); the *frequency* of (a, a_2) on the other hand is 1, since it appeared only once to (userID, 1). The *hit ratio* is a rather interesting measure in this case; the reason goes back to our object-driven approach mentioned earlier, which states that we need to treat each subsystem as an entirely independent system;

this will mean that the *hit ratio* needs to be calculated independently from each subsystem. Accordingly, the *hit ratio* of (a, a_1) will be equal to 0/3 (userID, 1) + 0/1 (userID, 2) + 2/2 (userID, 3), which evaluates to 1; this value is further normalized according to the weight of each object; the *hit ratio* for (a, a_2) on the other hand will be equal to 3/3 (userID, 1) + 1/1 (userID, 2) + 0/2 (userID, 3), which evaluates to 2. Similarly, the *weight* needs to be calculated for each subsystem independently; (a, a_1) will have *weight* $1 * 2$ (userID: 3), which evaluates to 2; consequently, (a, a_2) will have *weight* $1 * 1$ (userID: 1), which would evaluate to 1; and again, all weights need to be normalized. Finally, we calculate the *accuracy* value for both action reducts; the *accuracy* for (a, a_1) is 4/8; and the *accuracy* for (a, a_2) is 1/3. Recall that the *accuracy* of an action reduct is the sum of its occurrences in both X_α and X_β for all subsystems, divided by its frequency. Next, we show the list of characteristics for all specialized action reducts:

Table 29: Specialized α-reduct for information system S_{10}

α-reduct	Frequency	Hit Ratio	Weight	Accuracy
$\{(b, b_2) \wedge (c, c_2)\}$	3	1	.23	.6
$\{(a, a_2)\}$	1	1	.23	.33
$\{(b, b_1)\}$	2	.25	.08	.4
$\{(a, a_1)\}$	4	.5	.46	.5

6.5 Results of Applying Specialized Action Reducts to Hypernasality

Action Reduct 1. $r_1 = (motility, [4.5, 5.5])$; Cut configuration: desired:{0, .5, 1, 1.5}; undesired:{2, 2.5, 3}; frequency: 4; hit ratio: 1; weight: .118; accuracy: .94

Note that the weight of this action reduct is much higher than any of the generalized action reducts, this is due the low number of action reducts generated in the object-driven specialized approach. This action reduct states that by transitioning the motility of the soft palate for patients to [4.5, 5.5], we are expected to observe a condition change of patients who had one of the following undesired states: {2, 2.5, 3} to one of the following desired states {0, .5, 1, 1.5}

Action Reduct 2. $r_2 = (i - long, [2.5, 7.5)) \wedge (difference\ level\ F1\text{-}F2, < 4.5)$; Cut configuration: desired:{0, .5, 1}; undesired:{1.5, 2, 2.5, 3}; frequency: 1; hit ratio: 1; weight: .028; accuracy: .87

This rule states that by transitioning the state of patients' levels of /i/-long to [2.5, 7.5], and by decreasing the difference between the first two formants of the vocal tract for /i/-long, we would expect to improve patients' condition from one of the following undesired states: {1.5, 2, 2.5, 3} to one of the following desired states {0, .5, 1}.

Action Reduct 3. $r_3 = (bdg, [6.5, 8.6))$; Cut configuration: desired:{0}; undesired:{.5, 1, 1.5, 2, 2.5, 3}; frequency: 9; hit ratio: 1; weight: .042; accuracy: .90

This action reduct is rather simple and straight forward, but its high frequency (compared to other object-driven specialized action reducts) makes it an interesting and valuable pattern. It states that by shifting the patients' nasality of /bdg/ to [6.5, 8.6), we would expect for their Czermak's mirror test to transition from one of the following undesired states: {.5, 1, 1.5, 2, 2.5, 3} to {0}; which would be the absolute

best condition the patient may transition to.

CHAPTER 7: HIERARCHICAL OBJECT-DRIVE ACTION RULES

7.1 Motivation: From Concepts to Applications

So far, we have discussed in details the concept of object-driven and temporal action rules. In summary, we approached the problem of action rule extraction by treating our entire information system as a collection of multiple subsystems that, assumingly, come from the same distribution. This approach is highly encouraged when dealing with temporal information systems, and when mixing different distributions of observations might lead to unrepresentative action rules; though by treating our information system as one, we might get more action rules, the actual accuracy and intrinsic confidence might not indeed reflect the actual state of reality. For that reason, we decided to re-examine our information system, take a step back, and apply action rule extraction to the level of objects; while all details were explained in previous sections. Our results have been successfully examined (and tested) by physicians in the field, so we are confident to say that the object-driven and temporal based approach presented in this thesis was a successful attempt for action rule extraction.

Having said that, it is important to note that although we are extracting more accurate action rules, some argue that this approach; as it is right now, is imposing few limitations. Needless to say, the process of specializing applied to our entire information system, hence restructuring it into multiple independent sub-systems, is

preventing us from extracting rules from different patients when combined together; by doing so, it may seem that we are being overly cautious. To that end, we propose a hybrid approach in which we combine the more accurate but less vast approach of object-driven and temporal action rules, with the less accurate but more vast approach of the classical action rule approach.

In Chapter 4, we explained the motivation of object-driven and temporal action rule extraction, and that we only extract (or learn) action rules from events that in fact actually happened in our information system, this is the essence and motive that led us to limit the direction of action rule extraction to comply with the temporal aspect.

In this Chapter, we introduce a systematic approach to calculate a hierarchical similarity metric between objects in information systems. This will allow us to group objects that have high similarities; consequently, making it feasible to further extract action rules from objects in clusters rather than in isolation, which would in turn result in veritably stronger and more diverse collection of action rules. One of the fundamental motives for object-driven action rules however, is to avoid cross-object learning [3], [4]; by cross-object learning, we mean extracting action rules from instances that come from different distributions. For that reason, our clustering approach introduced in this work must comply with that principle as well; in other words, objects that are similar to each other (according to our similarity metric that will be introduced in this Chapter) are fundamentally objects that come from similar distributions. Having said that, it is worth stating here that action rules essentially describe some behaviors in term of others; which makes *object behavior* an intrinsi-

cally good choice to be used to measure similarities between objects; more details about the clustering phase is discussed in following sections. Our choice of hierarchical clustering is strictly motivated by the flexibility it provides to system users; as will be shown in later sections, the level of hierarchy chosen by system users determines the degree of specialization (or generalization) prior to action rule extraction; for example, in certain domains such as healthcare, it is always necessary to be entirely confident of the outcome of actions undertaken to patients, which would mean that action rule generalization needs to be done attentively; although by generalizing rules, we tend to cover a wider group of instances (typically patients in this case), the fact that some instances will not behave according to the action rule will make generalization an improper action. Note that this is not the case in other domains; for example, in marketing, we may be interested in sending letters to potential clients to buy a particular product; by generalizing our action rules, we are more likely to target a larger audience, however, the percentage of this larger audience may decrease, though it would still be more profitable to do so.

7.2 Clustering by Object Behaviour

In Chapter 5, we discussed the advantages of limiting action rule extraction to the level of objects. The motivation was to extract more accurate action rules that better reflect real world cases, and to provide a more accurate actual representation of the observed information system. However, we also mentioned that specialization could cause over-fitting problems, and for that reason, it is the case that in some situations, generalization may be preferred; mainly when our number of observations

within unique objects is limited. In this section, we provide a hybrid approach that combines both the specialized object-driven approach, and the general classical action rules approach. The hierarchy that we build will provide system users with tremendous convenience; allowing decision makers to take decision on how much generalization/specialization to be applied, by simply examining the dendrogram and specifying the proper level of specialization.

While there are many advantages for having a pure specialized object-driven approach to extract action rules, it is clear that by having more objects in our system, the number of extracted action rules, and their corresponding supports will automatically be negatively affected. Following the non-temporal pair-based action rule extraction (Subsection 5.3.3); for an object that consists of n number of instances, the maximum support (number of supporting pairs that could exist) is $(n/2)^2$, and that is only the case when half of our instances satisfy the preconditions of the action rule, and the other half satisfy the postcondition of the action rule; it would become apparent then, that by dividing our information system to multiple subsystems, the total support for our action rules will eventually decrease. For example, if we extract action rules from a system of 100 instances, the maximum support for a particular action rule would be 50^2, which is 2500; however, if we divide our 100-instance system into ten 10-element subsystems, the maximum support for a particular action rule would be $10 * 5^2$, which is 250, only one tenth the possible action rules that could be extracted by not using the object-driven approach; more accurately, the maximum support for a potential action rule is inversely proportional to the number of objects

(or clusters) in our information system.

Table 30 shows the number of action rules, and the total support for our hypernasality disorder dataset described in Chapter 3. The first column in Table 30 denotes the decision attribute of the action rules extracted from our system; the second column shows the total number of action rules extracted that trigger the corresponding decision shift from the first column, note that in the second column, we are treating each subsystem, identified by the object attribute, as an independent system. The third column on the other hand shows the number of action rules extracted, again, with respect to the corresponding decision shift from the first column, but after clustering objects (or patients), forming 40 clusters; before clustering patients together, we are essentially extracting action rules from each of the 225 objects (or patients) independently; however, when we cluster objects together, we are combining objects (or patients), hence resulting in less subsystems, but with more instances in each subsystem (or cluster). Necessary details about our information system will be provided in following sections; however, we can still observe that when more objects (in this case patients) are combined, forming less independent subsystems, the more unique action rules get extracted. Combining patients (or objects) to form bigger groups of patients occurs through a clustering procedure where a similarity metric is proposed, as will be seen next. Also, the fourth column shows values for the total support for all action rules instead of the actual number of action rules; similarly, we can observe a pattern of changes, in which the more objects we combine, the higher support for action rules we get.

To address the limitations that may occur when extracting action rules from pure

Table 30: Total number of action rules with respect to number of clusters

	Number of Action Rules		Total Support	
decision shift	*object-driven*	*40 clusters*	*object-driven*	*40 clusters*
$(2.5 \rightarrow 2)$	1608	35937	4456	354332
$(2.5 \rightarrow 1)$	700	31467	1580	304808
$(2 \rightarrow 1.5)$	112	12972	224	81072
$(2 \rightarrow 1)$	480	25728	960	223753
$(2 \rightarrow 0)$	14	91133	28	931985
$(1.5 \rightarrow 1)$	388	10054	776	66874
$(1 \rightarrow 0.5)$	96	59927	200	497465
$(1 \rightarrow 0)$	954	85769	1996	755361

object-driven approach, we present a new hierarchical approach in which we cluster objects that react similarly to particular treatments using the minimum (or single-linkage) clustering criterion; this approach, which we will refer to as the hybrid hierarchical approach, can be regarded as the product of combining both the classical action rule extraction approach and the object-driven approach. The goal thus become to examine objects and find groups of objects that we would expect to react similarly when a set of actions were to be performed; in other words, react similarly to particular action rules. Our assumption that we base this work upon is that objects that share high similarities with respect to actions, tend to share same reactions with other unexamined actions. For example, if two patients react similarly to ten shared treatments, then one would be relatively confident that they share similar responses, hence are more likely to respond similarly to other actions as well; in this section, we provide a detailed explanation of the hybrid hierarchical action rule extraction approach. The idea proposed here is to use our existing action rule extraction system to learn similarities in behavior found in objects (or patients). As mentioned earlier, the goal and outcome of action rule extraction is to bring forth actionable patterns

that describe reactions that will occur as a result of other performed actions; in other words, by performing some actions, others (in which we do not have direct control over) will be triggered as a result. Typically, we would be only interested in specific shifts; for example, in our hypernasality treatment case study, we were interested in transitions that shift the state of patient from more severe to less severe. It would be an act of bias however, if we cluster patients by only measuring similarities with respect to one direction of shift in our decision attribute. For that reason, to measure similarities, hence clusters, using an unbiased behavioral approach, it would be only fair to measure similarities in behavior in all directions. Next we provide steps to accomplish that.

Note that when objects are combined together forming clusters, it would be purposeless to consider the temporal aspect in further computations. The reason for that goes back to the assumption that objects are independent entities; the fact that one instance from one object occurred before another instance from another object has no significant meaning whatsoever; for example, by knowing that one patient was given a particular drug before another patient went into surgery will not help us to make a better prediction of the surgery outcome, which is clearly not the case when dealing with one entity (or patient).

1. The first step would be to identify all possible transitions in our decision attribute. For example, if our decision attribute was d, where its possible values/states are listed in the following set: $\{d_1,\ d_2,\ d_3\}$, then the following set of atomic actions shows all possible transitions: $\{(d, d_1 \rightarrow d_2), (d, d_1 \rightarrow$

$d_3), (d, d_2 \rightarrow d_1), (d, d_2 \rightarrow d_3), (d, d_3 \rightarrow d_1), (d, d_3 \rightarrow d_2)\}$

2. The second step would be to extract all action rules with respect to the transitions extracted in the first step; from every subsystem. We keep track of what subsystems support an action rule and what subsystems do not. In addition to that, we keep track of support and confidence for every action rule in every subsystem.

3. The third step would be to build a similarity matrix between every pair of objects, in which we measure the distance between the confidence and the relative support for every action rule in every subsystem. Since we are measuring similarities between objects that contain different number of observations; it is essential to use the relative support instead of the absolute support, defined as the support divided by the number of observation. The similarity between two objects o_1 and o_2 is defined as follows:

$$Similarity(o_1, o_2) = \sum_{r \in R} 2 - \ |conf(o_1) - conf(o_2)| + \left| \frac{supp(o_1)}{E(o_1)} - \frac{supp(o_2)}{E(o_2)} \right|$$

Where R contains the set of all action rules shared by both o_1 and o_2, and where $E(o_1)$ and $E(o_2)$ denote the number of observations in object o_1 and o_2 respectively. The division by the number of observation will result in the relative support. For each two objects, all action rules similarities are computed then further aggregated; if the confidence and relative support are equal for one particular action rule, then the similarity will be 2 (for only that action rule), which is essentially the maximum similarity value. Note that in this formulation,

we assume that the confidence and support have equal weights when calculating objects similarities; though we believe this is a fair assumption in our similarity formulation, adding a rate to set different weights is easily accomplished, and will not change the course of future computations.

4. Finally, we build a hierarchy of clusters using the minimum (or single-linkage) clustering criterion based on the similarity matrix built in step 3, hence producing a dendrogram.

Example to demonstrate the hybrid hierarchical object-driven approach: Here, we provide an almost complete example by following the previous four steps to extract action rules using the hybrid hierarchical object-driven approach.

Table 31: Information system S_{11}

	$objectID$	a	b	d
x_0	1	a_1	b_1	d_1
x_1	1	a_2	b_1	d_1
x_2	1	a_2	b_2	d_2
x_3	1	a_1	b_2	d_1
x_4	2	a_2	b_1	d_2
x_5	2	a_1	b_2	d_2
x_6	2	a_2	b_1	d_1
x_7	3	a_1	b_2	d_1
x_8	3	a_2	b_2	d_2
x_9	3	a_1	b_1	d_1
x_{10}	3	a_2	b_1	d_2
x_{11}	4	a_2	b_1	d_1
x_{12}	4	a_2	b_1	d_1
x_{13}	4	a_2	b_2	d_2
x_{14}	4	a_1	b_2	d_1
x_{15}	5	a_1	b_2	d_1
x_{16}	5	a_2	b_2	d_2
x_{17}	5	a_1	b_1	d_1
x_{18}	5	a_2	b_1	d_2

The first step would be to identify all possible transitions in our decision attribute. In this example, and since we only have two states for our decision attribute d, the two decision transitions are $(d, d_1 \rightarrow d_2)$ and $(d, d_2 \rightarrow d_1)$.

Next, we extract all action rules with respect to all decision transitions from every subsystem; by applying the non-temporal pair-based approach. For the sake of simplicity, let us assume that attribute b is a stable attribute. The output of this second step would be:

The following action rules are extracted from *Object ID 1*:

$(a, a_1 \rightarrow a_2) \Rightarrow (d, d_1 \rightarrow d_2)$; support 2; confidence 50%

$(a, a_2 \rightarrow a_1) \Rightarrow (d, d_2 \rightarrow d_1)$; support 2; confidence 100%

$(a, a_1 \rightarrow a_2) \wedge (b, b_2) \Rightarrow (d, d_1 \rightarrow d_2)$; support 1; confidence 100%

$(a, a_2 \rightarrow a_1) \wedge (b, b_2) \Rightarrow (d, d_2 \rightarrow d_1)$; support 1; confidence 100%

The following action rules are extracted from *Object ID 2*:

$(a, a_1 \rightarrow a_2) \Rightarrow (d, d_2 \rightarrow d_1)$; support 1; confidence 100%

$(a, a_2 \rightarrow a_1) \Rightarrow (d, d_1 \rightarrow d_2)$; support 1; confidence 100%

The following action rules are extracted from *Object ID 3*:

$(a, a_1 \rightarrow a_2) \Rightarrow (d, d_1 \rightarrow d_2)$; support 4; confidence 100%

$(a, a_2 \rightarrow a_1) \Rightarrow (d, d_2 \rightarrow d_1)$; support 4; confidence 100%

$(a, a_1 \rightarrow a_2) \wedge (b, b_1) \Rightarrow (d, d_1 \rightarrow d_2)$; support 1; confidence 100%

$(a, a_1 \rightarrow a_2) \wedge (b, b_2) \Rightarrow (d, d_1 \rightarrow d_2)$; support 1; confidence 100%

$(a, a_2 \rightarrow a_1) \wedge (b, b_1) \Rightarrow (d, d_2 \rightarrow d_1)$; support 1; confidence 100%

$(a, a_2 \rightarrow a_1) \wedge (b, b_2) \Rightarrow (d, d_2 \rightarrow d_1)$; support 1; confidence 100%

The following action rules are extracted from *Object ID 4*:

$(a, a_1 \rightarrow a_2) \Rightarrow (d, d_1 \rightarrow d_2)$; support 1; confidence 100%

$(a, a_2 \rightarrow a_1) \Rightarrow (d, d_2 \rightarrow d_1)$; support 1; confidence 100%

$(a, a_1 \rightarrow a_2) \wedge (b, b_2) \Rightarrow (d, d_1 \rightarrow d_2)$; support 1; confidence 100%

$(a, a_2 \rightarrow a_1) \wedge (b, b_2) \Rightarrow (d, d_2 \rightarrow d_1)$; support 1; confidence 100%

The following action rules are extracted from *Object ID 5*:

$(a, a_1 \rightarrow a_2) \Rightarrow (d, d_1 \rightarrow d_2)$; support 4; confidence 100%

$(a, a_2 \rightarrow a_1) \Rightarrow (d, d_2 \rightarrow d_1)$; support 4; confidence 100%

$(a, a_1 \rightarrow a_2) \wedge (b, b_1) \Rightarrow (d, d_1 \rightarrow d_2)$; support 1; confidence 100%

$(a, a_1 \rightarrow a_2) \wedge (b, b_2) \Rightarrow (d, d_1 \rightarrow d_2)$; support 1; confidence 100%

$(a, a_2 \rightarrow a_1) \wedge (b, b_1) \Rightarrow (d, d_2 \rightarrow d_1)$; support 1; confidence 100%

$(a, a_2 \rightarrow a_1) \wedge (b, b_2) \Rightarrow (d, d_2 \rightarrow d_1)$; support 1; confidence 100%

After we generate all possible action rules from every subsystem, we calculate the similarity matrix, using the metric defined above. Next, we calculate the degrees of similarity between every pair of objects:

$Similarity(o_1, o_2) = 0$

$Similarity(o_1, o_3) = (2 - |.5 - 1| - \left|\frac{2}{4} - \frac{4}{4}\right|) + (2 - |1 - 1| - \left|\frac{2}{4} - \frac{4}{4}\right|) + (2 - |1 - 1| -$

$\left|\frac{1}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) = 1 + 1.5 + 2 + 2 = 6.5$

$Similarity(o_1, o_4) = (2 - |.5-1| - \left|\frac{2}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{2}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) = 1.25 + 1.75 + 2 + 2 = 7$

$Similarity(o_1, o_5) = (2 - |.5-1| - \left|\frac{2}{4}-\frac{4}{4}\right|) + (2 - |1-1| - \left|\frac{2}{4}-\frac{4}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) = 1 + 1.5 + 2 + 2 = 6.5$

$Similarity(o_2, o_3) = 0$; $Similarity(o_2, o_4) = 0$; $Similarity(o_2, o_5) = 0$

$Similarity(o_3, o_4) = (2 - |1-1| - \left|\frac{4}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{4}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) = 1.25 + 1.25 + 2 + 2 = 6.5$

$Similarity(o_3, o_5) = (2 - |1-1| - \left|\frac{4}{4}-\frac{4}{4}\right|) + (2 - |1-1| - \left|\frac{4}{4}-\frac{4}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) = 2 + 2 + 2 + 2 + 2 + 2 = 12$

$Similarity(o_4, o_5) = (2 - |1-1| - \left|\frac{1}{4}-\frac{4}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{4}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) + (2 - |1-1| - \left|\frac{1}{4}-\frac{1}{4}\right|) = 1.25 + 1.25 + 2 + 2 = 6.5$

Here, we show the corresponding similarity matrix:

Table 32: Similarity matrix for the hybrid hierarchical object-driven example

	Object ID 1	*Object ID 2*	*Object ID 3*	*Object ID 4*	*Object ID 5*
Object ID 1	∞	0	6.5	7	6.5
Object ID 2	0	∞	0	0	0
Object ID 3	6.5	0	∞	6.5	12
Object ID 4	7	0	6.5	∞	6.5
Object ID 5	6.5	0	12	6.5	∞

As shown in Fig. 5, only by visual examination of the dendrogram, system users are able to grasp a decent idea of the hierarchical structure of any information system. It is clear that the two most similar objects are Object 3 and Object 5, hence they

would form the first cluster; Object 1 and Object 4 are then clustered together; next, the two clusters are joined together to form one combined cluster; and finally Object 2 joints the cluster.

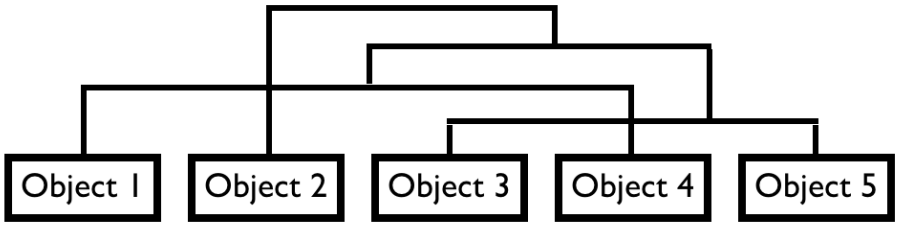

Figure 5: Corresoponding dendrogram for Table 32; using the hybrid hierarchical object-drive action rule approach.

By applying the proposed hybrid approach based on hierarchical clustering prior to rule extraction, we provide a substantial advantage to system users. Essentially, domain experts are given the convenience of choosing the level of generalization (or specialization) most appropriate to their application. As seen in Fig. 5, the degree of generalization can be specified by the level on which we decide to set the vertical threshold of the hierarchy; after predetermining the level of generalization, our objects are clustered accordingly, and finally action rule extraction is applied. At the lowest level, objects are not clustered together, and hence treated the way a typical object-driven rule-extracting approach. On the other hand however, at the highest level, all objects are clustered together, forming one big cluster, which will be identical to applying the classical action rule extraction approach.

7.3 Results of Applying Hierarchical Action Rules to Hypernasality

In this subsection, we provide sample results of action rules extracted by applying the hybrid hierarchical object-driven approach, while using three different levels of

generalization. As we have seen in Table 30, the more patients (or objects) we tend to cluster together, the more potential action rules are to be extracted, with relatively higher support (Figure 6 shows a more detailed chart of how the number of action rules increases as we decrease the number of clusters); however, by combining (or clustering) patients (or objects) into super-groups, generalization effect will be more apparent; until it reaches a point where generalization may cause negative effects; for that reason, using multiple levels of generalization is usually advised. Next, we provide action rules extracted from the hybrid hierarchical object-driven approach proposed in this paper using various levels of generalization, ranging from 10 clusters to 40.

Rule 1. $r_1 = (\textit{palatine tonsils}, \leq 2) \wedge (\textit{difference level F1-F2}, \leq 4.5 \rightarrow [6.5, 9.5])$ $\Rightarrow (\textit{Czermak's mirror test}, 1 \rightarrow 0.5)$; $\textit{supp}(r_1) = 18,\ \textit{conf}(r_1) = 100\%$.

This rule has a relatively high support; it states that for patients who are experiencing low levels of hypertrophied, by increasing the difference between the first two formants of the vocal tract for */i/ - long*, we expect for the patient's condition to improve from 1 to .5.

Rule 2. $r_2 = (\textit{palatine tonsils}, \leq 2) \wedge (\text{motility}, \geq 5.5) \wedge (\textit{yeaou}, \geq 7.5 \rightarrow [3.5, 4.5])$ $\Rightarrow (\textit{Czermak's mirror test}, 2 \rightarrow 0)$; $\textit{supp}(r_2) = 12,\ \textit{conf}(r_2) = 66.6\%$.

This rule states that for patients who are experiencing low levels of hypertrophied, and that have high motility level for their soft palate, by decreasing the levels of nasalization for vowels /I, e, a, o, u, i/, we expect the patient's condition to improve substantially from 2 to 1.

Rule 3. $r_3 = (\textit{palatine tonsils}, \leq 2) \wedge (\text{motility}, \geq 5.5 \rightarrow \leq 3.5) \wedge (\textit{bdg}, \geq 8.5 \rightarrow \leq 6.5)$ $\Rightarrow (\textit{Czermak's mirror test}, 2 \rightarrow 0)$; $\textit{supp}(r_3) = 18,\ \textit{conf}(r_3) = 75\%$.

Similar to rule 1, this rule has a relatively high support; it states that for patients who are experiencing low levels of hypertrophied, by both changing the motility level from greater than 5.5 to less than 3.5, and by decreasing the nasalization of /b, d, g/, we expect the patient's condition to improve substantially from 2 to 1.

Rule 4. $r_4 = (\textit{palatine tonsils}, \leq 2) \wedge (\text{diagnosis, III}) \wedge (\textit{yeaou}, \geq 7.5 \rightarrow [3.5, 4.5])$ $\Rightarrow (\textit{Czermak's mirror test}, 1 \rightarrow 0)$; $\textit{supp}(r_4) = 3,\ \textit{conf}(r_4) = 100\%$.

The above rule states that for patients who are experiencing low levels of hypertrophied, and that have been diagnosed with hypertrophy of adenoids and possibly palatine tonsils, by decreasing the level of nasalization for vowels /I, e, a, o, u, i/, we expect the patient's condition to improve from 1 to 0, which would completely remove any hypernasality.

Rule 5. $r_5 = (\textit{sleep apnoea}, \leq 2) \wedge (\text{bdg}, \leq 6.5 \rightarrow \geq 8.5) \wedge (\textit{motility}, \leq 3.5 \rightarrow [3.5, 4.5]) \Rightarrow (\textit{Czermak's mirror test}, 1 \rightarrow 0)$; $\textit{supp}(r_5) = 2,\ \textit{conf}(r_5) = 100\%$.

The above rule states that for patients who are experiencing low levels of sleep apnoea, by both increasing the level of nasalization for /b, d, g/, and by increasing the level of motility for the soft palate, we are expected to observe an improvement in the patient's condition from 1 to 0.

Rule 6. $r_6 = (\text{difference level F1-F2}, \leq 4.5) \wedge (\textit{motility}, [4.5, 5.5] \rightarrow [3.5, 4.5]) \Rightarrow (\textit{Czermak's mirror test}, 1 \rightarrow 0)$; $\textit{supp}(r_6) = 6,\ \textit{conf}(r_6) = 100\%$.

This rule states that for patients who have difference between the first two formants

of the vocal tract for */i/ - long* less than 4.5, by decreasing the motility level of the soft palate, we would expect the patient's condition to slightly improve from 1.5 to 1.

Rule 7. $r_7 = (\text{diagnosis, AD}) \wedge (\mathit{motility}, [4.5, 5.5] \rightarrow [3.5, 4.5] \Rightarrow$ $(\mathit{Czermak's\ mirror\ test}, 1 \rightarrow 0)$; $\mathit{supp}(r_7) = 6,\ \mathit{conf}(r_7) = 100\%$.

This rule states that for patients who have had their adenoids removed by adenotomy surgery, we can improve their condition from 1 to .5 by decreasing the motility level of the soft palate.

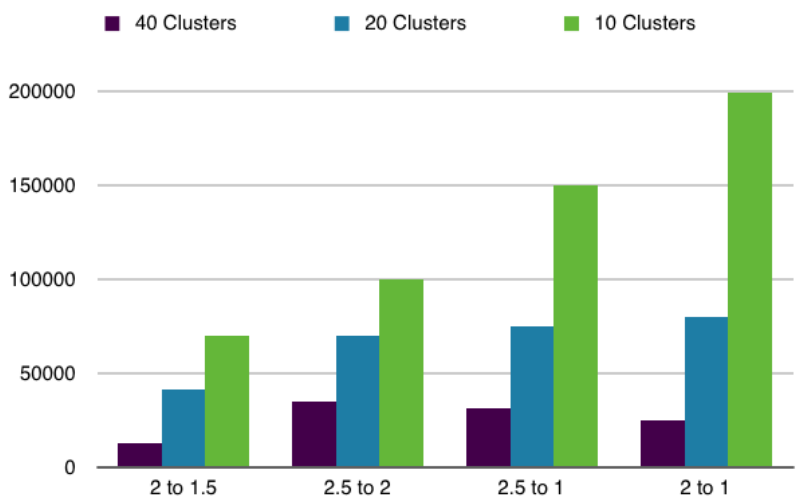

Figure 6: This figure shows how the number of action rules increases as the number of clusters decreases for some desired transitions in the Czermak's mirror test.

CHAPTER 8: CONCLUSION

In the first two chapters, we presented all necessary background and motivations for the actual novel body of work for object-driven action rules; starting with general concepts of data mining, such as knowledge representation, rule-based knowledge discovery and association rules, presented in Chapter 1; then moving to a yet more related, but vital topic of knowledge discovery, known as action rules; which is the base that our work of object-driven action rules has been building on top of.

In Chapter 3, we provided a detailed description of our hypernasality treatment information system, which also served as our case study for this body of work. Perhaps it is now a good time to acknowledge the help and support by thanking Dr. Danuta Chojnacka-Wądołowska and Dr. Cecylia Konopka from Children's Memorial Health Institute in Warsaw for their help with data collection and providing medical diagnoses.

Following Chapter 3, we provided our own and novel work of the subarea of object-driven and temporal action rules; presented in Chapter 4. We thought that it would be best to divide the body of work into two main component, namely the object-driven constraint and the temporal assumption. In each, we tried to provide a complete work, starting with the motivation of the assumption, following with the modifications and new definitions that we introduced. Furthermore, we divided the description of the temporal assumption into two areas, one which we called the classical object-

driven approach, and the other in which we called the pair-based approach. Following the detailed explanation of each component of our work, we presented the results of mined action rules generated when applying our work. As mentioned in their corresponding sections, all results matched the experience and preexisting practices of physicians; having said that, some patterns were highly interesting as they provided hidden insights, as our collaborators commented.

In Chapter 5, we introduced a second (much larger) dataset in which we extracted temporal-based and object-driven action rules in real-time. This approach is extremely useful in cases where our dataset is huge; hence, it would not be efficient to extract action rules beforehand for all possible cases (since there are too many), instead we wait for a new instance (patient) to come into the system, and extract specialized action rules for that particular case.

In Chapter 6, we applied our object-driven approach to an entirely different class of action rules, called action reducts; which is yet another evidence that the object-driven (and temporal-based) approach is flexible enough to be adapted by other data mining system. Finally in Chapter 7, we introduced a hybrid approach in which we combine the more accurate but less vast approach of object-driven and temporal action rules, with the less accurate but more vast approach of the classical action rule approach; and show results accordingly.

REFERENCES

[1] Raś, Z.W., Wieczorkowska, A.: Action-Rules: How to increase profit of a company. In: Zighed, D.A., Komorowski, J., Zytkow, J. (eds.) Proceedings of PKDD 2000, LNAI, Vol. 1910, Springer, 2000, 587–592

[2] He, Z.X. Xu, S. Deng, R. Ma (2005). Mining Action Rules from Scratch. Expert Systems with Applications, Elsevier, Vol. 29, No. 3, 691-699.

[3] Hajja, A., Wieczorkowska, A., Raś, Z. W., Gubrynowicz, R.: Object-driven Action Rules and their Application to Hypernasality Treatment. In: Proceedings of ECML-PKDD Workshop on New Frontier in Mining Complex Patterns, Bristol, UK, September 24-28, 2012, 104-115

[4] Hajja, A., Wieczorkowska, A., Raś, Z. W., Gubrynowicz, R.: Pair-based Object-driven Action Rules. In: Post-Proceedings of the First ECML/PKDD 2012 International Workshop on New Frontiers in Mining Complex Patterns, LNAI 7765, Springer, 2013

[5] Wasyluk, H., Raś, Z.W., Wyrzykowska, E.: Application of Action Rules to HEPAR Clinical Decision Support System. Experimental and Clinical Hepatology 4(2), 2008, 46–48

[6] Zhang, X., Raś, Z.W., Jastreboff, P.J., Thompson, P.L.: From Tinnitus Data to Action Rules and Tinnitus Treatment. In: Proceedings of 2010 IEEE Conference on Granular Computing, Silicon Valley, CA, IEEE Computer Society, 2010, 620–625

[7] Nekvapil, V.: Data Mining in the Medical Domain. Lambert Academic Publishing, 2010.

[8] Zhang, H., Zhao, Y., Cao, L., Zhang, C.: Combined Association Rule Mining. Asia Conference on Advances in Knowledge Discovery and Data Mining, Proceedings, LNCS, Springer, Vol. 5012, 2008, 1069-1074

[9] Wang, K., Jiang, Y., Tuzhilin, A.: Mining Actionable Patterns by Role Models. 22nd International Conference on Data Engineering, Proceedings, Atlanta, 2006, 16-26.

[10] Suzuki, E.: Discovering Action Rules that are Highly Achievable from Massive Data. PAKDD 2009, Proceedings. LNCS (LNAI), Springer, Vol. 5476, 713-722.

[11] Rauch, J., Simunek, M.: Action Rules and the GUHA Method: Preliminary Considerations and Results. Foundations of Intelligent Systems, Springer, LNAI 5722, 2009, 76-87.

[12] Raś, Z.W., Wieczorkowska, A. (Eds): Advances in Music Information Retrieval, Studies in Computational Intelligence, Vol. 274, Springer, 2010

[13] Raś, Z., Dardzińska, A.: From Data to Classification Rules and Actions. International Journal of Intelligent Systems 26(6), Wiley, 2011, 572–590

[14] Pawlak, Z.: Information systems - theoretical foundations. Information Systems Journal 6, 1981, 205–218

[15] Bazan, J. and Szczuka, M.: The rough set exploration system. In: Transactions on rough sets III, 25–42 (2005)

[16] Im, S., Raś, Z.W. , Tsay L.-S.: Action Reducts. In: Foundations of Intelligent Systems, Proceedings of ISMIS 2011 Symposium, LNAI, Vol. 6804, Springer, 2011, 62-69

[17] Zdzislaw Pawlak. 1992. Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Norwell, MA, USA.

[18] Skowron, A.: Rough Sets and Boolean Reasoning. In: Pedrycz, W. (Eds.) Granular Computing: An Emerging Paradigm, pp. 95-124. Springer, Heidelberg (2001)

[19] Pawlak, Z.: Rough sets. In: International Journal of Computer Information and Science 341-356 (1982)

[20] Raś Z.W., Tzacheva A., Tsay L.S., Grüdal O.: Mining for interesting action rules. In: Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2005), IEEE Computer Society, 2005, 187-193

[21] Raś Z.W., Wyrzykowska E., Wasyluk H.: Action rules discovery based on agglomerative strategy. In: Mining Complex Data, Post-Proceedings of 2007 ECML/PKDD Third International Workshop (MCD 2007), Springer, LNAI, Vol. 4944, 2008, 196-208

[22] Tsay L.S., Raś Z.W.: Action rules discovery system DEAR3. In: Foundations of Intelligent Systems, Proceeding of ISMIS 2006, Springer, LNAI, Vol. 4203, 2006, 483-492.

[23] Raś, Z.W., Dardzińska, A., Tsay, L.S., Wasyluk, H.: Association action rules. In: IEEE International Conference on Data Mining Workshops, 2008, 283–290

[24] Mayer-Schönberger, V., Cukier, K.: Big Data: A Revolution That Will transform How We Live, Work, and Think, Eamon Dolan/Houghton Mifflin Harcourt, 2013.

[25] U.S. Department of Health and Human Services Centers for Medicare and Medicaid Services 42 CFR Parts 412, 413, 422 et al. Medicare and Medicaid Programs; Electronic Health Record Incentive Program; Final Rule

[26] Gubrynowicz, R.; Chojnacka-Wądołowska, D., Konopka, C.: Assessment of Velum Malfunction in Children Through Simultaneous Nasal and Oral Acoustic Signals Measurements. Archives of Acoustics 32(1), 2007, 165–175

[27] SAMPA - computer readable phonetic alphabet, http://www.phon.ucl.ac.uk/home/sampa/

[28] Rakesh Agrawal, Tomasz Imieliski, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data (SIGMOD '93), Peter Buneman and Sushil Jajodia (Eds.). ACM, New York, NY, USA, 207-216. DOI=10.1145/170035.170072 http://doi.acm.org/10.1145/170035.170072

[29] Michael Paul and Mark Dredze: You Are What You Tweet : Analyzing Twitter for Public Health. In: 5th Interational Conference on Weblogs and Social Media, 2011

[30] Bermingham, Adam and Smeaton, Alan F.: On using Twitter to monitor political sentiment and predict election results. In: Sentiment Analysis where AI meets Psychology (SAAIP) Workshop at the International Joint Conference for Natural Language Processing (IJCNLP), 13th November 2011.

[31] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. 2010. A latent variable model for geographic lexical variation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 1277-1287.