

## An Antimicrobial Prescription Surveillance System that Learns from Experience

**Mathieu Beaudoin and Froduald Kabanza**

Department of Computer Science  
Université de Sherbrooke, Canada

{mathieu.beaudoin, froduald.kabanza}@usherbrooke.ca

**Vincent Nault and Louis Valiquette**

Department of Microbiology and Infectiology  
Université de Sherbrooke, Canada

{vincent.nault, louis.valiquette}@usherbrooke.ca

### Abstract

Inappropriate prescribing of antimicrobials is a major clinical and health concern, as well as a financial burden, in hospitals worldwide. In this paper, we describe a deployed automated antimicrobial prescription surveillance system that has been assisting hospital pharmacists in identifying and reporting inappropriate antimicrobial prescriptions. One of the key characteristics of this system is its ability to learn new rules for detecting inappropriate prescriptions based on previous false alerts. The supervised learning algorithm combines instance-based learning and rule induction techniques. It exploits temporal abstraction to extract a meaningful time interval representation from raw clinical data, and applies nearest neighbor classification with a distance function on both temporal and non-temporal parameters. The learning capability is valuable both in configuring the system for initial deployment and improving its long term use. We give an overview of the application, point to lessons learned so far and provide insight into the machine learning capability.

### Introduction

Inappropriate prescribing of antimicrobials (ATM) is a major clinical problem and health concern, as well as a financial burden, in hospitals worldwide. It has been reported that as many as 50% of ATM prescriptions are unnecessary or inappropriate (Dellit et al. 2007). ATM stewardship programs have been shown to reduce avoidable adverse effects (toxicity, ATM resistance, *Clostridium difficile*, etc. (Dellit et al. 2007; Valiquette et al. 2007)), improve patient health and reduce unnecessary costs. However, these programs require the review of an overwhelming amount of clinical data by dedicated experts, which proves to be an obstacle in the current context of limited healthcare resources. Therefore, hospitals are increasingly relying on automated decision support systems to review hospital-wide ATM prescriptions.

For the most part, prescription monitoring systems use a knowledge base (KB) of rules acquired from experts to detect inappropriate prescriptions and prevent potential adverse events. Local and commercial solutions (e.g., Premier Inc.; Pharmacy OneSource) are generally characterized by highly sensitive rules with poor precision that trigger a high rate of clinically unhelpful alerts (Hsieh et al. 2004;

Reichley et al. 2005). This high rate of false alerts impedes their use. The problem comes from the inability to create a complete and precise KB and the tendency to otherwise use a “totally inclusive” KB. It is difficult to model all variables that a prescribing physician will take into account, let alone model the decision-making process. ATM prescribing is a subjective process where physicians continually rely on their experience to select an effective treatment and prevent adverse events. In addition to published guidelines, hospitals have their own local practices (Reichley et al. 2005) that must be covered by these rules.

We have developed and deployed a new ATM prescription monitoring system called APSS – antimicrobial prescription surveillance system. Like other ATM prescription monitoring systems, APSS uses a KB of ATM prescription rules. As such, it also suffers from a high rate of false alerts. However, unlike any other previous system, APSS is able to learn new prescription surveillance rules. This learning capability is designed to allow APSS to self-reconfigure to local practices after deployment and to self-improve its KB over the long term supervised by user feedback. Although the application of machine learning to clinical temporal data is not new, to the best of our knowledge, this is its first application to the monitoring of drug prescriptions.

Prescriptions are temporal data by nature. ATM prescriptions are valid over periods of time; after selecting an initial treatment, a physician must review his earlier prescription to account for newly available information. Monitoring laboratory test results and variations in the patient’s health condition is critical since it can render an initially appropriate treatment inappropriate. Taking this temporal nature into account, APSS uses a supervised learning algorithm for discovering rules that classify temporal data. We approach the problem as a binary classification task into good and bad temporal sequences (i.e., prescriptions). The algorithm we use is a combination of rule induction and instance-based learning methods.

In the next section, we give an overview of APSS and discuss its development, deployment, and evaluation thus far. We then discuss the ongoing development of the machine learning extension. We follow with a presentation of preliminary results for this learning capability and conclude with future work.

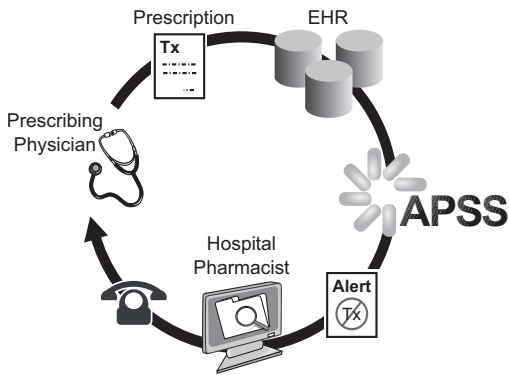


Figure 1: Asynchronous surveillance using APSS

## Antimicrobial Prescription Surveillance System

APSS is currently deployed at the *Centre Hospitalier Universitaire de Sherbrooke* (CHUS), a 713-bed Canadian medical academic centre. It assists clinical pharmacists in their ATM stewardship activities by identifying mismatches between prescribed ATM and published and local guidelines. APSS increases the amount of prescriptions revised daily and increases the impact of interventions by allowing reviewers to focus on alerts with higher clinical impact. Its KB includes approximately 50,000 rules for identifying inappropriate drug-drug interactions, drug-bug or drug-laboratory mismatches, cheaper alternatives, maximum daily dose, maximum and minimum dose and frequency, maximum duration, and route of administration.

APSS assists in the post-prescription revision process, as illustrated in Figure 1. The prescriber chooses an ATM therapy after an initial assessment of the patient. The Pharmacy Department performs *a posteriori* computerized order entry into our centre's electronic health record (EHR) system, *QuadraMed Computerized-Patient Record*. New and modified prescriptions are automatically sent to APSS along with all patient information. APSS reviews these prescriptions using the patient's most recent clinical information and produces documented alerts for potentially inappropriate prescriptions. The pharmacist first reviews these alerts and then contacts the prescriber by phone to recommend a prescription modification or discontinuation if deemed appropriate. If the prescriber accepts the recommendation, a new order is sent to the pharmacy and the cycle continues. The pharmacist records the validation of every revised alert.

The project began in 2005 with the objective of finding a solution to facilitate manual ATM optimization. Intensive manual ATM optimization was required in 2005 to control province-wide outbreaks of *C. difficile* infections by decreasing the use of high-risk antibiotics associated with *C. difficile*. ATM optimization decreased overall use of ATM during this period (Valiquette et al. 2007) and the outbreak subsided. However, these measures required important resources that could not be sustained subsequently. After manual surveillance ended, overall ATM consumption eventually returned and surpassed pre-outbreak levels.

The solution put forward was to use an automated prescription monitoring system to facilitate and enhance our ATM stewardship activities. We selected an asynchronous revision process because it dovetails nicely with the "pen and paper" prescribing practices that are still the norm at our centre. Synchronous validation at the time of computerized order entry was ruled out because it was perceived to hinder the prescribing physicians' workflow. Introducing APSS to our centre's ATM stewardship program was simple; we provided a tool that assists the clinical pharmacists already assigned to ATM revision, resulting in a transparent integration to the existing prescription practice.

We set about to develop APSS in 2007 and began with the complex task of developing its KB. Our multidisciplinary team consisted of one professor/researcher assisted by a doctoral candidate in computer science and a programmer-analyst, one infectious diseases physician/researcher assisted by a doctoral candidate in clinical science, and one hospital pharmacist. We extracted rules from published and local guidelines and tested them retrospectively to validate the potential for prospective interventions.

The approach used by APSS to revise ATM prescriptions is inspired from a standardized model for evaluating antibiotic prescriptions published by Gyssens *et al.* (Gyssens et al. 1992) and depicted in Figure 2. Following this model, a prescribed ATM therapy is considered valid unless it violates specified contraindications. The revision process consists of evaluating different parameters of importance for contraindications to the selected therapy. Prescriptions can be inappropriate according to different parameters, in which case multiple alerts are triggered. This revision process does not only occur at the time of prescription, but continues as additional clinical information becomes available. The ATM therapy is in principle revised as additional clinical information comes in. To illustrate, the physician may diagnose a bacterial infection after an initial clinical evaluation of the patient, prescribe a particular ATM therapy and, in the meantime, order a microbiology test. Three days later, the microbiology test might reveal the bacteria's susceptibility profile. Assuming the profile indicates that the bacteria is resistant to the previously prescribed ATM therapy, that is, the previously prescribed ATM therapy is ineffective against this particular bacteria, then the prescription should be revised for a more effective alternative.

As illustrated in Figure 2, prescription revision according to the Gyssens model begins by verifying whether there is sufficient available information for evaluating a prescription (Step VI). The revision process halts whenever critical information is missing. It then looks for clinical contraindications to the prescribed ATM therapy (Step V). As new clinical information becomes available, the ATM therapy evolves from the initial empiric broad spectrum therapy for a suspected infection to a specific narrow spectrum therapy targeted toward a specific pathogen (Step IVd). This new information may also indicate a more effective alternative (Step IVa), an equally effective but less toxic alternative (Step IVb) or an equally effective but less costly alternative (Step IVc). The total duration of the ATM therapy is also monitored. A too short therapeutic course (Step IIIb) may lead to treatment

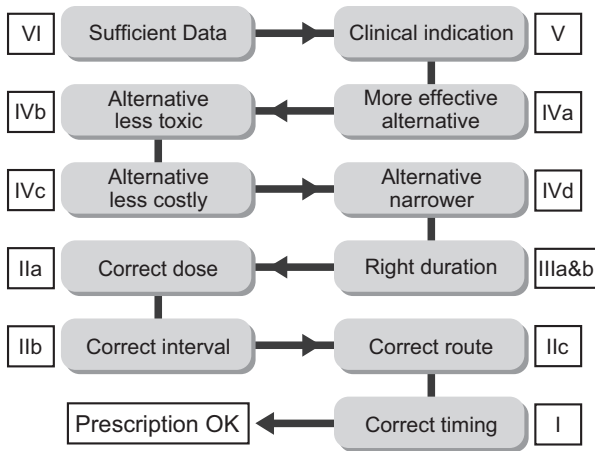


Figure 2: Standardized model to evaluate antibiotic prescriptions published by (Gyssens et al. 1992)

failure whereas inadequately prolonged therapy (Step IIIa) increases the risk of adverse drug events and may be associated with increased antimicrobial resistance. Doses are validated (Step IIIa) along with corresponding dosing frequency (Step IIb) to ensure effective serum concentration. The route of administration (Step IIc) must ensure delivery of sufficient concentration of ATM to the targeted site. The timing (Step I) must also be questioned, since therapy administered too early or too late could prove useless, if not harmful.

The design of the KB for APSS follows the above Gyssens model. Each step in this process corresponds to a set of contraindication rules to be checked. We characterized and segmented the ATM domain knowledge according to these steps, regrouping rules to be evaluated together (e.g., adequacy of dose *and* dosing interval). We also created higher level attributes, such as patient “type”, derived from multiple attributes that represent clinical concepts used by experts when reasoning about ATM therapies. Rules were created by category as a conjunction of propositions on relevant prescription parameters, patient data, patient type, etc. The rule segmentation proved to be helpful in maintaining the KB. It also contributes to the system’s efficiency in that the firing of rules in each step concerns a small set of rules, as opposed to firing all the rules in one step.

The Expert System for APSS (the KB and the algorithm for checking contraindication rules) was developed in C#.NET, using SQL-compatible database tables for the KB. The KB can be visualized and edited by using the Knowledge Management tool. The KB is for the most part maintained by clinical pharmacists. The maintenance in particular involves making changes that are required to reflect updates in published guidelines and local prescribing practices. Changes that involve new rule structures (e.g., new parameters that are not accounted for in the KB) require the assistance of a computer programmer. Since the deployment of APSS in August 2010, the interventions of a programmer have been quite limited. One instance occurred during a recent minor outbreak of *C. difficile* infections. The committee on health care-associated infections requested the addition

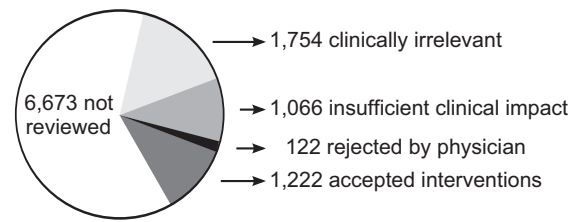


Figure 3: Alerts were triggered for 10,837 prescriptions during the first 53-week period

of temporary rules to monitor every high-risk ATM.

APSS can communicate with the CHUS’ EHR through a data communication interface that we have developed. In collaboration with the Information Technology (IT) Department, we identified the required variables and normalized their values. We developed the exportation and importation interfaces using Health Level Seven (HL7) standards. We deployed APSS’ databases and Knowledge Management Module. Support from the CHUS’ decision makers and IT Department management was required to ensure advancement of these steps.

In August 2010, we finally deployed APSS at the CHUS, making it the first ATM prescription monitoring system in Canada. During a 53-week period, a clinical pharmacist used APSS an average of 15 hours per week. APSS evaluated 37,770 ATM prescriptions. As summarized on Figure 3, alerts were triggered for 10,837 (29%) of these prescriptions. However, alerts for 6,673 prescriptions could not be reviewed because they occurred outside of the allocated time. Of the 4,164 reviewed alerts, 2,820 (68%) were overridden by a pharmacist; 1,754 (42%) were considered clinically irrelevant while another 1,066 (26%) did not have sufficient clinical impact to justify modifying the current treatment (e.g., alert triggered at the end of a prescription). The pharmacist contacted prescribers 1,344 times and they accepted 1,222 (91%) recommendations. Our evaluation period was associated with a reduction of 13.5% in ATM consumption and CAD 305,000 (15%) in ATM expenditures.

APSS was met by prescribers and decision makers with universal appreciation and recognition. In September 2011, full-time surveillance began, divided among a team of five pharmacists. More time allocated to APSS provided more feedback that helped us further improve APSS’ KB. As of September 2012, the override rate of alerts had subsided to 50%, with an acceptance rate by prescribers stable at 91%. APSS had contributed to 3,156 interventions, which were associated with reductions of 22% in intravenous ATM consumption and CAD 688,000 in overall ATM expenses. APSS also enabled us to extend our surveillance from high-risk wards (e.g., intensive care) to every bed of the CHUS’ two physical sites. We are currently evaluating the impacts of APSS on patient health.

### APSS Learning Capability

From the beginning of the project we started investigating a mechanism to improve APSS’ KB from experience. The

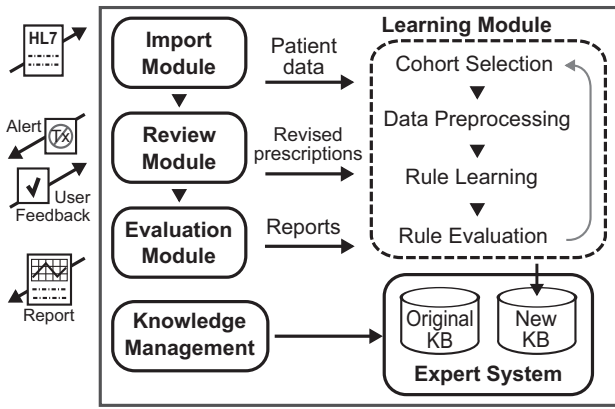


Figure 4: Overview of APSS' Learning Module

alerts were all documented, along with feedback from pharmacists and prescribers. In 2011, we began implementing the Learning Module whose aim is to discover new rules for classifying inappropriate prescriptions supervised by user feedback, such as the rejection of false alerts or the identification of unflagged inappropriate prescriptions.

The Learning Module interacts with APSS' other modules to discover and test new rules using patient data and revised alerts (see Figure 4). The Import Module is responsible for acquiring and normalizing clinical data from the CHUS' EHR and storing it into APSS' database. The Review Module uses APSS' KB to review ATM prescriptions and stores user feedback for revised alerts. The Evaluation Module produces custom reports on ATM consumption, alerts, etc.

The Learning Module enables pharmacists to discover rules for specific ATM and alert types. The user will use the Evaluation Module to identify commonly overridden alerts and patients who received specific ATM and did or did not present the alerts of interest. This data will be used to create datasets for training and testing rule sets. Rules with sufficient recall (sensitivity) and accuracy will be suggested for review. The user will be able to add clinically relevant rules to the KB.

This learning capability is expected to help configure APSS whenever it is deployed at a new hospital that has prescription practices differing from the CHUS' where APSS is currently deployed. More importantly, this capability will be expected to improve APSS in the long run in every hospital where it is deployed. Another expected use of this module is data mining to discover unforeseen yet clinically relevant patterns of inappropriate prescribing that may be addressed by our stewardship program with targeted in-service training. One such pattern was discovered during our experimentation and is discussed in the Preliminary Results section.

### Cohort Selection and Data Preprocessing

For the experiments discussed later in this paper, we considered the following patient cohort: every adult inpatient (18 years of age and older) receiving at least one monitored ATM admitted between January 1, 2012 and June 30, 2012.

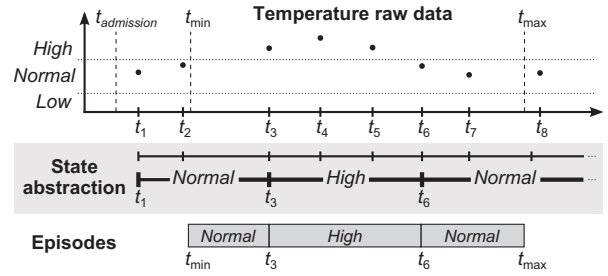


Figure 5: Example of state abstraction for the *temp* attribute

A cohort of 7,740 hospitalizations was created, consisting of 5,756 distinct patients who received 19,172 ATM prescriptions. Alerts were triggered for 7,027 prescriptions. We considered the following attributes: *gender*, *age*, Body Mass Index (*BMI*), patient location (*ward*), temperature (*temp*), white cell count (*WCC*), neutrophil count (*neut*), creatinine clearance (*CrCl*), respiratory rate (*resp*), *pulse*, and blood pressure (*BP*). An attribute was also created for each medication, where prescriptions were described using their *name*, *dose*, *frequency*, and *route* of administration, as well as their revised alerts.

We used temporal abstraction (Shahar 1997) to extract a uniform and meaningful data representation from the raw clinical data of APSS. This data contains qualitative and quantitative attributes sampled with both time points (e.g., *temp*) and time intervals (e.g., drug order). Figure 5 illustrates the process of state abstractions for the raw *temp* time series. Quantitative thresholds are used to identify qualitative states that hold over a period of time, which we call *episodes*. We used state abstractions on attributes *BMI*, *temp*, *WCC*, *neut*, *CrCl*, *resp*, *pulse* and *BP*. We extracted a single sequence for each hospitalization. Within a hospitalization, the observation period was restricted to the ongoing ATM of interest. We considered only data between the first ( $t_{min}$ ) and last ( $t_{max}$ ) administered dose. It ensured a common time zero ( $t_{min}$ ) between sequences. We used a temporal granularity of 1 hour.

### Selecting the Learning Algorithm

There are various applications of data mining and machine learning algorithms to clinical temporal data. Association rule discovery has been used to gain insight into the causes of clinical events of interest (Bellazzi et al. 2005; Concaro et al. 2009); however it is geared towards discovering rules for frequent patterns and performs poorly when addressing infrequent patterns (Zaki, Lesh, and Ogihara 2000), such as inappropriate prescriptions. It uses an *Apriori*-like strategy (Agrawal and Srikant 1994) with breadth-first search and candidate pruning based on *support* and *confidence*. The problem when looking for infrequent patterns is the necessity to lower support thresholds. It inefficiently prunes the candidate space and potentially leads to an intractable search space. It also produces an overwhelming amount of uninteresting patterns from which it is difficult to distinguish interesting ones (Zaki, Lesh, and Ogihara 2000).

Another method that was used to identify clinical events

of interest is case-based reasoning. Whereas instance-based learning (Aha, Kibler, and Albert 1991) accumulate observed instances and classify new instances using the nearest known ones, case-based reasoning (Aamodt and Plaza 1994) uses background information to create meaningful cases that are reused or adapted, in full or in part, to solve new problems. For example, case-based reasoning has been used to identify potential adverse drug events (Hartge, Wetter, and Haefeli 2006) and hemodialysis treatment failures (Montani, Portinale, and Leonardi 2006). While they are known to perform well with few instances, these algorithms are burdened with irrelevant attributes (Domingos 1996) and accumulate large quantities of cases. This is a problem when looking for a small set of highly accurate, concise and comprehensible rules aimed at a human user.

A complementary approach to instance-based learning is rule induction. Rule induction disposes easily of irrelevant features, separates classes with good accuracy and extracts a small set of rules that can lead to better predictions (Domingos 1996). However, it tends to be affected by a skewed distribution of classes and produces rules that favor the over-represented classes (Chawla, Japkowicz, and Kotcz 2004).

Following (Domingos 1996), we have chosen an algorithm that combines instance-based learning and rule induction. However, unlike (Domingos 1996), which learns classification rules for a labeled set of non-temporal feature-value data, our algorithm was designed to learn classification rules for a labeled set of episode sequences in addition to non-temporal feature-value data.

## Formulating the Learning Problem

Let us consider the attribute space  $A$  as the finite set of attributes for our domain and the feature space  $F$  as the finite set of qualitative values observed for these attributes. An episode  $e$  is defined as  $\langle a, f, ts, te \rangle$ , where  $(a = f)$  describes a symbolic state with  $a \in A$  and  $f \in F$  holding over the time interval  $[ts, te]$ . We refer to the attribute, feature, start, and end times of an episode as  $e.a$ ,  $e.f$ ,  $e.ts$ , and  $e.te$ , respectively. An example of episode from Figure 5 is  $\langle temp, normal, t_{min}, t_3 \rangle$ .

A sequence  $s$  is defined by  $\{e_1, \dots, e_n | \forall i = 1, \dots, n-1 : e_i.ts \leq e_{i+1}.ts\}$ , where  $n = |s|$  represents the size of the sequence. We refer to the subsequence of  $s$  for the  $i$ th attribute  $a_i \in A$  as  $att_i(s)$  defined by  $\{e_1, \dots, e_m | \forall e \in att_i(s) : e \in s; e.a = a_i; \forall j = 1, \dots, m-1 : e_j.te \leq e_{j+1}.ts\}$ , where  $m = |att_i(s)|$ . A hospitalization is described as a labeled sequence  $ls$  defined as  $\{id, s, l\}$ , where  $id$  is a unique identifier,  $s$  is a sequence, and  $l$  is a class label that belongs to the finite set of class labels  $L$ . We focus on a binary-class problem where  $L = \{appropriate, inappropriate\}$ . We used APSS' revised alerts to label every sequence, where *inappropriate* indicates a true positive (alert that has been validated by a user) and *appropriate* indicates a negative (no alert) or false positive (alert that was rejected by a user).

We can now formally state the supervised machine learning problem that concerns us. Given a finite training set of labeled sequences  $TS$ , discover a rule set  $R$  for inappropriate sequences. We only have two classes (appropriate and inappropriate). Learned classification rules identify inap-

propriate instances. The antecedent of a learned rule is a conjunction of propositions over time intervals whose satisfaction implies membership to the *inappropriate* class as the consequent.

## Temporal Induction of Classification Models

Our supervised learning algorithm, called *Temporal Induction of Classification Models* (TIM) combines instance-based learning and rule induction. Its main operations are the following: at first, the rule set  $R$  is initialized using inappropriate sequences of the training set as maximally specific rules. Distances between rules and sequences of the training set are computed and stored in a multidimensional distance matrix to reduce computation times. These distances are used for nearest neighbor classification. Rules are modified in parallel to increase interclass distance. In each iteration, the most promising local modifications are selected according to the rule's most similar appropriate sequences. Conditions are eliminated or their time intervals are shortened. Local modifications are performed according to similar appropriate sequences until they no longer improve a rule. Duplicate rules covering the same sequences as another are removed.

The rules are evaluated according to the *J-measure* (Smyth and Goodman 1991), which quantifies the average information content of a rule. We selected the *J-measure* for its ability to account for both simplicity and *goodness-of-fit*, measuring the probability and *cross-entropy* of a rule (Smyth and Goodman 1991). As a working hypothesis, a rule with high information content (i.e., high probability and cross-entropy) is also likely to have a high predictive accuracy.

## Classification

The distance function measures the similarity between rules and sequences. Since the rules are intended to classify sequences that involve temporal and non-temporal data, we use a distance function that considers both temporal and non-temporal parameters. A non-symmetric distance function is used where similarity is proportional to the number of conditions that a sequence shares with a rule, i.e., a sequence is perfectly similar to a rule it subsumes.

Given a rule  $r \in R$  with  $N_r$  attributes and a sequence  $s \in TS$ , the global  $distance(r, s)$  function is defined by (1). Normalizing  $distance(r, s)$  by  $N_r$  creates a coefficient between  $[0, 1]$  that does not arbitrarily favor shorter rules, where 0 denotes perfect similarity. To ensure that irrelevant sequences are not labeled *inappropriate* by the nearest yet dissimilar rule, we enforce a minimal distance threshold  $D_{min}$  under which a rule is said to *cover* a sequence. A sequence covered by a rule is labeled *inappropriate*.

$$distance(r, s) = \frac{\sum_{i=1}^{N_r} D_a(att_i(r), att_i(s))}{N_r} \quad (1)$$

The  $D_a$  function measures the distance between the subsequences  $att_i(r)$  and  $att_i(s)$  for the  $i$ th attribute of  $r$ . If  $att_i(s) = null$ ,  $D_a = 1$ , otherwise we use (2), which measures the distance between the conditions  $c_j \in att_i(r)$  and episodes  $e_k \in att_i(s)$ . An indexing mechanism retrieves attribute-specific subsequences in  $O(1)$ . We normalize the distance  $D_a \in [0, 1]$  to avoid arbitrarily increasing

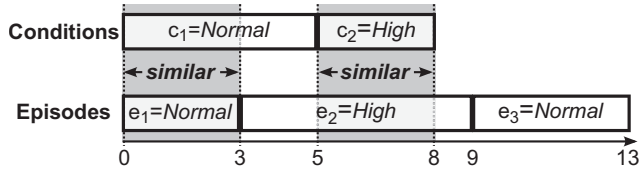


Figure 6: Example of a rule’s conditions and a sequence’s episodes

the weight of the  $i$ th attribute in the overall coefficient.

$$D_a(att_i(r), att_i(s)) = \frac{|att_i(r)| - \left( \sum_{j=1}^{|att_i(r)|} \sum_{k=1}^{|att_i(s)|} S_F(c_j, e_k) \times S_T(c_j, e_k) \right)}{|att_i(r)|} \quad (2)$$

**Feature similarity.**  $S_F$  measures the similarity between the symbolic features of  $c_j$  and  $e_k$  using the *overlap metric* where  $S_F(c_j, e_k) = 1$  if  $(c_j.f = e_k.f)$  and 0 otherwise.

**Temporal similarity.**  $S_T$  is proportional to the temporal overlapping of  $e_k$  over  $c_j$ , as measured by (3).  $S_T$  returns a coefficient between  $[0, 1]$ , where 1 implies  $[c_j.ts, c_j.te[ \subseteq [e_k.ts, e_k.te[$ .

$$S_T(c_j, e_k) = \frac{[c_j.ts, c_j.te[ \cap [e_k.ts, e_k.te[}{[c_j.ts, c_j.te[} \quad (3)$$

Consider the attribute-specific subsequences of Figure 6. A rule’s antecedent  $att_i(r)$  with conditions  $c_1$  and  $c_2$  overlaps a sequence’s  $att_i(s)$  with episodes  $e_1$ ,  $e_2$  and  $e_3$ . The distance between these subsequences is 0.2, which is computed as follows:

$$\begin{aligned} D_a(att_i(r), att_i(s)) &= \frac{2 - \left( \sum_{j=1}^2 \sum_{k=1}^3 (S_F(c_j, e_k) \times S_T(c_j, e_k)) \right)}{2} \\ &= \frac{2 - \left( \frac{(1 \times 0.6) + (0 \times 0.4) + (1 \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 0)}{(0 \times 0) + (1 \times 1) + (0 \times 0)} \right)}{2} \\ &= 0.2 \end{aligned}$$

## Refinement of the Rule Set

The intuition behind this rule refinement process is that increasing interclass distance creates more accurate rules. Rules are modified in parallel, where each iteration provides a set of locally promising modifications. Promising modifications are selected by comparing the conditions of a rule to its most similar appropriate sequence. Rules are modified by removing the temporal overlapping between a similar condition  $c$  and episode  $e$ , resulting in a modified condition  $c'$  being either entirely removed or subsumed by  $c$ .

For example, modifying the conditions of Figure 6 according to episodes  $e_1$ ,  $e_2$  and  $e_3$  reduces the time interval of  $c_1$  from  $[0, 5[$  to  $[3, 5[$  and completely removes  $c_2$ . Consequently, the distance between these subsequences increases from from 0.2 to 1.

	Dataset	Episode	Seq.	Inappr.	Attr.
METRO	Training	9,176	132	12	1206
	Test	19,182	278	46	
TAZO	Training	37,428	485	190	1581
	Test	68,188	947	413	

Table 1: Description of two datasets used in our experiments

## Preliminary Results for the Learning Capability

For a preliminary evaluation of the learning capability, we tested APSS with learning rules that identify an “early switch therapy”. Early switch therapy is a key intervention in ATM prescribing where an intravenous ATM is replaced by an oral ATM providing a less costly alternative and allowing the patient to be discharged earlier. A clinically valid recommendation requires the following three indications: 72 consecutive hours of intravenous therapy, 48 hours of stabilized state of health (e.g., normal levels of temperature and white cell count), and 24 hours of concurrent oral therapy.

The rules for recognizing patients who were eligible for an early switch involve non trivial temporal constraints, making them a good test case for the learning algorithm. In this experiment, this rule is not specified. The dataset only contains inappropriate and appropriate labels specifying if a hospitalization contains or not a recommendation for early switch therapy. The objective is to demonstrate that the rule is eventually learned from these alerts.

We created two datasets of different sizes and ratios of inappropriate sequences. The first was created with patients who received piperacillin-tazobactam (TAZO), our centre’s most prescribed intravenous ATM. We created a smaller dataset with patients who received metronidazole (METRO), an ATM predominantly prescribed orally. They were partitioned into *training* and *test* datasets. Table 1 describes their number of episodes, sequences, inappropriate sequences, and attributes. APSS preprocessed these datasets in 121.9 seconds and 6.9 seconds, respectively.

TIM extracted an accurate and sensitive set of 35 rules. While precision was lower, it remained above APSS without TIM. Rules were presented to an infectious diseases specialist who evaluated their clinical relevance using a five-point Likert scale ranging from 1-no relevance to 5-excellent relevance. Excellent relevance required the presence of all three indications for early switch therapy. Figure 7 presents the scores; 63% of the rules were found to be clinically relevant (score  $\geq 3$ ). Interestingly, rules with high relevance scores also had the highest information content (*J-measure*). On the other hand, rules with a relevance score of 1 were very specific and covered less than 1% of the test set. Removal of these rules from the rule set leads to little loss in alert coverage and accuracy.

Our expert used clinical information not associated to the three previous indications to identify patient profiles associated with early switch therapy. Consider the conditions of the rule in Figure 8 with a relevance score of 5. Noting  $C$  the conjunction of these conditions, we have the rule  $C \rightarrow \text{inappropriate}$ . Direct indications for early switch

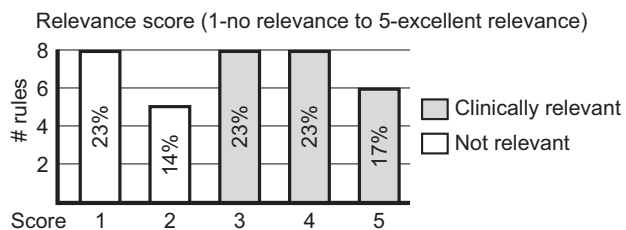


Figure 7: Relevance score of 35 extracted rules

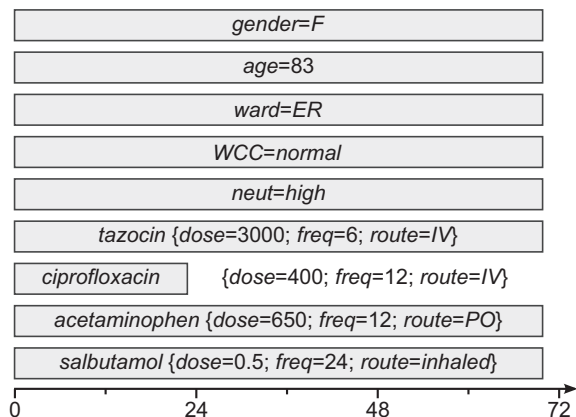


Figure 8: Conditions of a rule with relevance score of 5

therapy are respected with prolonged intravenous (*IV*) treatment, normal levels of white cell count (*WCC*), and concurrent oral treatment. Moreover, prolonged stay at the emergency room (*ER*), old age, *salbutamol*, and additional ATM coverage with *ciprofloxacin* may indicate suspicion of pneumonia caused by resistant pathogens. Ten rules targeted patients under post-operative ATM prophylaxis, a practice not supported by medical evidence that will be addressed by our stewardship program. Another finding was that eight rules targeted patients with  $BMI \geq 40$ . It could suggest that extended intravenous treatments are prescribed for very severely obese patients to ensure targeted concentrations are achieved. These patient profiles provide insight into our centre’s prescribing practices and are of high interest for further investigation, as they identify subgroups of patients that could require closer monitoring or wards that could benefit from targeted in-service training.

We also compared TIM to three well-known learning methods for this type of problem. Table 2 reports their respective number of rules, computation time, precision, recall, and accuracy. The first method used retrieval-only instance-based learning (IBL) where every known inappropriate sequence is used as a rule. The second method (CRL) used CN2’s (Clark and Niblett 1989) general-to-specific search where individual rules are created by iteratively selecting the “best” condition. Conditions are added until they no longer improve the rule’s *J-measure* or until every inappropriate instance is covered. The third method used an association rule mining (ARM) approach based on *Apriori* (Agrawal and Srikant 1994). Various strategies were

	Algo.	#rules	Time (s)	Prec.	Rec.	Acc.
METRO	TIM	5	0.4	53.8	76.1	85.3
	IBL	12	0.2	30.1	95.7	62.6
	CRL	1	46.0	56.5	76.1	86.3
	ARM	8,074	15.2	44.1	32.6	82.0
TAZO	TIM	30	73.5	62.5	99.0	73.7
	IBL	190	9.5	59.3	99.3	70.0
	CRL	6	583.1	71.4	88.1	79.4
	ARM	614,652	17,864.5	66.0	81.4	73.6

Table 2: Comparison of four algorithms on two datasets

used in CRL and ARM to focus on highly predictive rules for the inappropriate class. For example, ARM used candidate pruning on both support (METRO:  $supp \geq 0.015$ ; TAZO:  $supp \geq 0.02$ ) and confidence ( $conf \geq 0.75$ ), and eliminated dominated patterns (Zaki, Lesh, and Ogihara 2000). We restricted ARM to rules of size 4 for the TAZO test.

Overall, TIM achieved relatively similar or better recall and accuracy than CRL and IBL, except for the recall metrics in METRO. IBL succeeded in classifying correctly most unseen inappropriate sequences in both tests. However, the wide coverage of its rules also incorrectly classified several appropriate sequences, penalizing greatly its precision and accuracy. In contrast, CRL achieved good precision and accuracy on both datasets, yet by creating fewer rules for classifying sequences. However, they identified fewer inappropriate sequences in both tests. On the other hand, TIM combines the strengths of both previous methods. Performing a specific-to-general search and modifying every rule in parallel according to appropriate sequences speeds up the process, enabling TIM to outperform CRL by up to two orders of magnitude. TIM in addition better succeeds in identifying inappropriate sequences with equal or higher recall, without sacrificing much accuracy. Furthermore, TIM harnesses CRL’s ability to extract fewer rules than IBL. ARM performed poorly, being 30 to 200 times slower than TIM and producing much more rules, requiring heavy post-processing to identify a subset of accurate rules.

## Conclusion and Future Work

In this paper, we presented APSS, a clinical decision support system that evaluates antimicrobial (ATM) prescriptions and produces alerts for potentially inappropriate ones. Since its deployment in August 2010, APSS has been met by prescribers and decision makers with universal appreciation and recognition. We also presented an emerging machine learning capability for APSS. The learning capability combines instance-based learning and rule induction to learn prescription classification rules from user feedback.

We discussed preliminary results demonstrating the rule-learning capability for appropriate early switch from intravenous to oral ATM therapy. A majority of learned rules were found to be clinically relevant because they succeeded in identifying the clinical indications for early switch therapy. From these rules, a clinician identified patient profiles associated with early switch recommendations provid-

ing further insight into our centre's prescribing practices and a potential for targeted interventions (e.g., unsupported use of post-operative antimicrobial prophylaxis).

The next step is to pursue the experimentation of the learning capability before its release in the deployed version of APSS. Users will then be able to utilize the learning module to explore different rule sets and keep the rules they find clinically relevant and accurate. Learning from imbalanced data sets, where there are many more instances of some classes than others, is an important issue in domains such as ours, where inappropriate prescriptions are more the exception than the norm. Although the preliminary results of our algorithm seem encouraging, we have not yet characterized the algorithm with respect to the imbalanced data problem. This is on our agenda for future work. Other methods of temporal data mining could be integrated to the knowledge management tools in order to explore the vast quantity of data that we are accumulating and identify interesting patterns (i.e., repetitive behaviors of interest) that could be further investigated by our ATM stewardship team.

In the meantime, we are in the process of exporting APSS in other centres where we believe it will help reduce inappropriate ATM prescribing and improve patient health. The revision process used by APSS could also be adapted to other drugs since it already manages the patients' prescriptions, vital signs, and laboratory and microbiology test results.

### Acknowledgments

This project was partially funded by the *Fonds de recherche du Québec – Santé*, the *Fonds de recherche du Québec – Nature et technologies*, the *Natural Sciences and Engineering Research Council of Canada*, and the *Centre Hospitalier Universitaire de Sherbrooke*.

### References

- Aamodt, A., and Plaza, E. 1994. Case-based reasoning: Foundation issues, methodological variation and system approaches. *AI Communications* 7(1):39–59.
- Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th Int. Conf. on Very Large Data Bases, VLDB '94*, 487–499. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Aha, D. W.; Kibler, D.; and Albert, M. K. 1991. Instance-based learning algorithms. *Mach. Learn.* 6(1):37–66.
- Bellazzi, R.; Larizza, C.; Magni, P.; and Bellazzi, R. 2005. Temporal data mining for the quality assessment of a hemodialysis service. *Artif. Intell. Med.* 34(1):25–39.
- Chawla, N.; Japkowicz, N.; and Kotcz, A. 2004. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter* 6(1):1–6.
- Clark, P., and Niblett, T. 1989. The cn2 induction algorithm. *Mach. Learn.* 3(4):261–283.
- Concaro, S.; Sacchi, L.; Cerra, C.; Fratino, P.; and Bellazzi, R. 2009. Mining healthcare data with temporal association rules: Improvements and assessment for a practical use. In Combi, C.; Shahar, Y.; and Abu-Hanna, A., eds., *AIME 2009*, volume 5651 of *LNCS*. Springer Berlin Heidelberg, 16–25.
- Dellit, T. H.; Owens, R. C.; McGowan, J. E.; Gerding, D. N.; Weinstein, R. A.; Burke, J. P.; Huskins, W. C.; Paterson, D. L.; Fishman, N. O.; Carpenter, C. F.; Brennan, P. J.; Billeter, M.; and Hooton, T. M. 2007. Infectious diseases society of america and the society for healthcare epidemiology of america guidelines for developing an institutional program to enhance antimicrobial stewardship. *Clin. Infect. Dis.* 44(2):159–177.
- Domingos, P. 1996. Unifying instance-based and rule-based induction. *Mach. Learn.* 24(2):141–168.
- Gyssens, I. C.; van den Broek, P. J.; Kullberg, B. J.; Hekster, Y.; and van der Meer, J. W. 1992. Optimizing antimicrobial therapy. a method for antimicrobial drug use evaluation. *J. Antimicrob. Chemother.* 30(5):724–7.
- Hartge, F.; Wetter, T.; and Haefeli, W. E. 2006. A similarity measure for case based reasoning modeling with temporal abstraction based on cross-correlation. *Comput. Methods Programs Biomed.* 81(1):41–8.
- Hsieh, T. C.; Kuperman, G. J.; Jaggi, T.; Hojnowski-Diaz, P.; Fiskio, J.; Williams, D. H.; Bates, D. W.; and Gandhi, T. K. 2004. Characteristics and consequences of drug allergy alert overrides in a computerized physician order entry system. *J. Am. Med. Inform. Assoc.* 11(6):482–91.
- Montani, S.; Portinale, L.; and Leonardi, G. 2006. Case-based retrieval to support the treatment of end stage renal failure patients. *Artif. Intell. Med.* 37(1):31–42.
- Pharmacy OneSource. Senti7. <http://www.senti7.com/ID/>. Accessed: 2013/01/14.
- Premier Inc. Safetyurveillor - pharmacy. <https://www.premierinc.com/informatics/tools-services/performance-suite/pharmacy.jsp>. Accessed: 2013/01/14.
- Reichley, R. M.; Seaton, T. L.; Resetar, E.; Micek, S. T.; Scott, K. L.; Fraser, V. J.; Dunagan, W. C.; and Bailey, T. C. 2005. Implementing a commercial rule base as a medication order safety net. *J. Am. Med. Inform. Assoc.* 12(4):383–9.
- Shahar, Y. 1997. A framework for knowledge-based temporal abstraction. *Artif. Intell.* 90(1-2):79–133.
- Smyth, P., and Goodman, R. M. 1991. Rule induction using information theory. In Piatetsky-Shapiro, G., and Frawley, W. J., eds., *Knowledge Discovery in Databases*, 159–176. AAAI/MIT Press.
- Valiquette, L.; Cossette, B.; Garant, M. P.; Diab, H.; and Pepin, J. 2007. Impact of a reduction in the use of high-risk antibiotics on the course of an epidemic of clostridium difficile-associated disease caused by the hypervirulent nap1/027 strain. *Clin. Infect. Dis.* 45 Suppl 2:112–121.
- Zaki, M.; Lesh, N.; and Ogihara, M. 2000. Planmine: Predicting plan failures using sequence mining. *Artif. Intell. Rev.* 14(6):421–446.