J Med Syst (2012) 36:3861–3874
DOI 10.1007/s10916-012-9859-6

ORIGINAL PAPER

Using Machine Learning Classifiers to Assist Healthcare-Related Decisions: Classification of Electronic Patient Records

Juliana T. Pollettini · Sylvia R. G. Panico ·
Julio C. Daneluzzi · Renato Tinós ·
José A. Baranauskas · Alessandra A. Macedo

Received: 15 September 2011 / Accepted: 30 April 2012 / Published online: 18 May 2012

Abstract Surveillance Levels (SLs) are categories for medical patients (used in Brazil) that represent different types of medical recommendations. SLs are defined according to risk factors and the medical and developmental history of patients. Each SL is associated with specific educational and clinical measures. The objective of the present paper was to verify computer-aided, automatic assignment of SLs. The present paper proposes a computer-aided approach for automatic recommendation of SLs. The approach is based on the classification of information from patient electronic records. For this purpose, a software architecture composed of three layers was developed. The architecture is formed by a classification layer that includes a linguistic module and machine learning classification modules. The classification layer allows for the use of different classification methods, including the use of preprocessed, normalized language data drawn from the linguistic module. We report the verification and validation of the software architecture in a Brazilian pediatric healthcare institution. The results indicate that selection of attributes can have a great effect on the performance of the system. Nonetheless, our automatic recommendation of surveillance level can still benefit from improvements in processing procedures when the linguistic module is applied prior to classification. Results from our efforts can be applied to different types of medical systems. The results of systems supported by the framework presented in this paper may be used by healthcare and governmental institutions to improve healthcare services in terms of establishing preventive measures and alerting authorities about the possibility of an epidemic.

Keywords Medical informatics · Medical surveillance · Health service · Computer aided decision system · Machine learning classifiers

Introduction

The Brazilian healthcare network is based on both private and public services. The public sector, the Unified Healthcare System (SUS—Sistema Único de Saúde), is a state-controlled public healthcare system created to sanction universal access to healthcare in Brazil. The SUS is organized into a regional, hierarchical and complex network of services provided by the federal government, and state and city governments. Healthcare services are categorized in three levels of complexity: primary or basic healthcare (preventive care, primary healthcare, and disease control); secondary care or specialized, medium-complexity healthcare; and tertiary care, or high- complexity healthcare. A healthcare system whose organization begins by focusing on primary care can theoretically improve the chances for superior performance by the healthcare services network. A system with primary healthcare at its foundation can improve its performance because it allows for: (1) better fit between patient management and types of health problems patients

J. T. Pollettini · R. Tinós · J. A. Baranauskas · A. A. Macedo
Department of Computer Science and Mathematics, FFCLRP,
University of São Paulo,
Ribeirão Preto, São Paulo, Brazil 14040-901

S. R. G. Panico · J. C. Daneluzzi
Department of Pediatrics, FMRP, University of São Paulo,
Ribeirão Preto, São Paulo, Brazil 14049-900

A. A. Macedo (✉)
Av. Bandeirantes, 3900, Monte Alegre,
Ribeirão Preto, São Paulo, Brazil CEP 14040-90
e-mail: ale.alaniz@usp.br

present with at their first contact with health services; and (2) rationalization of the use of healthcare resources.

Primary healthcare professionals routinely have direct contact with patients and their families. These professionals can be considered the earliest and most up-to-date source of information about the health and evolution of Brazilian families. It is critical that primary healthcare professionals obtain useful information about the processes associated with the evolution of the health of families. This information can be used to support preventive healthcare measures and improve the well-being of individuals and society. In the field of pediatric care, for example, preventive measures are a vital piece of the process of ensuring the quality of child development. Primary healthcare professionals should be able to readily identify children who are at risk and prescribe the necessary intervention to minimize healthcare problems [1]. Healthcare professionals should also be able to identify factors that help improve the healthcare of children [1]. The application of computer-aided technologies to data mine healthcare information can help improve healthcare practices and procedures; for example, these technologies can help to identify children with developmental problems.

A Brazilian Interdisciplinary Research and Teaching Group carried out surveys with the objective of systematizing the procedures for follow-up and healthcare of patients treated in primary healthcare facilities. The group proposed a specific measurement to identify patient healthcare needs, called the Surveillance Level (SL). The SL indicates the type of healthcare procedure and service needed; it is an assessment of the follow-up of patients and of healthcare services in primary healthcare [2]. SL can be used to inform the recommendation of pediatric procedures in primary healthcare. It identifies significant risk factors and protective factors associated with patients and their families. Different SLs levels are associated with general and specialized educational or therapeutic measures, according to the following scale [2]:

- SL-Routine: routine measures by the primary healthcare facilities;
- SL-1: educational measures;
- SL-2: educational and therapeutic measures;
- SL-3: strong need for educational and specialized therapeutic measures;
- SL-Emergency: patient requires urgent care.

SL level assignment is based on patient information and updated after each medical appointment. It is a laborious task that demands personnel training and personalized evaluation of each patient. Consequently, computer-aided recommendation of SL based on electronic patient records is an innovative effort in the medical field. It can support (i) the work and decision-making processes of healthcare professionals (it provides an effective and fast second opinion about patient SL and an alternative course of action); and (ii) healthcare epidemic surveillance systems (it can identify possible outbreaks and the occurrence of systematic healthcare problems).

In this paper we present a software architecture-based framework that supports automatic recommendation of SLs based on the analysis and classification of patient information. It is a machine-learning architecture divided into presentation, classification and storage layers. The classification layer is a software framework composed of a linguistic module and classification modules. Each classification module corresponds to machine-learning and information retrieval classification methods that process patient information and automatically assign SLs. The linguistic module preprocesses text information using a thesaurus to find language patterns in medical history. Here we will advocate the use of machine learning classifiers to recommend SL scores for patient records. We will also argue in favor of text processing resources to enhance classification performance (automatic assignment of SL). The linguistic resources are based on multi-disciplinary healthcare-related vocabulary. The proposed architecture was implemented and validated using pediatric information from a primary healthcare facility. The results show how to use classifiers to produce healthcare measure recommendations, such as SL. The results also indicate the advantages of applying automated recommendation of SL based on classification of Electronic Patient Records.

The software architecture was implemented in a community medical center called Vila Lobato. The center is a primary healthcare facility that has been providing clinical, teaching, and research services for almost 40 years. Vila Lobato has an established tradition in child and adolescent healthcare. Most Brazilian primary healthcare institutions provide healthcare services only for patients who present with symptoms or disease. Vila Lobato routinely schedules preventive healthcare appointments. The Surveillance Level (SL) system was adopted by Vila Lobato to support their healthcare program and services. One of the goals of the institution is to foster a close relationship between multidisciplinary healthcare teams. The application of computer-aided SL assignment may help to further this relationship by assisting healthcare workers in their decisions and in reassessing recommendations as a team. The automatic assignment of SL at Vila Lobato or other institutions is our initiative.

Architecture

Computational techniques classify, retrieve, share, and manage healthcare-related information; these techniques can be used, for example, to recognize patterns and uncover evidence for evidence-based medical practices. The present paper describes software architecture with classification modules supported by a linguistic module. The objective was to provide pattern-based classification of patients into Surveillance Levels (SLs) based on the analysis of patient information. Figure 1 shows the software architecture, which is organized into three layers: (i) *Presentation Layer*, the graphical user interface (Fig. 1a); (ii) *Classification Layer*, a module-based software framework organized into six modules (five classification modules and a linguistic module responsible for preprocessing language data) (Fig. 1b); and (iii) *Storage Layer*, a layer that processes information based on patient medical history and allows inclusion of patient information (Fig. 1c). Each layer is described in turn in the following sections.

Presentation layer

The presentation layer supports the following functionalities: (i) recommendation of SLs as a second opinion for healthcare professionals (this function can be used, for example, during appointments); (ii) SL update; (iii) map projections of georeferenced SL; (iv) reevaluation of SL-3 or SL-Emergency recommendations; (v) software setup (for example, classifier parameters); and (vi) visual representation of the information generated by the classifiers, including rank-accuracy of classifiers. An Electronic Patient Record (EPR) system is used as the data set that supports the automatic classification system; it provides critical information such as personal patient information, exam and diagnostic results, medical procedures and others. Figure 2a shows the user interface and its functionalities, as implemented in our prototype system. (An additional function is not represented in the user interface: the control of access according to user profiles. Healthcare professionals can be assigned different levels of access to the information stored in the application). Classifier parameters in the presentation layer can be modified by an administrator to optimize classification performance. The setup interface presents five choices of pattern-based classifiers and the linguistic module. The administrator may choose the best classifier based on a comparison of results from different classifiers (Fig. 2b). The administrator should be a professional with knowledge on informatics and/or statistics; the administrator should, preferably, have experience with health sciences, e.g., a biomedical informatics or health informatics professional.

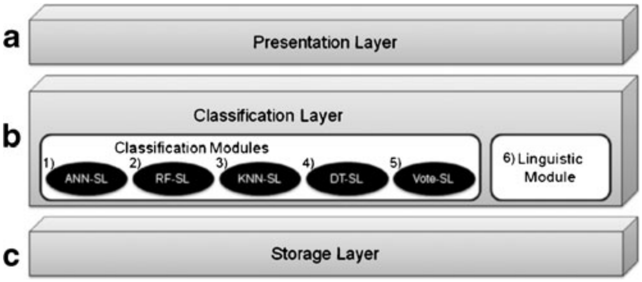


Fig. 1 System architecture organized into three layers: Presentation Layer, Classification Layer, and Storage Layer. The Classification Layer has five classification modules: KNN-SL, ANN-SL, RF-SL, DT-SL and Vote-SL and one linguistic module

The presentation layer is also responsible for maintaining the surveillance system operational. After patient information is entered, the classification layer assigns an SL for the patient. In order to achieve an SL recommendation, as shown in Fig. 3a, a healthcare professional provides the patient data and sends the information to the classifier, which was previously selected by the system administrator. The professional then receives the SL recommendation. The healthcare professional can also access a map projection of georeferenced information based on patient SLs addresses [3]. Figure 3b shows an example of the map projection of SLs and fictitious patient addresses. The application Google Maps was used to generate the map projection of addresses of all patients recently assigned an SL-3 level. The SLs are georeferenced based on patient addresses in the EPR system. The map projection can assist, for example, the identification of epidemic outbreaks by healthcare and government institutions. The presentation algorithm can be used to establish a map of any type of SLs.

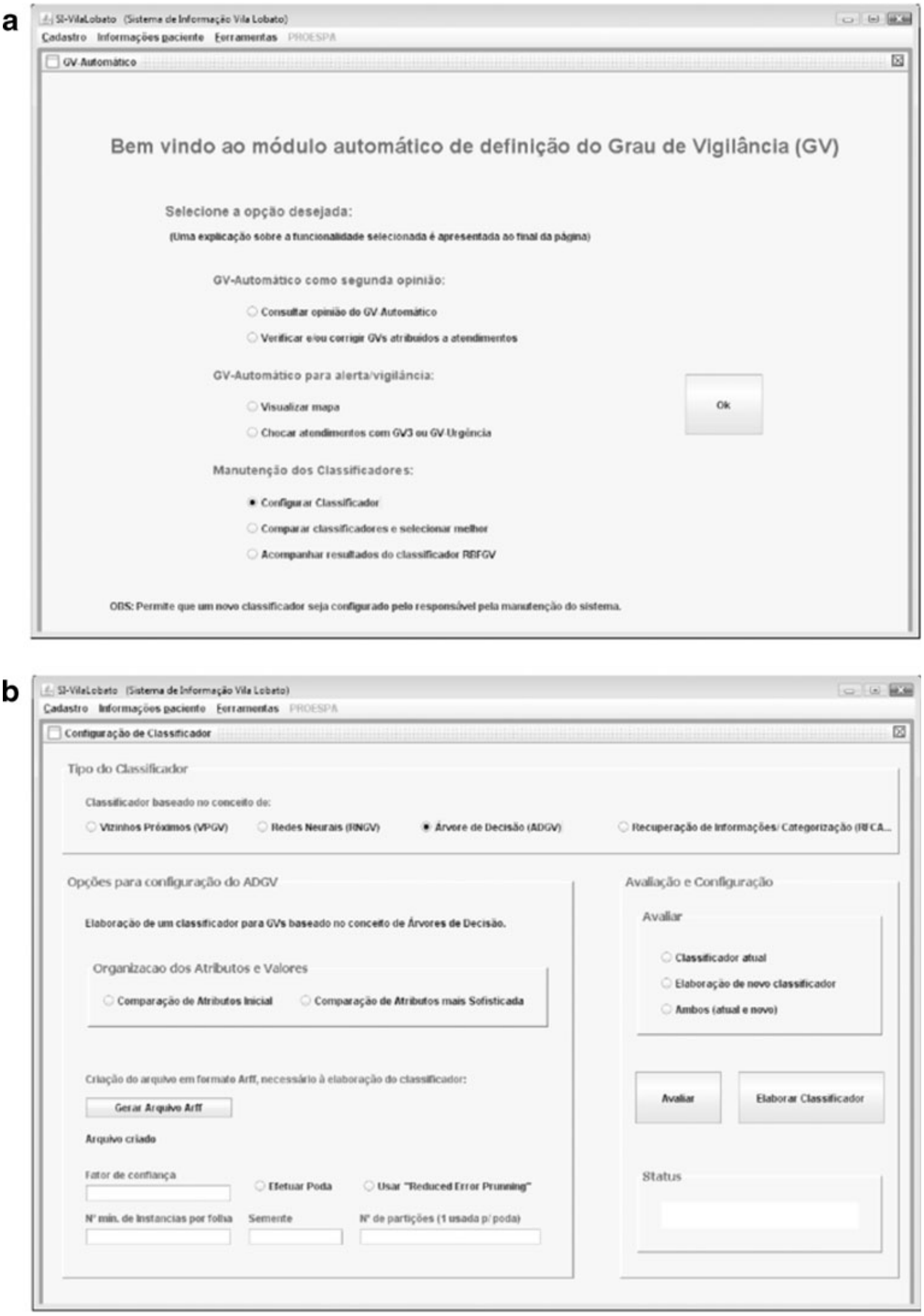
There are different visual representations for SLs assigned by healthcare professionals and those assigned automatically by the software. The display of georeferenced SL is a result of an integration of Geographic Information System (GIS) and EPR. This integration uses Extensible Markup Language (XML), Document Object Model (DOM), Java, JavaScript and AJAX. The Google Maps application was used as a map server. Google Maps plot georeferenced data from a specific archive based on data from the EPR system.

Storage layer

The storage layer (Fig. 1c) is composed of three software classes which are responsible for processing information based on patient medical history stored in a database. A **MedicalAppointmentSummary** class is responsible for accessing the EPR database to obtain information required for classification procedures. A **SurveillanceLevel** class applies basic methods to activate the pertinent classifier. Finally, the **OrganizeArff** class gathers patient information based on attributes and values that will be used by the classification module.

Classification layer

The classification layer is a software framework composed of a linguistic module and classification modules. Each classification module corresponds to machine-learning and information retrieval classification methods. The pattern-


a
SI-VilaLobato (Sistema de Informação Vila Lobato)
Cadastro Informações paciente Ferramentas PROESPA
GV-Automático
Ok
b
SI-VilaLobato (Sistema de Informação Vila Lobato)
Cadastro Informações paciente Ferramentas PROESPA
Configuração de Classificador
Gerar Arquivo Arff
Avaliar
Elaborar Classificador
Status

Bem vindo ao módulo automático de definição do Grau de Vigilância (GV)

Selecione a opção desejada:

(Uma explicação sobre a funcionalidade selecionada é apresentada ao final da página)

GV-Automático como segunda opinião:

○ Consultar opinião do GV Automático
○ Verificar e/ou corrigir GVs atribuídos a atendimentos

GV-Automático para alerta/vigilância:

○ Visualizar mapa
○ Chocar atendimentos com GV3 ou GV Urgência

Manutenção dos Classificadores:

● Configurar Classificador
○ Comparar classificadores e selecionar melhor
○ Acompanhar resultados do classificador RBFGV

OBS: Permite que um novo classificador seja configurado pelo responsável pela manutenção do sistema.

Tipo do Classificador

Classificador baseado no conceito de:

○ Vizinhos Próximos (VPGV) ○ Redes Neurais (RNGV) ● Árvore de Decisão (ADGV) ○ Recuperação de Informações/ Categorização (RFCA...

Opções para configuração do ADGV

Elaboração de um classificador para GVs baseado no conceito de Árvores de Decisão.

Organizacao dos Atributos e Valores

○ Comparação de Atributos Inicial ○ Comparação de Atributos mais Sofisticada

Criação do arquivo em formato Arff, necessário à elaboração do classificador:

Arquivo criado

Fator de confiança ○ Efetuar Poda ○ Usar "Reduced Error Prunning"

N° min. de Instancias por folha Semente N° de partições (1 usada p/ poda)

Avaliação e Configuração

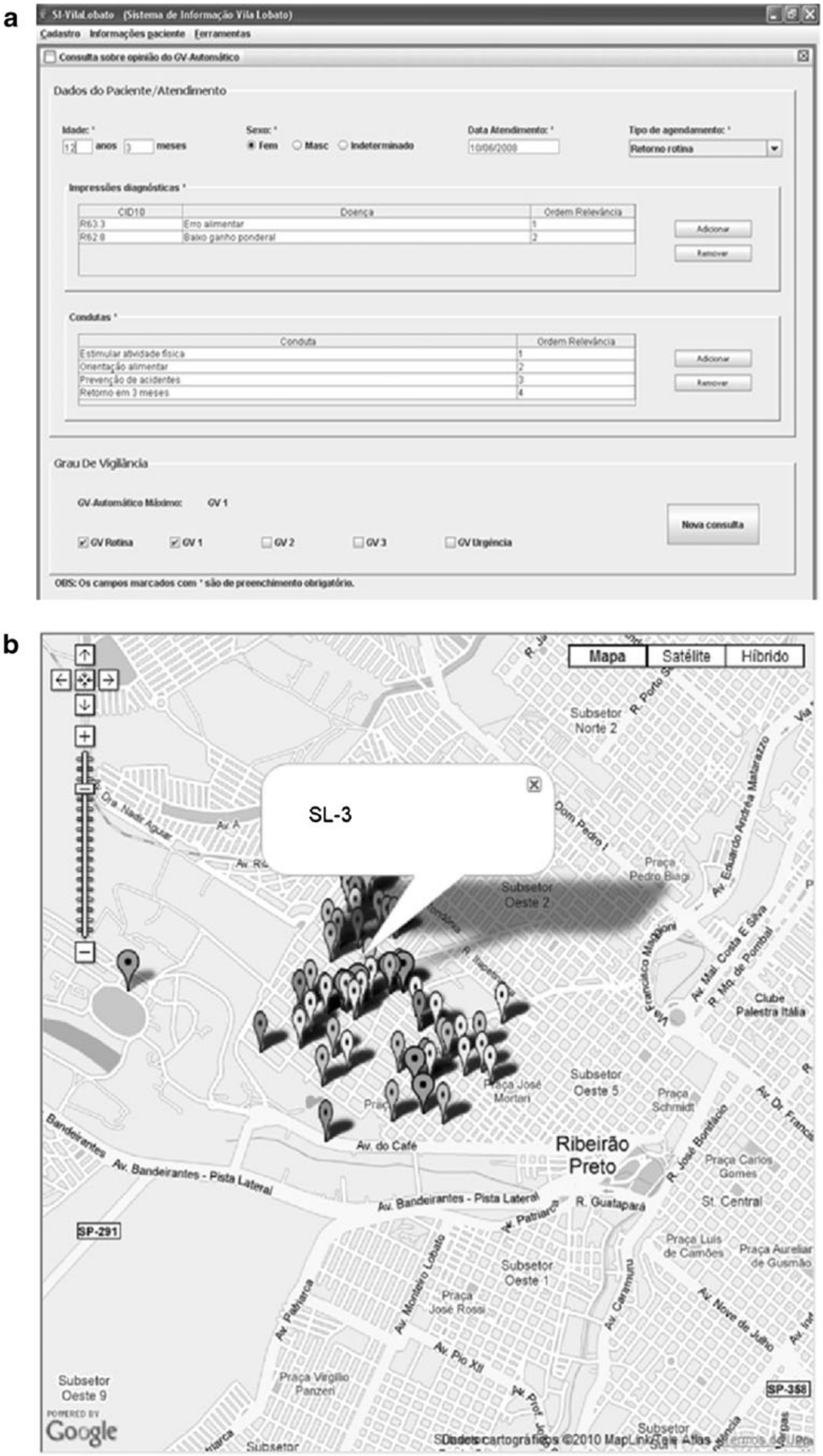
Avaliar

○ Classificador atual
○ Elaboração de novo classificador
○ Ambos (atual e novo)

Fig. 2 User interface supported by the presentation layer: (a) Interface main functionalities; (b) Setup (configuration) interface

based classification of patients is performed based on patient electronic records. The classification generates the SL recommendation that may be used by healthcare professionals.

The classification layer is composed of one linguistic module and different classification modules that apply varied classification techniques, namely: (i) K-nearest-neighbor (KNN-SL), (ii) Artificial Neural Network (ANN-SL), (iii) Relevance Feedback (RF-SL), (iv) Decision Tree (DT-SL), and (v) an ensemble classifier (Vote-SL). Classification algorithms in the classification layer are powerful tools. However, theoretical and empirical results show there is not a single algorithm that uniformly achieves the best performance in all domains. In this case, the best scenario can be obtained (a) by applying several learning algorithms, and picking up the best one for the problem at hand (classification techniques (i)–(iv) above) or (b) combining several models into an ensemble classifier (classification technique (v) above) [4–7].



SI-VilaLobato (Sistema de Informação Vila Lobato)

Cadastro Informações paciente Ferramentas

□ Consulta sobre opinião do GV-Automático

Dados do Paciente/Atendimento

Idade: * 12 anos 3 meses

Sexo: * ● Fem ○ Masc ○ Indeterminado

Data Atendimento: * 10/06/2008

Tipo de agendamento: * Retorno rotina

Impressões diagnósticas *

CID10	Doença	Ordem Relevância
R63.3	Erro alimentar	1
R62.8	Baixo ganho ponderal	2

Condutas *

Conduta	Ordem Relevância
Estimular atividade física	1
Orientação alimentar	2
Prevenção de acidentes	3
Retorno em 3 meses	4

Grau De Vigilância

GV-Automático Máximo: GV 1

☑ GV Rotina ☑ GV 1 □ GV 2 □ GV 3 □ GV Urgência

OBS: Os campos marcados com * são de preenchimento obrigatório.

Fig. 3 (**a**) Recommendation of SLs as a second opinion for healthcare professionals; (**b**) Fictitious map projection of georeferenced patient SL in the vicinity of a primary healthcare facility

The artificial neural network (ANN-SL), the relevance feedback (RF-SL), the K-nearest-neighbor (KNN-SL), and the decision tree (DT-SL) modules are represented, respectively, in Fig. 1b.1, 1b.2, 1b.3 and 1b.4. Moreover, Fig. 1b.5 shows the Vote-SL ensemble classifier which combines all four classification techniques. ANN-SL, KNN-SL, DT-SL and Vote-SL use respectively the following modules from the open source software Weka [8]: *Multilayer-Perceptron*, *IBk, J48* and *Vote a*lgorithms. DT-SL also uses the *Random-Forest* algorithm.

Figure 4 shows the classes used for automatic recommendation of SLs based on the five classification modules and other classes. For example, the **OrganizeArff** class manipulates the *ARFF* file containing all attributes and instances selected from electronic patient records. The ARFF file (Attribute-Relation File Format) is a text file with a list of instances that share a set of attributes used as input for all classification modules. This kind of file was developed for use with the Weka machine learning software [9].

The KNN-SL calls the Weka classifier **IBk** constructor, indicating the number of *k* nearest neighbors. The KNN-SL training set and test set use the **Instances** class from the **weka.core** package to represent the file containing the attributes from patient appointments (Fig. 4). KNN-SL also sets up SL as the classification label (the class attributes, for example, the feature to be learned in machine learning terminology). The algorithm calls a method from the **IBk** class and serializes the instantiated classifier using a method from the **SLClassifierSettings** class to save the individual fields into a file. For new medical appointment data, the KNN-SL module retrieves the saved classifier and runs the classification methods.

The ANN-SL module is similar to KNN-SL in terms of accessing external classes and the ARFF file. One difference is that KNN-SL uses the **IBk** class and ANN-SL activates methods from **MultilayerPerceptron** class instead. The ANN-SL module applies backpropagation algorithms for training purposes. The DT-SL module is also similar to KNN-SL and ANN-SL modules in that it accesses external classes; however, it calls methods from the **J48** class when a new appointment is entered or when the system administrator sets up the classifiers.

The RF-SL module is different from the three previous modules because it applies text processing prior to classification. As summarized in Fig. 5, initially it discards irrelevant words such as prepositions, articles and others, from patient records. Next, it normalizes text information using the *Porter* stemmer algorithm [10]. After that, RF-SL assigns weights to word stems in medical appointments. The weights are assigned by dividing the word stem frequency (*tf)* in the medical record by the frequency of the stem in the collection (*idf*). The output is an in-memory inverted file of stem terms with their respective weights.

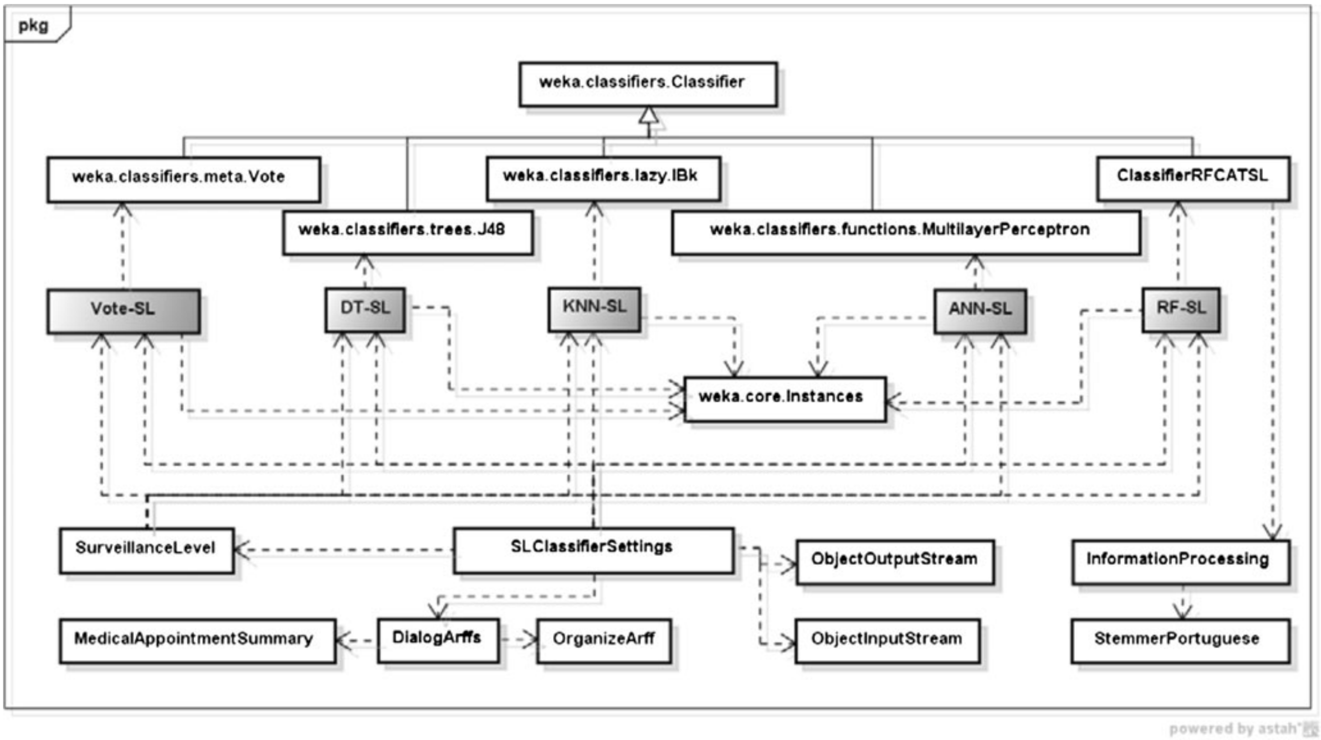


Fig. 4 Classes for the five classification modules

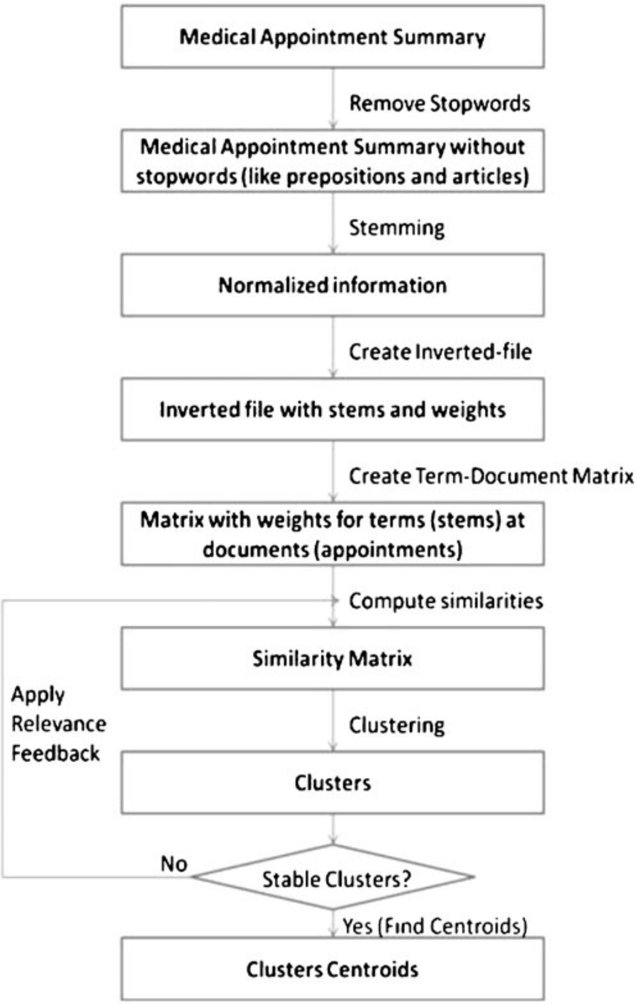

Medical Appointment Summary
Remove Stopwords
Medical Appointment Summary without stopwords (like prepositions and articles)
Stemming
Normalized information
Create Inverted-file
Inverted file with stems and weights
Create Term-Document Matrix
Matrix with weights for terms (stems) at documents (appointments)
Compute similarities
Similarity Matrix
Apply Relevance Feedback
Clustering
Clusters
No
Stable Clusters?
Yes (Find Centroids)
Clusters Centroids

Fig. 5 Fluxogram of the RF-SL module

The inverted file is used to generate a term-document matrix, called matrix X, in which rows represent stem terms and columns represent patient appointments (documents). Cosine measures are calculated to determine similarities among all medical records in matrix X. Next, classification processes begin by forming the first cluster of two comparable medical appointments (a cluster is composed of one or more such comparable appointments). The procedure is then repeated accessing the entire collection of medical records to establish the other clusters, as follows: search for a document with the highest similarity to a document a; once the document is found (document b), there are two possibilities: (i) document b is already in a cluster or (ii) document b is not part of a cluster. If (i), then the document is already in a cluster and document a is added to document *b's* cluster. If (ii), then the document is not part of a cluster and a new cluster is created with both a and b. If the search for similar documents returns empty, according to a specific threshold (for example, document a and b are similar when cos a; $b>$ 0), document a stands alone in a new cluster. Once cluster definition is finished, Rocchio classification [11] is applied to redefine the weights of the stems using the data in each cluster as a collection.

When new medical appointment information is entered into the RF-SL module, cosine measures are calculated for the new records; these measures are compared to the centroids representing all previous clusters. A centroid is basically the "average" of all medical records in a cluster. New medical appointment information will join the cluster with the highest cosine measure between the new document and an existing centroid. The assumption is that the higher the cosine measure, the more comparable the data in the medical appointments are. Each cluster is labeled according to the most frequent type of SL in that cluster. Figure 5 presents a fluxogram that summarizes this process. Text preprocessing allows the RF-SL classifier to use attributes that were used by other classifiers and new attributes. This classifier parses the unique ARFF file to select the attributes. In this case, the IBk, J48 and MultilayerPerceptron algorithms from the **weka.classifiers** were modified to allow for selection of attributes.

An ensemble module called Vote-SL was created by combining all previous classification modules. Therefore, new versions of IBk, J48 and MultilayerPerceptron classifiers were created to support Vote-SL.

Ensembles of classification modules: Vote-SL

The goal of the Vote-SL module was to improve the accuracy of individual classifiers. An ensemble h^* consists of a set of L individual classifiers $\{h_1, h_2, \ldots, h_L\}$ whose predictions are combined to label a new instance. In other words, for a problem with k class labels $\{C_1, C_2, \ldots, C_k\}$ a combination is performed by means of a majority vote, given by Eq. 1. In a majority vote, the classifier h^* will output the label with the highest frequency given by all classification modules $\{h_1, h_2, \ldots, h_L\}$. In the case of regression, h^* is usually an average of values obtained by each individual hypothesis given by Eq. 2.

$$h^*(x) = \underset{c \in \{C_1 C_2, \ldots C_k\}}{\arg\max} \sum_{l=1}^{L} \|h_l(x) = c\| \tag{1}$$

$$h^*(x) = \frac{1}{L} \sum_{l=1}^{L} h_l(x) \tag{2}$$

The ensemble is frequently more precise than any one hypothesis. The use of multiple hypotheses (for classification or regression) shows superior accuracy when labeling examples that were not in the training set. For example, suppose that an ensemble is composed of three classifiers $\{h_1, h_2, h_3\}$ and there is a new example x to be labeled; if the three classifiers are identical and an error is made in $h_1(x)$, that error will persist in $h_2(x)$ and $h_3(x)$. However, by using multiple hypotheses, if $h_1(x)$ is incorrect, $h_2(x)$ and $h_3(x)$ may still perform the correct classification. In these situations, the majority vote is applied and a new example x will be classified correctly by the ensemble. (If the errors of L classifiers are independent and less than ½, then the likelihood that a majority vote is wrong is given by the area above a binomial distribution in which $L/2$ classifiers are wrong).

The Vote-SL module uses the majority vote classifier from **weka.classifiers.meta**. The class **Vote,** used by this module, combines classifiers using unweighted averages of probability estimates (classification) or numeric predictions (regression). Using **meta**.**Vote,** it is possible to select several classifiers or any one classifier from a list, produced by **weka.classifiers** and user-defined classifiers. Vote-SL calls a method from the Vote-SL class to input the ARFF file used by all other classifiers. It also uses the labels classified by the other classifiers as votes to compose the final classification. As shown in Fig. 6, the previous classifiers such as KNN-SL and ANN-SL are trained with a training set. These classifiers are then combined by the ensemble to compose the final classification, which is the majority vote among individual classifiers' votes.

Linguistic module

The linguistic module (Fig. 1b.6) attempts to improve classification accuracies by taking into account language characteristics prior to classification of medical records [12]. Medical classification systems usually present a diverse vocabulary because each member of a multidisciplinary healthcare team proceeds to fill out medical records using different specialized vocabulary and writing style.

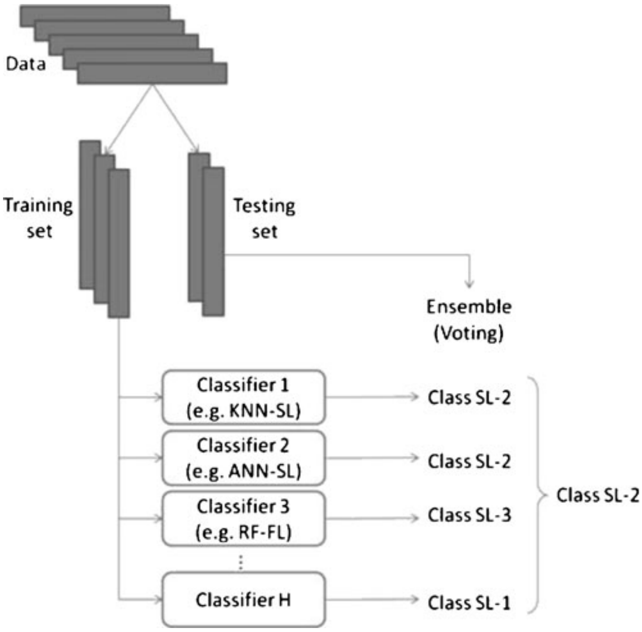


Fig. 6 Fluxogram of the Vote-SL module

The linguistic module aims to normalize medical history data using a more specific vocabulary. We argue that preprocessing text information can help optimize classification accuracy because preprocessing text information can identify and normalize synonyms. For example, a thesaurus can be applied to medical systems to normalize terms associated with medical procedures and diagnoses (ordinary attributes). Normalization can be carried out prior to classification to reduce the number of attributes and enhance the semantic information in multidisciplinary contexts, once language preprocessing decreases the variability of attributes in high-dimensional medical contexts.

The linguistic module has three main classes: **Attributes LinguisticAnalysis**, **AttributePossibleValues** and **Attribute Value**. The first class is given a pattern file that contains attributes and their values. The goal is to organize the machine learning repository of medical data. The attributes can be organized into a pattern file according to user definitions, a thesaurus or a system based on healthcare vocabulary. The **AttributesLinguisticAnalysis** class reads the pattern file, saves the information as objects, accesses patient information from a database, and it automatically standardizes the ARFF file if prompted by the presentation layer. The main attributes used by our linguistic module are primary diagnosis and medical procedure. The **AttributePossibleValues** and **AttributeValue** classes pack information processed by the linguistic module. Currently, the **AttributesLinguistic Analysis** class draws on the International Classification of Diseases-Tenth Edition (ICD-10) to support the standard ARFF file. The ARFF file can be used by modules as training and test sets before new medical appointment information is classified. The ARFF file contains the results generated by the linguistic module, which are used by KNN-SL, ANN-SL, DT-SL and Vote-SL.

The ICD-10 was used to reduce the number of attributes and values to be classified by the modules. The ICD-10 was applied to primary diagnosis information. Each ICD-10 chapter is an attribute to be verified by the classifiers. Each Chapter in the ICD-10 classification of diseases represents a different attribute value. Based on the organization of values and attributes, primary diagnosis information is mapped to a single attribute. A similar approach was applied to medical procedures, using the Unified Medical Language System (UMLS). The use of ICD-10 and UMLS is described in detail elsewhere [12].

Experiments, data and data analysis

This paper is part of a larger project, the VLIS-Project, developed by researchers from Vila Lobato and from the Biomedical Informatics Group at the University of São

Paulo. The VLIS-Project aims to apply biomedical informatics research to advance the current knowledge about healthcare and development of children, adolescents, and their families at Vila Lobato [13]. The VLIS-Project is the umbrella project for three other projects: PROISE [14], PROESPA [15] and Automatic-SL [12, 16, 17]. Of interest to this paper is the Automatic-SL project. The project aims to investigate mechanisms for the automatic classification of patients into surveillance levels at Vila Lobato. The first Automatic-SL version applied K-nearest-neighbor and artificial neural network classifiers [16]. The second version also included a relevance feedback module [12]; finally, the third version included the linguistic module [17].

The Automatic-SL system implemented by the software architecture proposed in this paper received two new modules: a decision tree module and an ensemble classifier. The architecture was validated using digital data from 100 medical records, totaling 534 medical appointments at Vila Lobato. The data were written in Portuguese. We chose records of patients who (i) presented symptoms or disease or (ii) just had preventive healthcare appointments in 2007. These patients were scheduled one or more times during that year. Electronic patient information was automatically classified into SL-Routine, SL-1, SL-2, SL-3, and SL-Emergency levels (see "Introduction" for description of the levels). Next subsections detail metrics, experiments and the data.

Evaluation of accuracy: Cross-validation

To estimate the accuracy of Automatic-SL, we investigated the machine learning algorithms used to classify patients. We applied a stratified 10-fold cross-validation, a widely recommended method for this type of validation [5]. In cross-validation, a sample is partitioned into training and test sets such that the classifier is trained on one set of data and then tested on other, to-be classified data set. The folding procedure (10-fold cross-validation) results in 10 possible folds for classification. A sample is partitioned into ten subsets: nine sets are used for training and the left-out, to-be classified set is the test set.

Our experiments were carried out ten times for each module; each time one subset was left out from the training set and used as the test set. Prior to testing the classification modules, the selection of attributes was set up based on medical history information from the machine learning repository of Vila Lobato data. The information was preprocessed using machine-learning techniques. Subsequently, we performed statistical analyses on the performance of classifiers.

Attribute selection

Relevant attributes were selected by a medical informatics professional who followed the manual assignment of surveillance performed by Vila Lobato healthcare professionals. The medical informatics professional observed the definition of SLs by healthcare workers during regular patient appointments. After several meetings, the following pieces of information were selected as attributes: age, sex, primary diagnosis information, medical procedures, and type of appointment (routine or not). These data were evaluated as critical for the definition of a SL. The attributes were subsequently used to prepare the ARFF file and implement the classification modules.

Data sets

Resampling of the data set was carried out to provide the best learning conditions for the machine-learning algorithms employed by our classification modules. The original data set used in our experiments was unbalanced. As a result, resampling the data was necessary to avoid bias in the classification procedures because some classes had more training data than others. An unbalanced data set indicates that there is a non-uniform distribution of labels. For example, there were 245 examples of SL-2, 160 examples of SL-1, 61 examples of SL-Routine and 68 examples of SL-3. There were no examples of SL-Emergency. As a result, the SL-Emergency level was discarded from the data set. To test our resampling of the data, we ran two experiments, one with the original data set and another using resampled data. The original data set included all examples of surveillance level recommendation at Vila Lobato, and all essential attributes for each classifier, without resampling.

To generate the resampled data set we used a filter, called "Resample" from Weka. The filter provided random resampling of the data with or without replacement. The filter can be adjusted to generate a subsample that maintains the same class distribution of the original data set or to change the class distribution towards uniform distribution. In the present study, we chose the latter. The two resampling parameters were: "**biasToUniformClass**" and "**noReplacement**." The first parameter establishes either that the original class distribution be maintained (value of 0), or that a bias in favor of uniform class distribution should occur in the output data (value of 1). The second parameter (noReplacement) enables (value False) or disables (value True) the replacement of instances. The Resample filter parameters used were: **biasToUniformClass**=1.0 and **noReplacement**=False; in other words, use replacement; generate subsample with the same number of instances of the original data set. The preprocessed subsample was saved as a new data set.

Data analysis

Different experimental designs were developed to search for the best results when comparing the different classifiers in our proposal. The Weka Experiment framework [8] was used to perform statistical analyses. Each data set was tested and comparisons were performed according to the algorithms presented in Table 1.

Two ANN-SL were evaluated with the same parameters and values, but using different algorithms: *MultilayerPerceptron* and *MultilayerPerceptron_JTPJAB*. Seven KNN-SL classifiers were set up using different parameters and IBk and IBk_JTPJAB algorithms. Three DT-SL were set up, two used parameters and values from a **J48** algorithm and the other, parameters from **RandomForest** algorithm, Weka. The RF-SL ran using our **ClassifierRFCATGV** algorithm. The Vote-SLs combined the classifiers and parameters using the **Vote** algorithm. Classifiers were configured as shown in Table 1.

Each of the 17 classifiers was trained, tested, and compared. Figures 6 and 7 show the results and comparisons. KNN-4, KNN-5, KNN-6, and KNN-7 are not represented because they were not considered independent classifiers. These classifiers were, however, part of the Vote-SL (the results of these individual classifiers were worse than their results in Vote-SL).

Paired t-tests are commonly used to test the difference between two classifiers [18]. To compare classifiers, we used Paired T-Tester (corrected for multiple presentations) from Weka Experiment. To assess whether the accuracy of a classifier was significantly different from another ($p<0.05$), a paired *t*-test was performed for each data set and each statistical measure (Percent correct and Area under ROC). T-tests were performed *four* times; *four* is the number of Vote algorithms. In each test, the first classifier (Test base) was compared with the other classifiers.

For Area under ROC (AUC), most studies reported in the literature apply only two classes. For cases with *k* classes, the confusion matrix for ROC calculations becomes complicated. One solution for this problem is to produce *k* different ROC graphs (one graph for each class). Consequently, the calculation of the AUC also changes. In [19], the authors defined the AUC for each class. Next, they calculated the weighted sum privileging prevalent classes. This is the solution adopted by the Weka Experiment framework consequently this solution was used by us.

Table 1 Algorithms, parameters and values used to setup the five classification modules, creating seventeen classifiers using all attributes

Classifier	Algorithm	Parameters & values
ANN-SL (ANN-1)	MultilayerPerceptron	-L0.3 -M0.2 -N500 -V0 -S1 -E20 -Ha
ANN-SL (ANN-2)	MultilayerPerceptron JTPJAB	-L0.3 -M0.2 -N500 -V0 -S1 -E20 -Ha
KNN-SL (KNN-1)	IBk	-K1-W0-A "..LinearNNSearch" -A "..EuclideanDistance"
KNN-SL (KNN-2)	IBk	-K2-W0-I-A "..LinearNNSearch" -A "..EuclideanDistance"
KNN-SL (KNN-3)	IBk JTPJAB	-K1-W0-A "..LinearNNSearch" -A "..EuclideanDistance"
KNN-SL (KNN-4)	IBk	-K1-W0-I-A "..LinearNNSearch" -A "..EuclideanDistance"
KNN-SL (KNN-5)	IBkJ48	-K3-W0-I-A "..LinearNNSearch" -A "..EuclideanDistance"
KNN-SL (KNN-6)	IBk JTPJAB	-K1-W0-I-A "..LinearNNSearch" -A "..EuclideanDistance"
KNN-SL (KNN-7)	IBk JTPJAB	-K1-W0-F-A "..LinearNNSearch" -A "..EuclideanDistance"
DT-SL (DT-1)	J48	-C0.25 -M2
DT-SL (DT-2)	J48 JTPJAB	-C0.25 -M2
DT-SL (DT-3)	RandomForest	-I10 -K0 -S1
RF-SL (RF-1)	ClassificadorRFCatGV	-A1 -B1 -G1 -Rf
Vote-SL (Vote-1)	Vote	-S1 -R AVG Ensemble: IB-1, IB-3,J-1,J-2,RF-1
Vote-SL (Vote-1)	Vote	-S1 -R AVG Ensemble: IB-1,IB-3, J-1,J-2
Vote-SL (Vote-1)	Vote	-S1 -R AVG Ensemble: IB-2, IB-4,IB-4,IB-5,IB-3,IB-6,IB-7
Vote-SL (Vote-1)	Vote	-S1 -R AVG Ensemble: IB-2, IB-4,IB-4,IB-5,IB-3,IB-6, IB-7,J-1,J-2,RF-1

-A = nearestNeighbourSearchAlgorithm (IBk) or Alpha from Rocchio (ClassifierRFCATGV); -B = ClassificadorRFCatGV: Beta from Rocchio algorithm; -C = confidenceFactor (confidence threshold for pruning); -E = ValidationThreshold; -F = DistanceWeighting = Weight by 1-distance (Neighbors will be weighted by their similarity when voting); -G = ClassificadorRFCatGV: Gamma from Rocchio algorithm; -H = HiddenLayers; -I = distanceWeighting: weight by 1/distance (IBk) or number of trees in the forest (RandomForest); -K = number of nearest neighbours used in classification (IBk) or number of features to consider (RandomForest); -L = LearningRate; -M = minNumObj (minimum number of instances per leaf); -M = Momentum; -N = TrainingTime; -R = CombinationRule; -S = Seed; -V = ValidationSetSize; -W = WindowSize.

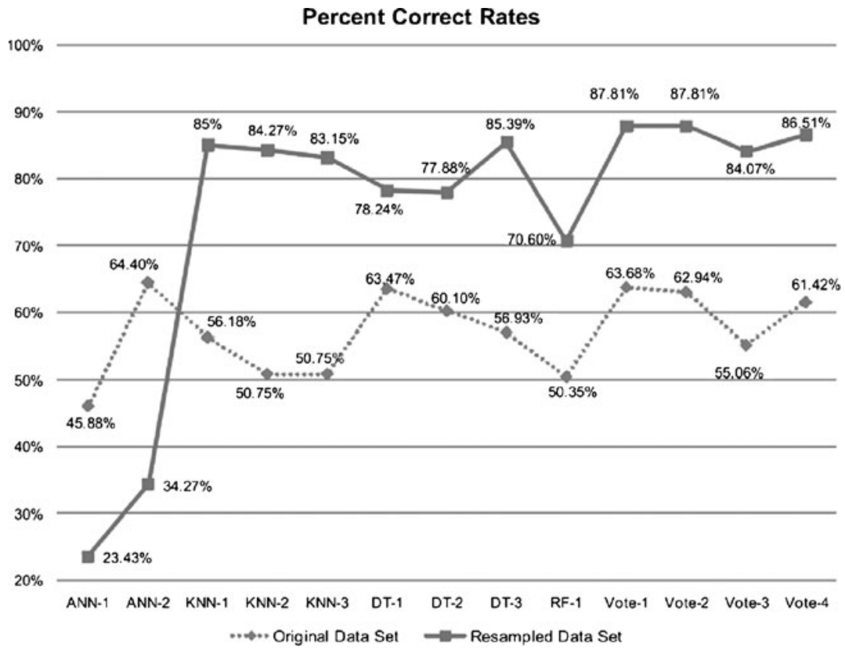


Fig. 7 Hit rates for all classifiers (original and resampled data sets)

Results

Percentage correct

Figure 7 shows the percentage of correct rates for all classifiers, using the original data set; Table 2 shows the results for the comparison of classifiers using the original data set. Tables 2, 3, 4 and 5 present uparrow and downarrow as cells. Statistically, column classifiers are better than row classifiers when the cells are filled with up arrow. On the other hand, column classifiers are worse than row classifiers when the cells are filled with down arrow. Vote-3 performed better than ANN-1. Vote-1 and Vote-2 performed better than ANN-1, all KNN, RF-1 and DT-3. Vote-4 performed better than ANN-1, KNN-1, KNN-2, RF-1and DT-3.

Figure 7 also presents percentages of correctly classified examples using the resampled data set. Table 3 shows the results for the statistical analysis comparing classifiers that used the resampled data set. The results show that Vote-1 and Vote-2 did better than ANN, DT-1, DT-2 and RF-1; and Vote-1 also performed better than KNN-2. Vote-4 performed better than all ANN, DT-1, DT-2, RF-1 and KNN.

Area under ROC (AUC)

Figure 8 shows area under ROC results using the original data set; Table 4 shows the statistical comparisons of classifiers using the original data set. Results show that Vote-1 performed better than all KNN, DT-1, RF-1and DT-3; Vote2 did better than KNN, DT-1 and RF-1; Vote-3 performed better than KNN-1 and RF-1; and Vote-4, better than KNN and RF-1.

Figure 8 shows the area under ROC results for classifiers using the resampled data set; Table 5 shows the statistical comparisons of classifiers using the resampled data set. The results show that Vote-1 and Vote-4 performed better than KNN-1, KNN-3, DT-1, DT-2 and RF-1; Vote-2 performed better than KNN-1, DT-1 and RF-1; and, lastly, Vote-3 performed better than KNN-1, KNN-3 and RF-1.

Discussion

The best results for each dataset, based on the percentage of correctly classified examples, were: (i) 64.40 % using ANN-2 and 63.68 % using Vote-1 for the original dataset; (ii)

Table 2 Percentage correct: original data set

Classifiers	ANN-1	ANN-2	KNN-1	KNN-2	KNN-3	DT-1	DT-2	DT-3	RF-1	(↑/↓)
Vote-1	↓		↓	↓	↓			↓	↓	(0/3/6)
Vote-2	↓		↓	↓	↓			↓	↓	(0/3/6)
Vote-3	↓	↑				↑				(2/6/1)
Vote-4	↓		↓	↓				↓	↓	(0/4/5)
(↑/↓)	(0/0/4)	(1/3/0)	(0/1/3)	(0/1/3)	(0/2/2)	(1/3/0)	(0/4/0)	(0/1/3)	(0/1/3)	

Table 3 Percentage correct: resampled data set

Classifiers	ANN-1	ANN-2	KNN-1	KNN-2	KNN-3	DT-1	DT-2	DT-3	RF-1	(↑/↓)
Vote-1	↓	↓		↓		↓	↓		↓	(0/3/6)
Vote-2	↓	↓				↓	↓		↓	(0/4/5)
Vote-3	↓								↓	(0/7/2)
Vote-4	↓	↓	↓	↓	↓	↓	↓		↓	(0/1/8)
(↑/↓)	(0/0/4)	(0/1/3)	(0/3/1)	(0/2/2)	(0/3/1)	(0/1/3)	(0/1/3)	(0/4/0)	(0/0/4)	

87.81 % using Vote-1 and Vote-2 for the resampled dataset. Considering the AUC values, the best results were: (i) 0.96 using ANN-2 and Vote-1 for the original dataset; (ii) 1 using KNN-3 and 0.99 using DT-3, Vote-1, Vote-2 and Vote-4 for the resampled dataset.

The best results for classifiers are obtained using resampled dataset because they allow classifiers to learn how to classify both classes with several examples and those with fewer examples [20]. Our results corroborate the literature. Figures 7 and 8 show that almost all (11 of 13) classifiers (with the exception of ANN-1 and ANN-2) provided better results using the resampled dataset.

We used a highly unbalanced data set. Consequently, some form of balancing the examples was required. Resampling the data set proved to be an advantage. It is important to optimize classification considering that errors in patient classification can present a problem for patients and healthcare facilities. For example, one possible complication would be to classify a routine patient as requiring urgent care; of course, an even worse situation would be to fail to identify SL-3 or Emergency level patients.

Related work

Recently, new advanced approaches have been developed to apply machine learning classifiers to healthcare. Only a few of these approaches compare to the work presented in this paper. One of the novelties of the present work is that it describes the use of machine learning classifiers to rank patients according to a healthcare measure. This procedure is similar to establishing SLs.

One of the studies similar to the work presented in this paper classified emergency patients into severity grades using data mining methods [21]. The records of 402 patients from an emergency department were classified by two expert physicians into five severity grades. Naïve Bayes and C4.5 were applied to generate classifiers based on patient data and determine severity grades [21]. Another related study described a non-supervised learning method based on cluster analysis and genetic algorithms. The goal was to classify patient risk at the moment of admission to a medical unit [22]. The proposal included a method for incorporation of information contained in the diagnostic hypotheses into the classification system [22].

But there are several studies that present metrics for patient ranking; these studies, however, neither present an automatic system for classification of patients according to these metrics; nor do they apply machine learning methods to assist the ranking of patients. For instance, [23] presents measuring tools that classify patients at Intensive Care Units (ICUs) considering the severity of illnesses and Nursing staff workload. The metrics include: Acute Physiology and Chronic Health Evaluation (APACHE), Simplified Acute Physiological Score (SAPS), Mortality Prediction Model (MPM), Sepsis Related Organ Failure Assessment (SOFA), Logistic Organ Dysfunction System (LODS), Sepse Score, OMEGA Score System, Time Oriented Score System (TOSS), Project of Research of Nursing (PRN) and Therapeutic Intervention Scoring System (TISS). The authors in [24] re-defined a system of classification of surgical complications to increase its accuracy and acceptability in the surgical community. The proposed grading system draws on the therapy used to treat

Table 4 Area under ROC (AUC): original data set

Classifiers	ANN-1	ANN-2	KNN-1	KNN-2	KNN-3	DT-1	DT-2	DT-3	RF-1	(↑/↓)
Vote-1			↓	↓	↓	↓		↓	↓	(0/3/6)
Vote-2			↓	↓	↓	↓			↓	(0/4/5)
Vote-3			↓						↓	(0/7/2)
Vote-4			↓	↓	↓				↓	(0/5/4)
(↑/↓)	(0/4/0)	(0/4/0)	(0/0/4)	(0/1/3)	(0/1/3)	(0/2/2)	(0/4/0)	(0/3/1)	(0/0/4)	

Table 5 Area under ROC (AUC): resampled data set

Classifiers	ANN-1	ANN-2	KNN-1	KNN-2	KNN-3	DT-1	DT-2	DT-3	RF-1	(↑/↓)
Vote-1			↓		↓	↓	↓		↓	(0/4/5)
Vote-2			↓			↓			↓	(0/6/3)
Vote-3			↓		↓				↓	(0/6/3)
Vote-4			↓		↓	↓	↓		↓	(0/4/5)
(↑/↓)	(0/4/0)	(0/4/0)	(0/0/4)	(0/4/0)	(0/1/3)	(0/1/3)	(0/2/2)	(0/4/0)	(0/0/4)	

the complication and it classifies patients into five grades. The classification was experimented in a cohort of patients. Its reproducibility and its personal judgment were evaluated through an international survey with questionnaires sent to surgical centers worldwide.

To our knowledge, there are no studies that combine the technology and methods presented in this paper: (1) machine learning algorithms that classify patients to identify healthcare needs in a primary healthcare unit, and (2) a surveillance system to support surveillance of the classification.

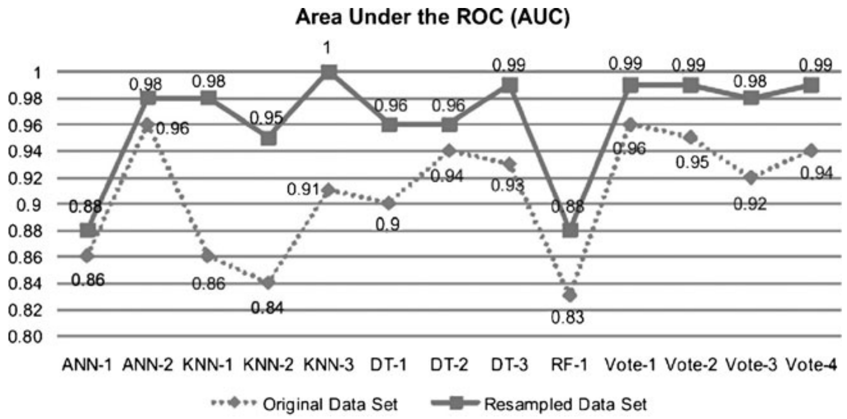
Conclusion

The present study investigated automatic recommendation of SL scores using pattern-based classification of electronic patient records. We also explained how machine learning classifiers are able to achieve SL recommendation. The paper described and evaluated a software architecture-based framework supporting the automatic classification of SLs using k-Nearest Neighbor (KNN), Artificial Neural Network (ANN), Relevance Feedback (RF), Decision Tree (DT), Ensemble of classifiers (Vote) and a thesaurus. Results indicate that the best classifier was the Vote-SL because Vote-1 and Vote-2 were able to correctly classify 87.81 percent of patients, recommending routine, educational, therapeutic or special medical procedures.

Our study suggests classifiers can be used to recommend SL based on patient records and alert pediatricians about a demand for procedures in primary healthcare. The results also indicate that selection of attributes can have a great effect on the performance of the system. Therefore we believe that our automatic recommendation of surveillance level can still benefit from improvements in processing procedures.

It is our goal to further our investigation using larger training sets and automatic normalization of attributes, for example, using semantic networks. We realized that healthcare workers apply a varied specialized vocabulary when they fill out exam sheets, provide diagnoses and treatment information. As a result, we argue that healthcare-related classification systems should be augmented with concepts from different medical sources. Consequently, computational linguistics can be a useful method to treat data provided in language format and to optimize computational processes such as pattern-based classification. Results from our scientific efforts can be applied to different types of medical systems. The results of systems supported by the framework presented in this paper may be used by healthcare and governmental institutions to improve healthcare services and even alert authorities about the possibility of an epidemic.

Fig. 8 Hit rates for all classifiers (original and resampled data sets)



Acknowledgments The authors would like to thank FAPESP and RUSP for the financial support. The authors would also like to acknowledge Professor Evandro Ruiz, Gisele Miranda, Professor Joaquim Felipe, Renato Pedigoni, Professor Rudinei Goularte, Thiago da Costa, and everyone at Vila Lobato.

Conflict of interest statement All authors declare that they have no conflicts of interest.

References

1. Salles, R. F., *Análise de um programa de intervenção com bebês e famílias atendidas em unidades básicas de saúde—SUS [dissertation]*. São Carlos (SP): Universidade Federal de São Carlos, 2001.
2. Panico, S. R. G., Canziani, M. L., and Guerchon, N., Políticas Públicas Municipais. In: Panico, S. R. G., (Ed.), *Indicadores Nipe: Subsídios para Políticas Municipais de Saúde*. São Carlos: NIPE; 1997.
3. Pollettini, J. T., Miranda, G. H. B., Goularte, R., Panico, S. R. G., Daneluzzi, J. C., and Macedo, A. A., Sistema de Informação Geográfica: uma Abordagem Integrada a Sistemas de Informação em Saúde. In: XII Congresso Brasileiro de Informática em Saúde (CBIS), 2010. Porto de Galinhas, PE. Anais do CBIS 2010,18–22 outubro de 2010.
4. Dietterich, T. G., Limitations on inductive learning (extended abstract), 1997. web.engr.oregonstate.edu/~tgd/publications/ml89-limits.ps.gz, Visited Apr. 2012.
5. Kohavi, R. A., Study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Int J Artif Intell Tool*. 6 (4):537–566, 1997.
6. Opitz, D., and Maclin, R., Popular ensemble methods: An empirical study. 11:169–198, 1999.
7. Schaffer, C., A conservation law for generalization performance. In: Cohen, W. W., and Hirsh, H., (Eds.), *Proceedings of the Eleventh International Conference on Machine Learning*. New Brunswick, New Jersey, 1994, p 259–265.
8. Witten, I. H., and Frank, E., *Data mining: practical machine learning tools and techniques*. 2nd ed. San Francisco: Morgan Kaufmann, 2005.
9. A.R. F. F. (ARFF). Wiki article ARFF. http://www.cs.waikato.ac.nz/ml/weka/arff.html, Visited Feb. 2010.
10. Porter, M., An algorithm for suffix stripping. *Program*. 14(3):130–7, 1980.
11. Rocchio, J. J., Relevance feedback in information retrieval. In: Salton, G., (Ed.), *The Smart Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River: Prentice-Hall, Inc., 1971.
12. Pollettini, J. T., Nicolas, F. P., Panico, S. R. G., Daneluzzi, J. C., Tinós, R., Baranauskas, J. A., and Macedo, A. A., A software architecture-based framework supporting suggestion of medical surveillance level from classification of electronic patient records. In: *Proc of the 12th IEEE International Conference on Computational Science and Engineering*. Vancouver, Canada: IEEE Computer Society, 2009, p 166–173. doi:10.1109/CSE.2009.231
13. Costa, T. M., Daneluzzi, J. C., Panico, S. R. G., Felipe, J. C., Projeto de Dados de Sistema para Integração Longitudinal de Informações e Procedimentos em Centros Médicos. In: Proceedings of XI Congresso Brasileiro de Informática em Saúde, 2008. Campos do Jordão (SP): CBIS, 2008. 6p.
14. Costa, T. M., Ruiz, E. E. S., and Panico, S. R. G., Projeto interdisciplinar para a criação de uma ferramenta computacional que facilite a pesquisa acadêmica e o acompanhamento da saúde e do desenvolvimento de adolescentes em atenção básica à saúde. Relatório de Iniciação Científica apresentado à Fundação de Amparo ao Ensino e Pesquisa Aplicada do HCFMRP. Ribeirão Preto (SP), 2006.
15. de Paula, D. S., Panico, S. R. G., Daneluzzi, J. C., Ruiz, E. E. S., Felipe, J. C., and Macedo, A. A., Sistema de Informação de Apoio ao Programa de Educação para Pais e Famílias. In: *Proceedings of XI Congresso Brasileiro de Informática em Saúde, 2008*. Campos do Jordão (SP): CBIS, 2008. 6p.
16. Pollettini, J. T., Tinós, R., Panico, S. R. G., Daneluzzi, J. C., and Macedo, A. A., Classificação automática de pacientes para atendimento médico pediátrico multidisciplinar a partir do seu Grau de Vigilância. In: VIII Workshop de Informática Médica (evento paralelo ao Congresso da Sociedade Brasileira de Computação), 2008, Belém-Pará-Brazil. Anais do VIII Workshop de Informática Médica, 2008. p. 61–70.
17. Pollettini, J. T., Tinós, R., Panico, S. R. G., Daneluzzi, J. C., and Macedo, A. A., Vigilância em atenção básica à saúde a partir do uso de relevance feedback para classificação de pacientes em diferentes níveis de cuidado em saúde. In: Workshop de Informática Médica (evento paralelo ao XXIX Congresso da Sociedade Brasileira de Computação), 2009, Bento Gonçalves. IX Workshop de Informática Médica (WIM 2009), 2009. p. 1945–1954.
18. Demsar, J., Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research (JMR)*. 7:1–30, 2006.
19. Provost, F., and Domingos, P., Tree induction for probability-based ranking. *Mach Learn*. 52(3):199–215, 2003.
20. Batista, G., Prati, R., and Monard, M., A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*. 6:20–9, 2004.
21. Zmiri, D., Shahar, Y., and Taieb-Maimon, M., Classification of patients by severity grades during triage in the emergency department using data mining methods. *J Eval Clin Pract*. 18(2):378–388, 2012. doi:10.1111/j.1365-2753.2010.01592.x. Available at: http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2753.2010.01592.x/abstract, Visited Apr. 2012.
22. Chacón, M., and Luci, O., Patients classification by risk using cluster analysis and genetic algorithms. In: *Progress in Pattern Recognition, Speech and Image Analysis*. Lecture Notes in Computer Science. 2905/2003:350–358, 2003. doi:10.1007/978-3-540-24586-5_43. Available at: http://www.springerlink.com/content/02m4u30ktkfrfvlm/, Visited Apr. 2012.
23. Tranquitelli, A. M., and Padilha, K. G., Sistemas de classificação de pacientes como instrumentos de gestão em Unidades de Terapia Intensiva. *Rev. Esc. Enferm. USP [online]*. 41(1):141–146, 2007. ISSN 0080–6234. doi: http://dx.doi.org/10.1590/S0080-62342007000100019. Visited Apr. 2012.
24. Dindo, D., Demartines, N., and Clavien, P. A., Classification of surgical complications: A new proposal with evaluation in a cohort of 6336 patients and results of a survey. *Ann Surg*. 240(2):205–213, 2004. doi:10.1097/01.sla.0000133083.54934.ae. Visited Apr. 2012.