A Framework for CBR Development and Experimentation with Application to Medical Diagnosis

Beatriz López, Pablo Gay, Albert Pla, and Carles Pous

Universitat de Girona, Campus Montilivi, edifice P4, 17071 Girona
beatriz.lopez@udg.edu, {pgay,apla}@eia.udg.edu, carles.pous@udg.edu

Summary. Most of the data mining techniques focus on classification tasks and so several tools have been designed to help engineers and physician in building new classification systems. However, most of the tools does not take into account the application domain. In this paper we present a case-based reasoning (CBR) tool with the aim to support the development and experimentation of a particular medical classification task: diagnosis. Our tool, eXiT*CBR provides navigation facilities to the engineers and physician across several experimentation results, assures experiment reproducibility and incorporates functionalities for independizing data, if required. The use of eXiT*CBR is illustrated with a Breast Cancer diagnosis system we are currently developing.

1 Introduction

There is an increasing interest on the use of data mining techniques in the Life Sciences [4, 17, 11, 12]. Particularly, learning techniques for classification tasks are widely present in several fields. As an example, the gene expression settings can be cited, when we might want to predict the unknown functions of certain genes, given a group of genes whose functional classes are already known [16]. We have another example in medical diagnosis, where there may exists many cases that correspond to several diseases, together with their associated symptoms. Classification tools can offer a second opinion to the physicians with a potentially low cost, noninvasive solution to improving the diagnosis [6]. Thus, given a set of sample data, the objective of learning classification techniques is to build such a method that enables the classification of new data successfully, with the corresponding error rates as low as possible.

Our research concerns Case-Based Reasoning (CBR). Case-based reasoning is an approach to problem solving and learning based on examples (past experiences). It consist of four stages, that are repeated for each new situation: Retrieval, Reuse, Revise, and Retain. *Retrieval* consists on seeking for past situation or situations similar to the new one . *Reuse* is the second step and it uses the extracted cases to propose a possible solution. *Revise* the solution proposed in the reuse phase is often a human expert. Finally, in the *Retain* it has to be

J.M. Corchado et al. (Eds.): IWPACBB 2008, ASC 49, pp. 45–54, 2009.
springerlink.com

decided if it is useful to keep the new situation in the case base in order to help on the future diagnosis of new situation.

Each particular domain requires the selection of the appropriate techniques for each phase and the appropriate tuning of the different parameters. Several tools have been designed in order to guide the development as for example cf [3], CBR* [10] and CBR Shell [2]. They are general purpose tools regarding CBR, so they can be used for developing a cancer diagnosis system as well as an electricity fault diagnosis system. When designing a new tool, however, it is important to identify the target of the user, since each discipline has its own particularities that establish some requirements on the tool. Particularly, the field of Medical diagnoses requires to provide a user-friendly interface, result visualization based on ROC curves, and experiment reproducibility. First, a user-friendly interface allows physicians and engineers to work side by side when defining a new system. This requirement is common to most of the previous techniques. Second, ROC curves are the kind of visualization tools with which physicians usually work and helps them in interpreting the obtained results. ROC (Receiver Operator Characteristics) curves depict the tradeoff between hit rate and false alarm rate [8]. Finally, experiment repetition is often a critical but necessary process to assure the definite parameters that provide the best results. And what is more important, assures reproducibility, so that the results obtained by another team working independently agree given the same method and identical test material. From our knowledge, nowadays there is no tool supporting these two features. This two features provides, in addition to a framework for developing new systems, a platform that helps in the CBR experimentation.

In this paper we present a new tool, eXiT*CBR, with the aim of providing support to the development and experimentation of case-based classification in medicine, that satisfy the above requirements. This paper is an extension of the work initiated in [19]. In this paper, we provide and explain the different modules of the architecture of our tool, as well as the common data representation required for any application. We have also extended the system with additional functionalities regarding case independency. Case independency is assumed by most of the data mining methods, but should be specifically tested when dealing with medical applications, since most of the times this assumption does not hold.

This paper is organized as follows. First, we introduce the architecture of our system. Next, we briefly describe the functionalities of eXiT*CBR. We continue by giving some details of the breast cancer diagnostic application being developed with the tool. Afterwards, we relate our work with previous ones of the state of the art. We end the paper with some conclusions.

2 eXiT*CBR Framework

The aim of our tool is to help engineers and physicians to achieve the appropriate CBR system for medical diagnosis, given the data they have and based on an empirical approach. For this purpose, our architecture follows a modular approach based on the different phases required by any CBR system (see Figure 1).

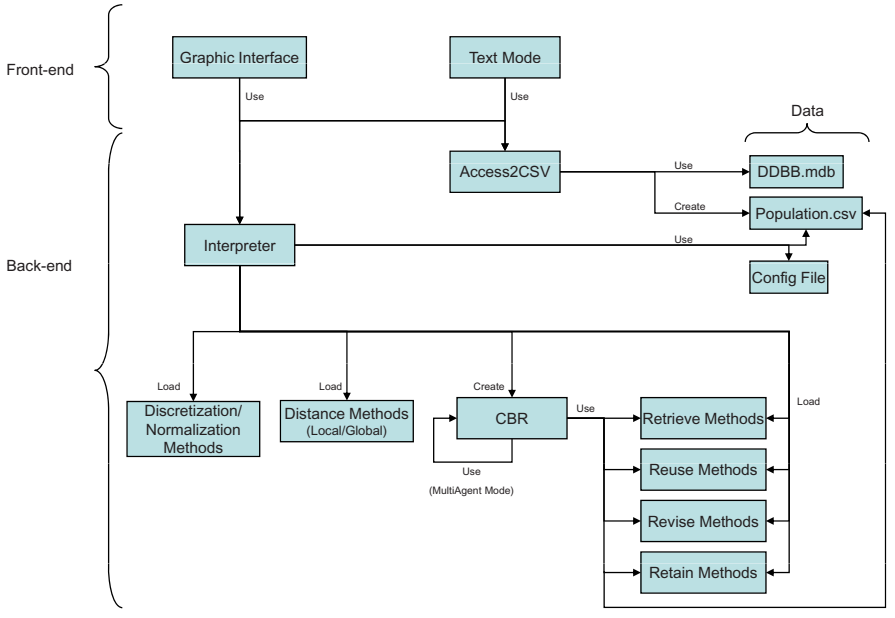

Fig. 1. Tool architecture

Particularly, we distinguish the following modules: Pre-process, Retrieve, Reuse, Revise, Retain, Experimentation, Post-process. Thanks to the object oriented implementation of the architecture, that provides modularity, reusability and extensibility properties, any module is implemented as a generic class. So when a new method is required and not provided in the system, it can be added as a particular instance of the generic class.

In order to handle the data by any method, a common representation is required. In eXiT*CBR we have chosen a plain csv file. The structure of the file is the following:

- First row corresponds to the attribute descriptions (for example, "Age when the first kid was born")
- Second row corresponds to the attribute name (usually in a cryptic form, as for example, "age1stborn").
- Third row corresponds to the attribute type (-1 ignore, 0 discrete, 1 numerical, 2 textual, 3 data)
- Fourth row corresponds to the attribute weight.

We believe that this simply plain representation covers most of the data used in Medical applications[1]. The advantage is that this representation is easy to

[1] In fact, physicians are usually gathering data in Microsoft excel or Spss files, with a similar format.

manage and general enough to be used by any of the current techniques of CBR (mainly, distance functions).

2.1 Pre-processing Module

As preprocessing steps we can have mainly three kind of process: discretization, normalization and feature selection. Discretization methods deal with numerical to categorical data conversion [15]. Normalization methods assures that numerical values are all comparable in the [0,1] interval. Finally, feature selection methods determine which of the available features are relevant for the application. Usually relevant and irrelevant features can be labeled with weights in a weighted learning process, so that relevant features have high weights, while irrelevant features have low weights assigned [13, 15].

Note, however, that in order to learn which features are relevant, other Data Mining or Computing techniques should be considered as a previous step of the CBR, not considered as pre-processing. For this purpose, eXiT*CBR admits plug in of any Data Mining technique in the same interface.

2.2 Retrieve Module

The key issue in the retrieve phase is the definition of the similarity measure. There are local and global similarity measures. On one hand, local similarity measure concern the comparison of two data. There can be as many kind as local similarity measures as type of operands available. For example, the Euclidean distance is the most often proposed measure to handle numeric data, while the Hamming distance is set for categorical (discrete) data [22]. Other local similarity measures regarding data trees (to handle, for example, inheritance information), series, and dates can also be used. On the other hand, global similarity measures is related to the combination of the different local similarity measures. An example of them is the weighted average, but much other can be considered [20].

Other important methods in this phase are the ones related to unknown values: missing completely at random (MCAR), missing at random (MCR), and not missing at random (NMAR) [23]. MCAR is when the probability of missing a value is the same for all attributes, MCR is when the probability of missing a value is only dependent on other attribute, and NMAR is when the probability of missing a value is also dependent on the value of the missing attribute. Missing values are cost sensitive, as explained in [23], and widely present in Medical applications.

Finally, the selection method employed should also be determined: (e.g. k-nearest neighbor).

2.3 Reuse Module

One of the principal limitation of CBR in medical diagnosis is the reuse phase [14]. The majority of medical-CBR rely on suggesting past solutions without further adaptation process. Recent works as [5] propose a probabilistic approach;

however, the Bilska-Wolak method has been applied with a few number of features, and much more research effort should be done in order to deploy it in real environments.

2.4 Revise Module

As a revise phase, most of the current medical-CBR relies on human feedback. Up to now, no further alternatives are open. However, in other environments, simulators are also possible [9].

2.5 Retain Module

Once the solution proposed for the new presented case is revised, a decision has to be made concerning to the necessity of retaining the new case. This decision will depend on whether the cases that we already have, produce a correct solution or not. Since CBR system's core is a base of cases that can be very large, it has a lot in common with the data mining methodologies for data processing. When the case base grows, it is necessary to have a good maintenance policy. This means that we have to delete, add or modify cases in order to keep the system performing well. In this stage Instance based Learning algorithms such as IB3 ([1]) or DROP4 ([21]) can be applied to help on the decision of just keeping the new case, forgetting it, or keeping it in expenses of another cases deletion.

2.6 Experimentation Module

There are three main experimentation parameters: experimentation measures, experimentation methodology, and visualization result method. First, experimentation measures concerns the kind of measure that physicians are interested on. Others as recall, precision, success, failure and much more can be considered, although specificity and sensibility are most commonly used in the medical domain [8]. Second, the experimentation method can be any statistically supported methodology, being the stratified cross-validation technique the one that has proved to be the most adequate when few cases are available, as in medical applications happens [7]. Finally, the usual way of visualizing results in medical applications is by means of ROC curves [5]; even so, other alternatives such as cost plots, and others can also be used.

2.7 Post-processing Module

Current trends on data mining systems should consider different use of the models learned, and consistently, different kinds of post-processing steps should be taken into account. Mainly, we distinguish between off-line and on-line modes. In an off-line mode, the system is being developed, and thus several batch processes and validation procedures should be applied according to the experimentation parameters. The answer of the system in this context are the visualization plots. On the other hand, in the on-line mode the system cooperates with other systems in the solution of a given problem. There could be several approaches based on ensemble learning techniques, including distributed and agent-based approaches [18].

3 eXiT*CBR Functionalities

The aim of our tool is to help engineers and physicians to achieve the appropriate CBR system for Medical diagnosis, given the data they have, based on an empirical approach. For this purpose, our architecture offers the following set of functionalities regarding the user interaction:

1. **Edit configuration:** the user can define a configuration file in a friendly user manner, in which all the available techniques are prompted out in pop-up menus.
2. **Data independization:** Machine learning methods in general, and CBR in particular, assume that examples are independent. However, medical data can not always satisfy this constraint. For example, in a breast cancer data, two members of the same family can be also in the same dataset. This clearly dependency situation can be removed, if family relationship are identifiable in the dataset. Of course, other kind of hidden dependencies are more difficult to handle.
3. **Data conversion:** transforms any original data set in a csv file. For example, we have converted a breast cancer relational data base in a csv file.
4. **Data set generation:** This functionality allows the definition of a data set for experimentation. The input is the original data base where the medical information is contained (cases). The output, a set of different data sets that allow training and testing the system according to the experimentation procedure (for example, cross-validation).
5. **CBR application:** runs the CBR experiment according to a given configuration file. As a results, several internal files are generated that contain the outputs of the application and other result-visualization supporting information. All of the files involved in a run are kept internally by the system (with an internal code): configuration file, input file (data sets), output files, and others. So in each moment, the experiment can be replicated, even if

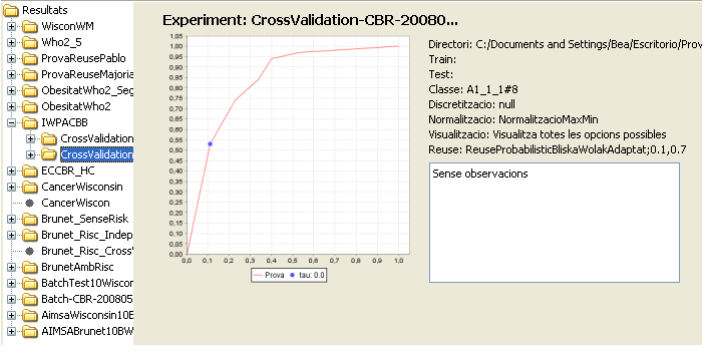

Fig. 2. Experimenter navigator window

the user has removed the files from the directory originally indicated in the configuration file.

6. **Experimenter navigator:** This facility allows the users to navigate in a friendly manner across the different experiments run so far. Figure 2 shows a snapshot of the experimenter navigator. On the left panel, the different experiments executed are listed. When the user moves the cursor across the different experiments, the corresponding results are shown on the top right panel (e.g. ROC curve in Figure 2).

The experimenter navigator is perhaps the most innovative issue in our framework. The explicitly consideration of an independent data functionality is also new.

4 Application to Breast Cancer Diagnosis

Our framework has been developed using the Java language and the jfreechart library [2]. It is compatible with Linux and Windows operating system platforms. The first application we are trying to complete with eXiT*CBR is a breast cancer case-based system.

First of all, we have converted our original access file into a csv file thanks to the data conversion procedure. We have continued by generating different datasets in order to perform a cross validation experimentation. Then, we have set up the CBR system thanks to the edit configuration functionality. The results finally obtained are the ones shown in the navigation window of Figure 2. Figure 3 illustrates the process followed. Unfilled boxes show optional steps, in case we wish to change the destination directory of the results, and the use of another data mining method (plug in).

Next, we realize that the original dataset have dependent data. Therefore, we have run the data independization functionality in order to remove dependent cases and obtaining a new dataset. We run again the experiment with the new data, obtaining new results. Both experiments, have been enlarged by using the corresponding button of the experimenter navigator, and their comparison can be seen in Figure 4. Thus, the tool has facilitated the CBR development and experimentation.

5 Related Work

There are several previous works in which general CBR tools have been developed. For example, cf [3] is a lisp environment designed to be used for rapid prototyping. CBR* [10] and CBR Shell [2] both have been designed according to an object oriented methodology and implemented in java, as our system. Thus, the modularity, reusability and extensionality properties of the object oriented paradigm has been inherited in the CBR frameworks. The main difference of

[2] http://www.jfree.org/jfreechart/

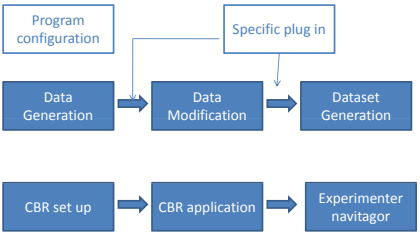

Fig. 3. Main steps followed to develop the breast cancer application

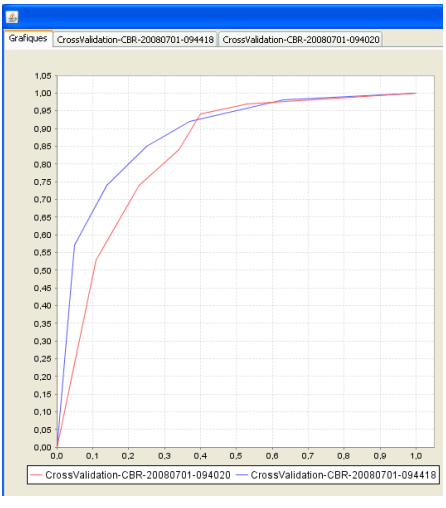

Fig. 4. Comparison of two experiments through the experimenter navigator enlargement facility

the previous approaches is their goal. Our framework is designed to provide to engineers and physicians navigation functionalities across different experiments, helping them to choose the appropriate CBR methods and parameters, and assuring experiment reproducibility. Additionally, we are focussing on a particular case of CBR: medical diagnosis. So we are considering CBR for classification and the visualization techniques are mainly based on ROC graphs, the most commonly used technique in this domain.

6 Conclusions

The development of a CBR system for medical diagnosis purpose is always a time consuming activity. Even that several tools have been developed to facilitate the rapid construction of prototypes, none of them has contemplate the navigation through the experiments of different CBR configurations. The experiment is a

cornerstone in the empirical approach of engineers and physicians to acquiring deeper knowledge about the data they have.

The tool we have developed, eXiT*CBR, tries to fill this gap. We believe that our development and experimental framework would facilitate the interactions with the physicians, without getting lost in the different experiments generated. In addition, our tool allows reproducibility, since we are able to replicate the experiments as many times as required. We think that this is an important issue when dealing with research results concerning medical applications, and we encourage to the medical community to use this kind of tools. We are currently applying eXiT*CBR for developing a breast cancer CBR diagnostic system, as illustrated along the paper.

Acknowledgments

This research project has been partially funded by the Spanish MEC project DPI-2005-08922-CO2-02, Girona Biomedical Research Institute (IdiBGi) project GRCT41 and DURSI AGAUR SGR 00296 (AEDS).

References

1. Aha, D.W., Kibler, D., Albert, M.: Instance based learning algorithms. Machine Learning 6, 37–66 (1991)
2. Aitken, S.: Cbr shell java-v1.0, `http://www.aiai.ed.ac.uk/project/cbr/cbrtools.html`
3. Arcos, J.L.: cf development framework, `http://www.iiia.csic.es/~arcos/`
4. Bichindaritz, I., Montinali, S., Portinali, L.: Special issue on case-based reasoning in the health sciences. Applied Intelligence (28), 207–209 (2008)
5. Bilska-Wolak, A.O., Floyd Jr., C.E.: Development and evaluation of a case-based reasoning classifier for prediction of breast biopsy outcome with bi-radstm lexicon. Medical Physics 29(9), 2090–2100 (2002)
6. Bilska-Wolak Jr., A.O., Floyd, C.E., Nolte, L.W., Lo, J.Y.: Application of likelihood ratio to classification of mammographic masses; performance comparison to case-based reasoning. Medical Physics 30(5), 949–958 (2003)
7. Demar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research (7), 1–30 (2006)
8. Fawcett, T.: Roc graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Labs (2003)
9. Hammond, K.J.: Explaining and repairing plans that fail. Artif. Intell. 45(1-2), 173–228 (1990)
10. Jaczynski, M.: A framework for the management of past experiences with time-extended situations. In: CIKM, pp. 32–39 (1997)
11. Jurisica, I., Glasgow, J.: Applications of case-based reasoning in molecular biology. AI Magazine 25(1), 85–95 (2004)
12. Lavrac, N., Keravnou, E., Zupan, B.: Intelligent Data Analysis in Medicine and Pharmacology. Kluwer Academic Publishers, Dordrecht (1997)
13. Martinez, T.: Selecció d'atributs i manteniment de la les base de casos per a la diagnosi de falles. Master's thesis, Universitat de Girona (2007)

14. Montani, S.: Exploring new roles for case-based reasoning in heterogeneous ai systems for medical decision support. Appl. Intelligence (28), 275–285 (2008)
15. Francisco Nez, H.F.: Feature Weighting in Plain Case-Based Reasoning. PhD thesis, Technical University of Catalonia (2004)
16. Orengo, C.A., Jones, D.T., Thornton, J.M.: Bioinformatics. Genes, Proteins & Computers. BIOS Sientific Publishers (2004)
17. Perner, P.: Intelligent data analysis in medicine - recent advances. Artificial Intelligence in Medicine (37), 1–5 (2006)
18. Plaza, E., McGinty, L.: Distributed case-based reasoning. The Knowledge Engineering Review 20(3), 261–265 (2005)
19. Pous, C., Pla, A., Gay, P., López, B.: exit*cbr: A framework for case-based medical diagnosis development and experimentation. In: ICDM Workshop on Data Mining in Life Sciences (2008) (accepted, in press)
20. Torra, V., Narukawa, Y.: Modeling Decisions: Information Fusion and Aggregation Operators. Springer, Heidelberg (2007)
21. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)
22. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
23. Zhang, S., Qin, Z., Ling, C.X., Sheng, S.: Missing is useful: Missing values in cost-sensitive decision trees. IEEE Transactions on Knowledge and Data Engineering 17(12), 1689–1693 (2005)