EXPERIMENTAL COMPARISON OF MACHINE LEARNING APPROACHES TO MEDICAL DOMAINS: A CASE STUDY OF GENOTYPE INFLUENCE ON ORAL CANCER DEVELOPMENT

Flavio Baronti Federico Colla Valentina Maggini Alessio Micheli
Alessandro Passaro Anna Maria Rossi Antonina Starita
Università di Pisa, Italy

ABSTRACT

Research in medical domains is facing new challenges as the available information increases in quantity and quality. In this context, Machine Learning methodologies can provide the right tools for data analysis, which can cope with recurring problems in medical research, such as the integration of clinical and genetic data. In this study we provide an experimental comparison of an heterogeneous subset of Machine Learning methods. For such a purpose, a representative dataset for medical analysis was chosen which regards Head and Neck Squamous Cell Carcinoma (HNSCC). HNSCC is a kind of oral cancer associated with smoking and alcohol drinking habits; however the individual risk could be modified by genetic polymorphisms of enzymes involved in the metabolism of tobacco carcinogens and in the DNA repair mechanisms. To study this relationship, the data set comprised demographic and life-style (age, gender, smoke and alcohol), and genetic data (the individual genotype of 11 polymorphic genes), with the information on 124 HNSCC patients and 231 healthy controls. Strengths and weaknesses of the different algorithms when applied to medical datasets, such as the one considered, will be analyzed, with particular attention to the issue of missing values.

Keywords: machine learning, oral cancer, XCS, learning, classifier systems, decision trees, neural networks, support vector machines, missing values.

1. INTRODUCTION

Medical research is undergoing major changes as it gets access to growing amounts of data. In particular, in the last years, the increasing availability of genetic information has urged the need of more powerful methods for data analysis. New techniques such as DNA sequencing and microarrays allow the acquisition of vast databases, in which genetic information is mixed with clinical data collected in research and health-care centers.

The discovery and the study of genetic interactions is central to the understanding of molecular structure and function, cellular metabolism, development of cells and tissues, and response of organisms to their environments. If such interaction patterns could be measured for various kinds of tissues and the corresponding data could be interpreted, there would be obvious clinical benefits and novel tools for diagnostics, identification of candidate drug targets, and predictions of drug effectiveness for many diseases would emerge.

Tools for analyzing medical data generally share a set of characteristics. They have to be tolerant to noise and uncertainty: medical data is seldom accurate, and the diagnosis can be subject to errors. They should be able to deal with missing values: data collection is generally a by-product of medical treatment — and even when it is the main aim of a study, completeness cannot be enforced. Finally, the result of the analysis should be easily convertible in human-readable form, in order to allow the researcher to understand the knowledge the tool extracted from the data.

Machine Learning approaches to medical domains are very promising, although their potentialities are not yet fully exploited. For this reason, an extended analysis of the strengths and weaknesses of the different Machine Learning algorithms, when applied to medical datasets, will be of the greatest interest. This work is an experimental step in this direction.

In order to provide a benchmark application for our study, we considered a dataset on the development of head and neck squamous cell carcinoma (HNSCC), whose characteristics are representative of the typical medical data analysis problem. Moreover, previous studies regarding the application of a Machine Learning technique (XCS) to same dataset were available — see Maggini et al (1) and Passaro et al (2).

HNSCC is mainly associated with smoking and alcohol drinking, but genetic polymorphism of enzymes involved in the metabolism of tobacco carcinogens and in the DNA repair mechanisms can influence the risk factor. The subjects were thus described with a combination of individual demographic and lifestyle data (gender, age, smoking and drinking habits) and genetic data (the individual genotype at 11 polymorphic genes potentially relevant to this disease) — along with a single value, which stated if they had cancer or not when the database was compiled.

The dataset was analyzed through four different machine learning approaches. The first two, namely XCS and C4.5, were chosen since they produce easily interpretable results, and can seamlessly work with missing data and different data types. The last two, neural networks and support vector machines, were employed for their high discriminative power — although they do not accept missing data, and do not produce readily interpretable results.

The rest of this paper is structured as follows. Section 2 describes the analysis aim, and summarizes the structure of the data set. Section 3 gives a brief overview of the Machine Learning algorithms we used

to perform the analysis. Results of our tests are reported in Section 4. Section 5 draws conclusions and sketches future research directions.

2. PROBLEM

The data set we analyzed was designed to explore the influence of genotype on the chance to develop HNSCC. It is already well-known that this kind of cancer is associated with smoking and alcohol-drinking habits, it is more common among males and its incidence increases with age. The individual risk however could be modified by genetic factors, such as polymorphisms of enzymes involved in the metabolism of tobacco carcinogens and in the DNA repair mechanisms. The patients were thus described with a combination of demographic and lifestyle data (sex, age, smoking and drinking habits) and genetic data (the polymorphisms of eleven genes believed to be relevant to this disease) — along with a clinical value which stated if they had cancer or not when the database was compiled.
The genotype information provided by molecular testing regarded eleven genes involved with carcinogen-metabolizing (CCND1, NQO1, EPHX1, CYP2A6, CYP2D6, CYP2E1, NAT1, NAT2, GSTP1) and DNA repair systems (OGG1, XPD). Nine of these genes have two allelic variants; let's call them a_1 and a_2. Since the DNA contains two copies of each gene, there exist three possible combinations: a_1a_1, a_2a_2 (the homozygotes) and a_1a_2 (the heterozygote — order does not matter). The homozygotes where represented with values 0 and 2, while the heterozygote with 1. Due to dominance, the heterozygote is possibly equivalent to one of the homozygotes; however, for many of the considered genes this dominant effect is not known. So class 1 can be either equivalent to class 0, or to class 2. The remaining two genes have 4 allelic variants, which result in 9 combinations; they were sorted by their activity level, and put on an integer scale from 0 to 8.
The full data consists of 355 records, with 124 positive elements (HNSCC patients) and 231 negative (controls). They were collected in different periods between 1997 and 2003; this has led to many missing data among the genotypic information of patients. Actually only 122 elements have complete genotypic description; the remaining 233 have missing values ranging from 1 to 9, with the average being 3.58. As an overall figure, of the $11\times355=3905$ genotype values, just 3070 are present: 21% of the genotype information is missing.

3. METHODS

XCS

XCS was first introduced by Wilson (3) and (4), as an evolution of the Learning Classifier Systems (LCS) by Holland (5), a machine learning technique which combines reinforcement learning, evolutionary computing and other heuristics to produce adaptive systems. Similarly to its ancestors, an XCS maintains and evolves a population of classifiers (rules) through a genetic algorithm. These rules are used to match environmental inputs and choose subsequent actions. Environment's reward to the actions is then used to modify the classifiers in a reinforcement learning process.
XCS introduces a measure of classifiers' fitness based on their accuracy, i.e. the reliability of their prediction of the expected payoff, and applies the GA only on the action set, the subset of matching classifiers which suggest the chosen action. This gives the system a strong tendency to develop accurate and general rules to cover problem space and allow the system's "knowledge" to be clearly seen. A thorough description of XCS can be found in Butz (6).
During learning XCS tends to evolve an accurate and complete mapping of condition-action-prediction rules matching the data, so that the final number of rules is often quite high, as reported in Wilson (7) and (8). Thus at the end of a run a ruleset reduction algorithm must be applied to extract a small subset of rules which attain the same performance level (7).

Decision Trees

Decision trees are a classical machine learning tool for classification and prediction – see Quinlan (9). Moreover they provide some of the most valuable features required for our kind of application, such as results' interpretability, treatment of different data types, and robustness to missing data.
A decision tree is a classifier in the form of a tree structure, where each leaf node indicates the value of a target class and each internal node specifies a test to be carried out on a single attribute, with one branch and sub-tree for each possible outcome of the test. The classification of an instance is performed by starting at the root of the tree and moving through it until a leaf node is reached, which provides the classification of the instance.
Among the variety of algorithms for decision trees induction from data, probably the most known and used are ID3 and its enhanced version C4.5, Quinlan (10). ID3 searches through the attributes of the training instances and extracts the attribute that best separates the given examples. The algorithm uses a greedy search, that is, it picks the best attribute and never looks back to reconsider earlier choices. The central focus of the decision tree growing algorithm is selecting which attribute to test at each node in the tree. The goal is to select the attribute that is most useful for classifying examples. A good quantitative measure of the worth of an attribute is a statistical property called information gain, which measures how well a given

			Rules	Training Accuracy	Test			
					Accuracy	Specificity	Sensitivity	AUC
XCS	400	O	19 ± 10	87.6 ± 0.8%	79.2 ± 1.7%	88.6 ± 1.4%	61.8 ± 4.7%	75.2 ± 3.0%
		F	28 ± 8	84.1 ± 0.4%	70.7 ± 0.9%	82.4 ± 1.6%	49.0 ± 2.8%	65.7 ± 2.2%
	6400	O	42 ± 12	99.3 ± 0.4%	73.3 ± 2.9%	78.0 ± 3.1%	64.5 ± 3.9%	71.3 ± 3.5%
		F	58 ± 17	99.6 ± 0.6%	65.8 ± 1.3%	71.0 ± 1.3%	56.1 ± 2.7%	63.5 ± 2.0%
C4.5		O		79.8%	71.5%	80.5%	55.8%	68.2%
		F		88.2%	76.5%	85.7%	58.9%	72.3%
NN		F		99%	71.3 ± 0.7%	82.9 ± 0.4%	50.9 ± 3.0%	66.9 ± 1.7%
		F		90%	70.3 ± 1.2%	83.6 ± 0.9%	46.9 ± 3.8%	65.2 ± 2.3%
SVM		F		99%	75.5%	82.0%	63.8%	72.9%
		F		90%	74.7%	81.2%	63.3%	72.2%

Table I. Results from the experiments on the HSNCC dataset. (O and F in the second column stand respectively for original dataset and filled dataset)

attribute separates the training examples according to their target classification. This measure is used to select among the candidate attributes at each step while growing the tree.

Neural Networks

Neural Networks (NN) methods encompass a large sets of models and learning algorithms able to deal with different tasks and classes of data – see Haykin (11) and Bishop (12). The relevance of the neural approach in ML is due to its ability to capture underlying functional relationships in the data within regression, classification, and pattern recognition problems. Actually, many tasks have been solved using such models, with performances often at "the state of the art" in the machine learning area. An extended survey on the application of NN to medical analysis, both for diagnosis, prognosis and survival analysis functions, is provided in Lisboa (13). General advantages of neural networks approach rely on its properties of tolerance to the lack of data integrity (dealing with noise and uncertainty), i.e. the robustness property. Moreover, NN are universal approximators (Theorem of Cybenko, 1988) (11). Due to their flexibility, neural networks approach is often the first model used when dealing with new application problems characterized by poor background information. The hypothesis space considered by the neural model learning algorithms is the *continuous* space of all functions that can be represented by assigning the parameters values, i.e. the weight values of the given NN architecture. This allows the model to represent a rich space of non-linear functions, making neural networks a good choice for learning discrete and continuous functions whose general form is unknown in advance. Since the hypothesis space is a continuously parameterized space and error function is differentiable, the search of the best hypothesis, i.e. the learning algorithm, can be based on a gradient descent technique.

The main drawbacks of the neural network models are their poor interpretability of learned models and the dependency of the training process from the initial conditions, i.e. the initial weights values and the values of the parameters used to tune the learning algorithm. Moreover, the network topology is often built empirically, through a trial-and-error approach.

Support Vector Machines

Support Vector Machine (SVM) and kernel-based approaches have originated from the foundations of the statistical learning theory and have recently emerged as a major topic within the field of machine learning – see Vapnik (14), Burges (15), and Smola and Schölkopf (16).

Like Neural Networks, SVM can be used for pattern recognition, classification and regression tasks to learn from data an high-dimensional nonlinear hypothesis. The basic difference between SVM and other ML models similar to NN is in the inner methodology of learning, rather than in how they are applied.

The original linear machine built by SVM can be extended including nonlinearity (in the original input space) by implicitly embedding the data into an internal feature space through a *kernel* function. Given a training set of instance-label pairs $\{(\mathbf{x}_i, y_i) \mid i = 1,...,k\}$ where $x_i \in \mathbb{R}^n$ and $y \in \{1,-1\}$, the support vector machines require the solution of the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2} w^T w + C \sum_{i=0}^{k} 1 - \left| y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \right|_+$$

The training vectors $\mathbf{x}_i$ are mapped into a higher (possibly infinite) dimensional space by the function ϕ. Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. C>0 is the penalty parameter of the error term.

It should be noted that the definition of a kernel

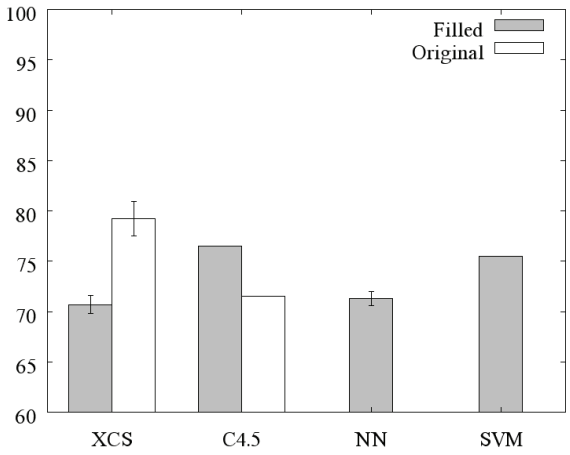

Fig. 1. Accuracy values (%)

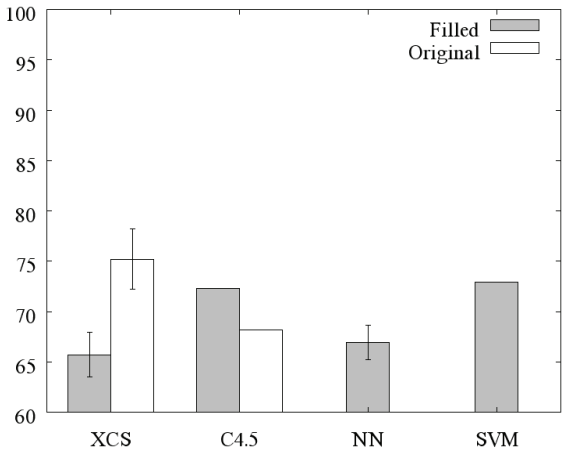

Fig. 2. Area under ROC curve values (%)

function corresponds to the definition of a prior similarity measure on data; hence, the choice of the proper kernel is a critical issue in the application of the method.

4. EXPERIMENTS

Missing Values

Many real world tasks in machine learning involve data sets with arbitrary patterns of missing data. This problem is especially found in the medical and biological context, where data gathering is often imprecise and error-prone. In particular in the dataset we are analyzing 21% of the genotype information was missing.

Among the methodologies considered in this study, XCS and decision trees are able to natively treat missing data, whilst SVM and neural networks are not. Thus, in order to apply the latter methods we need to devise methods to deal with the problem. The simplest option is to remove the rows with missing values; however, in our situation this procedure was not viable, since it would have reduced the dataset size by over 60%. We adopted then as a first approach two basic techniques for filling missing values, namely *mean imputation* (for continuous data) and *mode imputation* (for categorical data). With mean (resp. mode) imputation, means (resp. the most frequent values) from set of observed data are substituted for the unknown features.

The drawback from a statistical point of view is that the standard deviation of the sample is underestimated, so the results are somehow unreliable but provide a baseline in benchmarking model performance on the considered dataset.

For the sake of comparison, all the algorithms were tested on the filled dataset, while XCS and decision trees were also applied to the original dataset.

Experimental settings

Experiments with the four different algorithms were carried out by performing a 10-fold cross-validation on the HNSCC dataset. Results are reported in Table **I**. For the SVM and C4.5 algorithms figures are relative to a single execution, while for the XCS and neural networks they are the average and standard deviation of 10 runs, in order to take into account the stochastic nature of these algorithms.

For each algorithm the parameters employed in the experiments has been selected by performing some preliminary tests. In particular for the XCS algorithm, we present the results obtained by setting the maximum number of rules to 6400 and to 400. In fact this parameter has a major influence on data fitting: a high number of rules allows higher accuracy levels on the training set. Moreover, for comparison purposes, we adopted the early stopping technique to stop the training process at similar accuracy levels – 90% and 99% – for the algorithms which allowed it, namely neural networks and SVM.

In the following we report specific parameters and describe the peculiar representation employed with each algorithm.

XCS. In order to apply the XCS algorithm to the problem at hand, we adapted it to handle the type of information contained in the data set, which varies from binary (i.e. sex), to continuous-valued (i.e. age, indicators of smoking and alcohol-drinking habits), and to a special class data for the genotype. As for the integer and real data types, possible XCS implementations already exist in literature (Wilson (17) and (8) for instance). However, for the genotypic values we needed a slightly different treatment: nine of the genes considered have two allelic variants, thus we need three classes (considering also the heterozygote)

for the input values, but the classifiers have in fact to merge the heterozygote with either one of the homozygotes. So the values we used are the following: as input we have 0 for a_1a_1, 1 for a_1a_2, and 2 for a_2a_2; in classifiers 11 is not allowed, but we admit 01 (matching inputs 0 and 1), 12 (matching 1 and 2) and ## (matching all values).

Decision trees. We submitted our dataset to a standard implementation of the C4.5 algorithm with pruning. The pruning confidence factor was 25% with a minimum of 2 instances per leaf.

Neural network. The experiments were performed with a multi-layer perceptron with one non-linear (Tanh) hidden layer. The output layer was composed by two non-linear units, and the activation function used was the Log-SoftMax, defined as

$$\varphi(x_i) = \log\left(\frac{e^{x_i}}{\sum_j e^{x_j}}\right)$$

where i is the current unit, j varies on all the output units, and x_j is the net input of the j-th unit.
Data in the numeric columns of the dataset have been normalized to have 0 mean and variance 1. Binary data have been represented using a 1-of-2 (One-Hot) representation. For the genes with two allelic variants we used the following representation: $0 \rightarrow 100$, $2 \rightarrow 001$, and $1 \rightarrow 111$. The representation for 1, the heterozygote, was chosen to partially overlap those of the homozygotes.
The reported results have been obtained using a network with 14 neurons in the hidden layer and setting the learning rate to 0.01, with a weight decay parameter to control the model complexity.

Support vector machines. To perform the experiments with the SVM we used the same dataset prepared for the NN case. The SVM employed for the experiments used a radial basis kernel function, defined as

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{\sigma^2}\right)$$

The presented result have been achieved using C=28 and σ=13.

5. DISCUSSION

From a general point of view, the results we obtained convey the impression of a "difficult" data set: none of the tested methods was able to reach 80% accuracy. The imbalance between classes is apparent in the quite different figures for sensitivity and specificity; its impact on learning is also evident in the difference between the AUC and accuracy values. Moreover, this difference seems stable between the different approaches, which suggests that they are all similarly affected by class imbalance.
NN and SVM appear to be quite robust to overfitting: their performance on the test set continues to increase along with performance on the training set. This does not happen in XCS, where the 6400 results show a higher degree of overfitting than the 400 results.
Comparing the best test set accuracy, on the filled data SVM and decision trees have the best results — 75.5% and 76.5% respectively. XCS and NN on this dataset perform worse, attaining 70.7% and 71.3%. XCS' performance however appears to be severely impaired by the filling of missing values: on the original dataset in fact, the test result jumps to 79.2% — which is the best result reached on the HNSCC dataset. It is interesting to notice that decision trees suffer from missing data, suggesting their way to treat the issue is not as effective as XCS' one.
The performed work suggests how important a correct treatment of incomplete data can be in order to achieve good performances. In subsequent experiments we will considered more effective well-known methods to fill missing values, e.g. multiple imputation. Particular efforts will be focused on the study of ad-hoc kernel functions, able to deal with incomplete samples, to be used in conjunction with the standard SVM algorithm.
Our results clearly cannot be assumed to describe the general behavior of the tested algorithms on other data sets; however, they pose the basis for understanding strength and weaknesses of methodologies; this is useful not only in order to clarify the issues to expect when applying the same algorithms on other data sets in the biomedical setting, but also to suggest which characteristics can be more profitable to exploit when combining various methodologies together.

ACKNOWLEDGMENTS

This work has been carried out under the framework of the European Network of Excellence "Computational Intelligence for Biopattern analysis in Support of eHealthcare" (BIOPATTERN), Sixth Framework Programme Priority 2, Information Society Technologies.

REFERENCES

1. Maggini, V., Abbondandolo, A., Barale, R., Baronti, F., Monatti, S., Canzian, F., Casartelli, G., Guidi, L., Margarino, G., Mereu, P., Micheli, A., Passaro, A., Rossi, A.M., and Starita, A. , 2005, Tumori, I Supplementi, 4(2), 72

2. Passaro, A., Baronti, F., Maggini, V., Micheli, A., Rossi, A. M., and Starita, A., 2005, Proceedings of

MedGEC 2005, ACM online

3. Wilson, S. W., 1995, Evolutionary Computation, 3(2)

4. Wilson, S. W., 1998, Proceedings of Genetic Programming 1998, 665–674

5. Holland, J. H., 1976, "Adaptation", In Rosen, R. and Snell, F. M. editors, "Progress in theoretical biology", 4, New York, Plenum

6. Butz, M. V. and Wilson, S. W., 2001, *Proceedings of IWLCS 2000, Volume* 1996 of *LNAI,* 253–272

7. Wilson, S. W., 2001, IWLCS 2001, volume LNAI 2321, 197–210

8. Wilson, S. W., 2001, IWLCS 2000, volume LNAI 1996, 158–174

9. Quinlan, J. R., 1986. Machine Learning, 1, 81 – 106

10. Quinlan, J. R., 1993. "C4.5: programs for machine learning", Morgan Kaufmann Publishers

11. Haykin, S., 1999, "Neural Networks, A Comprehensive Foundation", Prentice Hall, 2nd edition

12. Bishop, C. M., 1995, "Neural Networks for Pattern Recognition", Oxford University Press

13. Lisboa, P. J. G., 2002, Neural Networks, 15(1), 11–39

14. Vapnik, V. N., 1995, "The Nature of Statistical Learning Theory", Springer-Verlag, New York

15. Burges, C. J., 1998, Data Mining and Knowledge Discovery, 2(2),121 – 167

16. Smola, A. and Schölkopf, B., 2002, "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond", MIT Press

17. Wilson, S. W., 2000, Learning Classifier Systems. From Foundations to Applications, LNAI 1813, 209–219