World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:1, 2016

Patient-Specific Modeling Algorithm for Medical Data Based on AUC

Guilherme Ribeiro, Alexandre Oliveira, Antonio Ferreira, Shyam Visweswaran, Gregory Cooper

Abstract—Patient-specific models are instance-based learning algorithms that take advantage of the particular features of the patient case at hand to predict an outcome. We introduce two patient-specific algorithms based on decision tree paradigm that use AUC as a metric to select an attribute. We apply the patient specific algorithms to predict outcomes in several datasets, including medical datasets. Compared to the patient-specific decision path (PSDP) entropy-based and CART methods, the AUC-based patient-specific decision path models performed equivalently on area under the ROC curve (AUC). Our results provide support for patient-specific methods being a promising approach for making clinical predictions.

Keywords—Approach instance-based, area Under the ROC Curve, Patient-specific Decision Path, clinical predictions.

I. Introduction

CLINICAL decision-making may be improved by using predictive models [1]. Predicting patient outcomes under uncertainty constitutes an important health care problem. Enhanced decision models can lead to better patient outcomes as well as efficient use of health care resources.

The typical paradigm in predictive modeling is to learn a single model from a database of patient cases, which is then used to predict outcomes for future patient cases [2]. This approach is known as *population-wide model* because it is intended to be applied to an entire population of future cases. Examples of popular population-wide methods are decision trees, logistic regression, neural networks and Bayesian networks.

In contrast to that general approach, a *patient-specific model* consists of learning models that are tailored to the particular features of a given patient. Thus, a patient-specific model is specialized to the patient case at hand, and it is optimized to predict especially well for that case. Moreover, patient-specific models can also be seen as examples of instance-based learning schemes, of which k-nearest neighbor, local regression and lazy decision trees are examples.

Instance-based algorithms learn a distinct model for a test instance and take advantage of the features in the test instance to learn the model [3]. Typically, the instance-based algorithms are lazy, since no model is constructed *a priori* before a test instance becomes available, and a model is inferred only when a prediction is needed for a new instance [4]. In contrast, algorithms that learn population-wide models are eager since such explicitly build a model by training data before a test instance becomes available.

There are several advantages of patient-specific models over population-wide methods. For instance, patient-specific models may have better predictive performance for taking advantage of any particular characteristic of the case at hand, whereas population-wide methods converge to an average method, derived for an entire population. Second, a patient-specific model structure is usually simpler than its population-wide counterpart. Thus, a patient-specific model can provide a more succinct explanation of its decision making. Third, the construction of a patient-specific models may be computationally faster, though this advantage is offset by the observation that a patient-specific method has to construct a distinct model for each patient case of interest while a population-wide method has to construct just a single model. Finally, the task of handling of missing features is simplified on patient-specific approach.

In this paper, we investigate the performance of two patient-specific methods, based on the lazy decision tree approach. We compare the performance of the AUC-based patient-specific methods with the entropy-based and CART models. We focus on the discriminating performance of the four methods and evaluate them using the area under the ROC curve (AUC) [5].

The remainder of this paper is organized as follows. Section II presents background and related work on instance-based methods. Section III provides details of the patient-specific decision path algorithms that we have developed. Section IV describes the datasets, shows experimental methods and presents and discusses the results of the patient-specific decision path algorithm on several datasets. Section V presents our conclusions.

II. Background

The canonical instance-based method is the nearest-neighbor technique, that is, when the most similar training instance to a given test instance is located its target value is returned as the prediction [6]. For a test instance, the k-Nearest Neighbor (KNN) method, for example, selects the k most similar training instances and either averages or takes a majority vote of their target values. Modified versions of kNN have been applied successfully to medical databases for diagnosis and knowledge extraction [7]. Other instance-based methods are not as reliant on a similarity measure as is the case for the nearest-neighbor methods, taking advantage of the values of the predictors in the test instance to learn a model.

The authors would like to thank Federal University of Maranhao and University of Pittsburgh for their structure and support and CAPES for it financial support.

Guilherme Ribeiro, Antonio Ferreira and Alexandre Oliveira are with the Federal University of Maranhao, Sao Luis, MA 65080-805 BRA (phone: +55 98-98234-0333, e-mail: guilherme.ufma@gmail.com).

Shyam Visweswaran and Gregory Cooper are with University of Pittsburgh.

One of the most popular algorithms of decision tree is CART [8]. This algorithm is based on population-wide model and it is used specially to explain and predict one outcome from collected data. This method allows to build individual groups that are characterized by the same values of attributes. Another peculiarity is that CART is a kind of binary tree. These characteristics make it easy to be interpreted.

Friedman et al. [4] have described the LazyDT method that searches for the best CART-like decision tree for a test instance. When compared to standard population-wide methods, for inducing decision trees, as ID3 and C4.5, LazyDT does not perform pruning, handles only discrete variables, and has higher accuracy on average.

Zheng et al. [9] have developed an instance-based method called the Lazy Bayesian Rules (LBR) learner that builds a rule tailored to the values of the predictors of the test instance, used to classify it. A LBR rule consists of: 1) a conjunction of the features (predictor-value pairs) present in the test instance as the antecedent, and 2) a consequent naive Bayes classifier that consists of the target variable as the parent of all other predictors that do not appear in the antecedent. The classifier parameters are estimated from those training instances that satisfy the antecedent. A greedy step-forward search selects the optimal LBR rule for a test instance to be classified. LBR uses values of predictors in the test instance to drive the search for a suitable model in the model space and, when compared to a variety of population-wide methods, LBR has reached higher accuracy on average.

Visweswaran and colleagues have developed and applied an instance-based algorithm that performs Bayesian model averaging over LBR models [2], [10], using the features of the test case to drive the search. The prediction for the test case target variable is obtained by combining the predictions of the selected models weighted by their posterior probabilities. This method has obtained higher accuracy than LBR on a range of non-medical datasets and also performed better than several population-wide methods on a pneumonia dataset, when evaluated within a clinically relevant range of the ROC curve. Furthermore, instance-based algorithms that use the test instance to drive the search over a space of Bayesian network models have been developed and applied to patient-specific modeling with good results [11], [12].

Ferreira et al. [13] have developed patient-specific decision path (PSDP) algorithms that can build a decision path to predict patient outcome. Given a patient for whom the values of the features are known, these algorithms construct a decision path using a subset of those features. Two selection criteria were investigated for selecting features: balanced accuracy and information gain. Results obtained with those algorithms using clinical datasets were compared with CART using the AUC metric.

III. Patient-Specific Decision Path Algorithms based on AUC

The proposed patient-specific decision path algorithm uses AUC as a metric to select patient's features that will compose the path [14]. The Area under the ROC Curve (AUC) is a widely used measured of performance of supervised classification rules. It has the attractive property of circumvent the need to specify misclassification costs.

Algorithm 1 Algorithm of decision models to patient-specific, that use AUC-Split metric (PSDP-AUC-Split).

Input: $labels$, $dataset$, $testset$
repeat
  for $i = 1$ **to** $size(testset)$ **do**
    $subset = getsubset(dataset, testset(i))$.
    $diffset = dataset - subset$

    $partition(1) = sum(subset_+, diffset_+)$
    $partition(2) = sum(subset_-, diffset_-)$

    {$descend$, in sort function (below), means the way of ordering the partitions}
    $coordinates = sort(partition,' descend')$

    fpr = *false positive rate based on coordinates matrix*
    tpr = *true positive rate based on coordinates matrix*
    AUC = $trapz(fpr, tpr)$ {$trapz$ calculate the area of an trapezoid, through of rates extracted from $coordinates$ matrix}

    $AUC_{vector}(i)$ = AUC
    if $AUC < AUC_{best}$ **then**
      $AUC_{best} = AUC$
    end if
  end for
  $path$ = *selected attribute according with* AUC_{best}
  $dataset$ = *dataset according with the selected attribute*
  $dataset\_diff$ = *dataset without instances according selected attributes*
  $eProb_{path}$ = *calculate the prediction probability of one label belong to positive or negative class*
  $predictedLabel = max(eProb_{label})$
until $dataset$ to be empty, $dataset$ belongs only an unique class or the number of attributes of the $dataset$ to be empty
Output: $eProb_{label}, predictedLabel$ and $path$

The use of AUC as a metric to select attributes in a decision tree was introduced by Ferri and colleagues [15], [16]. Based on the optimal choice of possible labels of the tree, the AUC-split criterion leads to good AUC values, without compromising the accuracy if a single labeling is chosen. Thus, for a two class classification problem and a tree with n leaves, there are 2^n possible labels, of which $n + 1$ are optimal. Such optimal labeling gives the convex hull for the considered leaves. Algorithm 1 shows the pseudo code of the patient-specific decision path that uses the AUC-split proposed by Ferri.

In contrast, another way to use the AUC metric to select patient features requires the prediction of a class probability or some other score as proposed by [17].

In this case, a leave-one-out cross-validation scheme was employed in order to generate the probability estimate and further calculating a Mann-Whitney-Wilcoxon test statistic, which directly related to the AUC. To avoid over fitting, Laplace smoothing [18] was employed when estimating class probabilities. The patient-specific decision path that uses this standard approach just described is shown in the pseudo code of Algorithm 2.

Algorithm 2 Algorithm of decision models to patient-specific, that use AUC metric (PSDP-STD-AUC).

Input: $labels$, $dataset$, $testset$
repeat
  $eProb$ = *calculate the probability of happens one of the two classes to each one of the dataset instances*

  for $i = 1$ **to** $size(testset)$ **do**
    $subset = getsubset(dataset, testset(i))$.
    $diffset = dataset - subset$

    $eProb(subset)$ = *calculate the probabilities according with* $testset$
    $eProb(diffset)$ = *calculate the probabilities for different data of the* $testset$

    $AUC = colAUC(trueProbability, labels)$

    $AUC_{vector}(i)$ = AUC
    if $AUC < AUC_{best}$ **then**
      $AUC_{best} = AUC$
    end if
  end for
  $path$ = *selected attribute according with* AUC_{best}
  $dataset$ = *dataset according with the selected attribute*
  $dataset\_diff$ = *dataset without instances according selected attributes*
  $eProb_{path}$ = *calculate the prediction probability of one label belong to positive or negative class*
  $predictedLabel = max(eProb_{label})$
until $dataset$ to be empty, $dataset$ belongs only an unique class or the number of attributes of the $dataset$ to be empty
Output: $eProb_{label}, predictedLabel$ and $path$

IV. Experimental Results

In this section, we present the computational experiments performed to validate the proposed methods, as well as the employed datasets, the performance measures and algorithm parameter settings used for evaluation.

The proposed patient-specific decision path algorithms were implemented in [19] and all computational experiments were performed on a Intel Core i5 personal computer, with frequency of 2.5GHz, 8GB of RAM and running the the Mac OS X 64-bit Yosemite.

A. Datasets

Concerning the datasets, there are *UCI clinical datasets*, including two on heart disease, two on diabetes and one cancer patients. Brief descriptions of the datasets are given in Table I.

In addition to UCI datasets, two others about heart failure are also used. These datasets were provided by the *University of Pittsburgh*. A brief description of them is given below.

Heart failure is a problem that affects approximately 5 million people in the United States. This disease has taken about one million people to hospital each year due to a primary discharge diagnosis of heart failure and approximately two million with a secondary discharge diagnosis of same condition. All data about heart failure was collected by 192 general acute care hospitals in Pennsylvania in 1999, consisting of heart failure patients who were hospitalized from the Emergency Departments. Both datasets sum up $11,178$ cases and 21 variables, including electrocardiograph, radiographic, clinical and other data collected.

The outcome of heart failure, datasets are binary and express two results: first, the occurrence of death during the hospitalization and second, the development of one or another serious medical complications during the hospitalization, including death.

When pre-processing the data, continuous variables were discretized using the entropy-based method developed by Fayyad and Irani [20]. Missing values were imputed using an iterative non-parametric imputation algorithm described by Caruana [21] which has previously been applied to fill in missing predictor values for medical datasets with good results.

B. Test Settings

The algorithms were evaluated using 20-fold cross-validation, i.e, each dataset were randomly divided into 20 approximately equal sets, such that each set has a similar proportion of individuals who developed the positive outcome. For each algorithm, 19 sets were combined and evaluated on the remaining test set. This procedure was repeated for each possible test set. A prediction for the outcome variable was obtained for every instance in a dataset. The final result of the algorithms are presented in terms of AUC, processing time and path length. The AUC algorithms were compared with the PSDP-Entropy algorithm proposed in [13].

C. Results

Table II shows the AUCs obtained by the four algorithms. For each dataset, we present the mean AUC based on the 20-fold cross-validation and the respective confidence intervals at the 0.05 level. Overall, the two AUC based split algorithms perform comparably to the PSDP-Entropy and CART methods. Except for the Tic-tac-toe dataset, there is no statistically significant difference between the four methods.

In the tic-tac-toe dataset, the PSDP-STD-AUC performed better, with a mean AUC of 0.98. ANOVA analysis [22] reveal that there is not statistical significant different between the four methods ($p >> 0.05$).

TABLE I
BRIEF DESCRIPTION OF USED UCI DATASETS

Dataset	Instances	Attributes	Positive Cases (%)
Australian	690	14	44%
Breast	699	9	34%
Cleveland	296	13	45%
Corral	128	6	44%
Crx	653	15	45%
Diabetes	768	8	35%
Flare	1066	10	17%
Glass 2	163	9	47%
Heart	270	13	44%
Pima	768	8	35%
Sonar	208	60	53%
Tic-tac-toe	958	9	65%
Vote	435	16	61%

TABLE II
THE AUC RESULTS FOR THE *UCI datasets*. FOR EACH ALGORITHM THE TABLE GIVES THE MEAN AUC OBTAINED FROM 20-FOLD CROSS-VALIDATION ALONG WITH 95% CONFIDENCE INTERVALS. STATISTICALLY SIGNIFICANT MEAN AUC ARE IN BOLD

Datasets	CART	PSDP-Entropy	PSDP-STD-AUC	PSDP-AUC-Split
Australian	0.880 [0.852,0.908]	0.919 [0.910,0.928]	0.896 [0.877,0.916]	0.889 [0.869,0.909]
Breast	0.952 [0.931.0.974]	0.984 [0.980,0.988]	0.985 [0.978,0.992]	0.983 [0.975,0.991]
Cleveland	0.835 [0.769,0.901]	0.862 [0.837,0.888]	0.845 [0.779,0.911]	0.834 [0.773,0.895]
Corral	0.985 [0.971,1.000]	1.000 [1.000,1.000]	1.000 [1.000,1.000]	1.000 [1.000,1.000]
Crx	0.887 [0.853,0.921]	0.920 [0.911,0.929]	0.879 [0.855,0.903]	0.885 [0.861,0.909]
Diabetes	0.818 [0.780,0.856]	0.827 [0.812,0.842]	0.815 [0.781,0.850]	0.820 [0.785,0.855]
Flare	0.692 [0.646,0.739]	0.730 [0.717,0.745]	0.718 [0.680,0.757]	0.715 [0.681,0.749]
Glass 2	0.848 [0.764,0.931]	0.831 [0.795,0.867]	0.870 [0.794,0.946]	0.865 [0.785,0.945]
Heart	0.840 [0.783,0.897]	0.879 [0.862,0.898]	0.877 [0.832,0.921]	0.877 [0.831,0.923]
Pima	0.825 [0.794,0.855]	0.825 [0.810,0.840]	0.813 [0.769,0.858]	0.811 [0.781,0.841]
Sonar	0.844 [0.784,0.903]	0.889 [0.867,0.911]	0.862 [0.818,0.907]	0.887 [0.835,0.939]
Tic-tac-toe	0.920 [0.899,0.942]	0.960 [0.952,0.969]	**0.989** [0.978,1.000]	0.977 [0.968,0.986]
Vote	0.966 [0.934,0.999]	0.986 [0.982,0.990]	0.985 [0.970,1.000]	0.985 [0.975,0.995]

Table III shows the results obtained by four algorithms to heart failure datasets. The two AUC algorithms perform comparably to the entropy and CART methods. In this case, it was make the ANOVA analysis and we verify that there is not statistical significant difference between the four methods too ($p >> 0.05$).

Table IV shows the running time of the proposed algorithms (means pm standard deviation). Each approach was run 30 times for each dataset. Since the entropy based and CART algorithms require less operations, they presented better mean running time when compared to the other two methods. The PSDP-STD-AUC requires an estimation of the class probabilities, which demands several extra computational operations. Even though the PSDP-AUC-Split does not require class probability estimation, however it needs to sort leaves for obtaining the convex hull, for each split, also demanding extra CPU time. As per an ANOVA analysis, we verified that there is significant difference on running time, since the entropy-based model runs faster ($p << 0$).

The same experiment was made for heart failure datasets. The results are quite similar to *UCI datasets* ones: there is meaningful running time difference between PSDP-Entropy and CART algorithms compared to PSDP-STD-AUC and PSDP-AUC-Split algorithms, evidencing differences in algorithm complexities. For these datasets, the ANOVA analysis shows that there is statistical significant difference between the four methods too ($p << 0$). Another result obtained by the algorithms that can be analyzed is the complexity of the generated models, considering the path length to reach a label or diagnosis. Table V shows the path length of the generated models for UCI datasets.

PSDP-based algorithms generate specific paths for each instance and an only one average can be calculated for each approach. However, CART algorithm generate decision trees and we fairly introduce two columns taking in account the worst and the best way for reaching labels in the decision trees.

CART_B algorithm has obtained, on average, smaller paths than the other algorithms, for all UCI datasets, except for datasets Breast and Corral, in which PSDP-STD-AUC has reached the best averages, 2.193 and 3.000 respectively.

Table VI shows the results of complexity of the models generated for the two heart failure datasets. Once again, path length averages of CART algorithm take in account the worst and the best cases for obtaining a label in the decision tree.

The results presented by Table VI show that the PSDP-STD-AUC algorithm has obtained, on average, smaller paths than the other algorithms for heart failure datasets. The difference was evident when compared with another approach instance-based (PSDP-Entropy) and when compared with the population-based approach, even in the best case. One can see that there are evidences that the PSDP-STD-AUC algorithm

TABLE III
THE AUC RESULTS FOR THE HEART FAILURE DATASETS. FOR EACH ALGORITHM THE TABLE GIVES THE AUC MEAN, OBTAINED FROM 20-FOLD CROSS-VALIDATION ALONG WITH 95% CONFIDENCE INTERVALS. STATISTICALLY SIGNIFICANT MEAN AUC ARE IN BOLD

Algorithms	HEART_D	HEART_DC
CART	0.682[0.651,0.712]	0.652[0.635,0.670]
PSDP-Entropy	0.697[0.672,0.722]	0.664[0.651,0.678]
PSDP-AUC-Split	0.702[0.671,0.732]	0.636[0.620,0.651]
PSDP-STD-AUC	0.709[0.684,0.735]	0.652[0.635,0.670]

TABLE IV
RESULTS OF EXECUTION TIME FOR PROPOSED ALGORITHMS, COMPARED TO CART AND PSDP-ENTROPY ALGORITHMS

Datasets	CART	PSDP-Entropy	PSDP-STD-AUC	PSDP-AUC-Split
Australian	0.443±0,019	3.346±0.046	35.821±1.464	7.388±0.138
Breast	0.376±0,006	1.140±0.010	10.102±0.048	2.115±0.025
Cleveland	0.361±0,006	1.091±0.011	10.827±0.052	2.466±0.026
Corral	0.327±0,004	0.170±0.005	1.651±0.019	0.341±0.009
Crx	0.433±0,012	3.342±0.032	34.111±0.094	7.620±0.229
Diabetes	0.384±0,005	1.974±0.013	17.305±0.094	4.002±0.040
Flare	0.447±0,008	4.348±0.021	36.206±1.179	8.019±0.322
Glass 2	0.330±0,005	0.427±0.007	3.298±0.038	0.920±0.014
Heart	0.351±0,005	1.042±0.011	10.301±0.057	2.220±0.019
Pima	0.383±0,006	1.973±0.012	19.175±0.093	3.781±0.137
Sonar	0.372±0,004	3.730±0.029	34.608±0.134	8.040±0.038
Tic-tac-toe	0.430±0,006	2.527±0.013	26.731±0.124	4.647±0.029
Vote	0.358±0,005	1.366±0.017	14.300±0.089	2.686±0.018

TABLE V
RESULTS OF PATH LENGTH FOR ALL ALGORITHMS TESTED AT THIS WORK, FOR UCI DATASETS. THE CART_W IS THE TRADITIONAL CART, CONSIDERING THE WORST CASE, I.E. HIGHER LEVEL OF THE DECISION TREE. CART_B IS ALSO THE TRADITIONAL CART, HOWEVER CONSIDERING THE BEST CASE, I.E., LOWER LEVEL OF THE DECISION TREE (FIRST LEAF FOUND). FOR EACH CASE (BEST AND WORST ONES), THE PATH LENGTH AVERAGES WAS CALCULATED

Datasets	CART_W	CART_B	PSDP-Entropy	PSDP-STD-AUC	PSDP-AUC-Split
Australian	12.050[9,17]	**2.000**[2,2]	6.162[2,13]	4.801[3,9]	7.007[2,14]
Breast	6.450[5,10]	2.850[2,3]	2.331[1,6]	**2.193**[1,4]	2.381[1,6]
Cleveland	7.700[6,9]	**2.750**[2,3]	5.804[3,12]	2.895[2,8]	5.959[3,13]
Corral	4.000[4,4]	**3.000**[3,3]	3.890 [3,5]	**3.000**[3,3]	3.890[3,15]
Crx	11.400[10,14]	**2.000**[2,2]	6.058[2,14]	4.578[3,7]	6.934[3,15]
Diabetes	7.200[7,8]	**2.950**[2,3]	5.967[2,7]	3.804[2,6]	6.656[3,8]
Flare	12.200[10,15]	**2.000**[2,2]	7.429[2,9]	3.257[2,6]	8.134[3,10]
Glass 2	5.000[5,5]	**2.000**[2,2]	7.429[2,9]	3.061[2,4]	8.134[3,10]
Heart	5.650[5,7]	**2.800**[2,3]	6.607[3,12]	3.555[2,6]	6.618[3,13]
Pima	7.450[7,8]	**2.950**[2,3]	5.998[2,7]	3.786[2,6]	6.690[3,8]
Sonar	6.500[6,8]	**2.050**[2,3]	6.442[2,59]	3.125[2,5]	5.831[3,56]
Tic-tac-toe	9.750[9,10]	**3.000**[3,3]	4.696[3,7]	3.037[2,6]	4.750[3,7]
Vote	5.500[4,6]	**1.950**[1,2]	3.112[2,11]	2.845[2,6]	3.087[2,13]

TABLE VI
RESULTS OF PATH LENGTH FOR ALL ALGORITHMS TESTED AT THIS WORK, FOR PITTSBURGH DATASETS. THE CART_W IS THE TRADITIONAL CART, CONSIDERING THE WORST CASE, I.E. HIGHER LEVEL OF THE DECISION TREE. CART_B IS ALSO THE TRADITIONAL CART, HOWEVER CONSIDERING THE BEST CASE, I.E., LOWER LEVEL OF THE DECISION TREE (FIRST LEAF FOUND). FOR EACH CASE (BEST AND WORST ONES), THE PATH LENGTH AVERAGES WAS CALCULATED

Algorithms	HEART_D	HEART_DC
CART_W	16.250[14,19]	20.800[18,24]
CART_B	3.950[3,4]	4.400[4,5]
PSDP-Entropy	11.002[3,19]	9.592[3,16]
PSDP-AUC-Split	11.139[5,20]	9.291[4,17]
PSDP-STD-AUC	**3.412**[2,10]	**2.875**[2,6]

may generate less complex models or, at least, shorter paths than its counterparts.

V. CONCLUSION

We have introduced PSDP, a new patient-specific approach for predicting outcomes, based on the AUC metric to select

patient attributes. We evaluated this method on several datasets, including medical data. Computational experiments have shown that the PSDP-AUC based methods perform equivalently to the state-of-art-based methods concerning AUC mean, in 20-fold cross-validation with 95% confidence interval. However, regarding the path lengths obtained for each instance, the PSDP-STD-AUC has presented very competitive results, specially in larger datasets.

Patient-specific models may have better predictive performance for taking advantage of any particular characteristic of the case at hand, providing a more succinct explanation of its decision, once they construct a distinct model for each patient case of interest..

In future work, we plan to deal with some of limitations of the proposed method and to evaluate the effect of different discretization methods. Besides, we also intent to extend the performance evaluation of both proposed approaches for large datasets.

REFERENCES

[1] Abu-Hanna, A. and P. J.:Lucas, Prognostic models in medicine. AI and statistical approaches. Methods of Information in Medicine, 2001. 40(1): p. 1-5.
[2] Visweswaran, S. and G.F. Cooper.: Patient-specific models for predicting the outcomes of patients with community acquired pneumonia. In AMIA Annu Symp Proc. 2005.
[3] T. M. Mitchell.: Machine Learning. 1st. ed, McGraw-Hill, Inc., New York, NY, USA, 1997.
[4] J. H. Friedman.: Lazy decision trees. In Proceedings of the thirteenth national conference on Artificial intelligence, v.1 (AAAI'96), v.1 AAAI pp. 717-724, 1996.
[5] Foster J. Provost, Tom Fawcett, and Ron Kohavi.: The Case against Accuracy Estimation for Comparing Induction Algorithms. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98), San Francisco, CA, USA, 445-453, 1998.
[6] Cover, T. and P. Hart.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 1967. 13(1): p. 21-27.
[7] Gagliardi, F.: Instance-based classifiers applied to medical databases: diagnosis and knowledge extraction. Artif Intell Med, 2011. 52(3): p. 123-39.
[8] L. Breiman and J. Friedman and R. Olshen and C. Stone: Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA, 1984.
[9] Zheng, Z.J. and G.I. Webb.: Lazy learning of Bayesian rules. Machine Learning, 200. 41(1): p. 53-84.
[10] Visweswaran, S. and G.F. Cooper.: Instance-specific Bayesian model averaging for classification. In Proceedings of the Eighteenth Annual Conference on Neural Information Processing Systems. Vancouver, Canada, 2004.
[11] Visweswaran, S., et al.: Learning patient-specific predictive models from clinical data. J Biomed Inform, 2010. 43(5): p. 669-685.
[12] Visweswaran, S. and G.F. Cooper.: Learning instance-specific predictive models. Journal of Machine Learning Research, 2010. 11(Dec): p. 3333-3369
[13] Ferreira A, Cooper GF and Visweswaran S.: Decision path models for patient-specific modeling of patient outcomes. Proceedings of the Annual Symposium of the American Medical Informatics Association (2013) 413-21. PMID: 24551347. PMCID: PMC3900188
[14] David J. Hand and Robert J. Till.: A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Mach. Learn. 45, 171-186, 2, October, 2001.
[15] Csar Ferri and Peter Flach and Jos Hernndez-Orallo.: Learning Decision Trees Using the Area Under the ROC Curve. Proceedings of the 19th International Conference on Machine Learning, Pages 139-146, Sydney, NSW, Australia, July 8-12, 2002.
[16] Csar Ferri and Peter Flach and Jos Hernndez-Orallo.: Rocking the ROC Analysis within Decision Trees. Technical Report, December, 20, 2001.
[17] Tom Fawcett.: An Introduction to ROC Analysis. Elsevier Science Inc. New York, USA, June, 2006.
[18] Bianca Zadrozny and Charles Elkan.: Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01). San Francisco, CA, USA, 609-616, 2001.
[19] MATLAB: Version 8.4.0 (R2014b) The Mathworks Inc., 2014. Accessed: 02-17-2015
[20] Fayyad, U. M. and Irani, K. B.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. 'IJCAI' , pp. 1022-1029, 1993.
[21] Caruana, R.: A non-parametric EM-style algorithm for imputing missing values. in Proceedings of Artificial Intelligence and Statistics. 2001.
[22] Janez Demšar.: Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7, 1-30, December, 2006.