COMP3308 Report

Topic Category: Computer Science

Sub-categories: A.I., Machine Learning, Medicine

Report Title:

An Evaluation Of Different Predictive Classifiers In The Diagnosis of Disease.

Author | SID: 308201809 | The University of Sydney |
Date: 24/05/2019

Aim:

Innovation and improvements in machine learning are quickly rising, as new problems are solved using AI. The growth of this field of Computer Science has led to a plethora of approaches to choose from. This study aims to compare the performances of a number of classifying algorithms, in analyzing numerical medical data with predictive and diagnostic power. Particular attention is paid to contrasting an example of supervised and unsupervised machine learning. This can serve as a valuable guide in deciding which algorithm to implement, given time or memory constraints of a particular setting. Quick diagnostic tools have potential for widespread use in medical practices, while future large scale studies need to decide if previously successful approaches are applicable, or innovation is required. This is an important choice when considering the complexity of Neural Network Design, both in optimizing accuracy and time cost.

Data:

The study used a modified data set originally obtained from The National Institute of Diabetes and Digestive and Kidney Disease. It consisted of 768 samples with 8 numerical attributes and a resulting 9th probability coinciding with diagnosis of Diabetes Mellitus. The 9th numerical value was replaced with a binary valued class representing a positive or negative diagnosis. All samples were full; No attribute values were missing. All participants which formed the data set were 21 years of age or older, female, residents near Phoenix, Arizona, USA and of Pima Indian Heritage.

The following are the attributes in the data set as provided by Prof. Irena Koprinska (Pima Indians Diabetes Database (modified), 2015):

1. Number of times pregnant
2. Plasma glucose concentration after 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable ("yes" or "no")

Samples which had class evaluation of "no" (negative diagnosis) totaled 500, while the remaining 268 resulted in a "yes" classification.

The data was normalized between 0 and 1 before being run through the classifiers. Additionally, each algorithm used the data set described above, and another version with decreased dimensionality. The additional version was obtained through Correlation-based Feature Selection (CFS) using a Greedy Hill climbing Best First search, that augments its path with backtracking. Of the 8 numerical attributes, the following 5 were chosen:

1. Plasma glucose concentration
2. 2-Hour serum insulin
3. Body mass index
4. Diabetes pedigree function
5. Age

Results:

The following algorithms were run on the original and reduced data set using Weka's 10-fold cross validation, and performances were evaluated:

- ZeroR (zero rule majority)
- OneR (one rule)
- 1NN (k-nearest neighbors where k = 1, d is Euclidean)
- 5NN (k-nearest neighbors where k = 5, d is Euclidean)
- NB (Naive Bayes)
- DT (Decision Tree)
- MLP (Multilayer Perceptron)
- SVM (Support Vector Machine)
- RF (Random Forest)

Additionally, KNN and NB were implemented independently in accompanying MyClassifier.py, and evaluated using 10-fold cross validation in MyEvaluator.py. The following tables show evaluation of performance as a percentage of correct classifications.

Table 1.0

Weka	**ZeroR**	**OneR**	**1NN**	**5NN**	**NB**	**DT**	**MLP**	**SVM**	**RF**
no feature selection	65.10	70.83	67.84	74.48	75.13	71.75	75.39	76.30	74.87
CFS	65.10	70.83	69.01	74.48	76.30	73.31	75.78	76.69	75.91

Table 1.1

MyClassifier	**My1NN**	**My5NN**	**MyNB**
no feature selection	68.36	75.65	75.01
CFS	68.49	74.87	75.78

μ = 72.80 δ = 3.60

The table below ranks the algorithms by overall accuracy

Table 1.2

Rank	With CFS	No feature Selection
1	SVM	SVM
2	NB	My5NN
3	RF	MLP
4	MyNB	NB
5	MLP (equal 4th)	MyNB
6	My5NN	RF
7	5NN	5NN
8	DT	DT
9	OneR	OneR
10	1NN	My1NN
11	My1NN	1NN
12	ZeroR	ZeroR

Weka's fold models were used to obtain SVM and NB (with CFS) fold accuracies and evaluate the difference in their performances between these top 2 performers. The results are shown below. Weka output files can be found in /WekaFiles/ directory within the submission.

Table 2.0

Fold	NB with CFS	SVM with CFS	$\lvert d_{Fold} \rvert$
1	79.22	76.62	2.60
2	83.12	84.42	1.30
3	75.32	75.32	0.00
4	80.52	83.12	2.60
5	79.22	79.22	0.00
6	71.43	72.73	1.30

Fold	NB with CFS	SVM with CFS	$\|d_{Fold}\|$
7	74.03	74.03	0.00
8	70.13	66.23	3.90
9	75.00	77.63	2.63
10	75.00	77.63	2.63

μ_d = 1.70 δ_d = 1.38 Z = 1.70 $\pm$ 1.04 at 95% confidence interval.

The same procedure was used to obtain relevant data for DT before and after CFS was applied.
Table 2.1

Fold	DT	DT with CFS	$\|d_{Fold}\|$
1	77.92	79.22	1.30
2	76.62	79.22	2.60
3	70.13	75.32	5.19
4	72.73	70.13	2.60
5	75.32	74.03	1.29
6	62.34	62.34	0.00
7	68.83	68.83	0.00
8	70.13	71.43	1.30
9	68.42	77.63	9.21
10	76.32	75.00	1.32

μ_d = 2.48 δ_d = 2.80 Z = 2.48 $\pm$ 2.11 at 95% confidence interval.

The table below uses precision and recall For a performance measure. Values were obtained from confusion matrices produced by Weka and MyEvaluator. All algorithms used CFS and are ordered by performance F1 ranking.

Table 3.0

Algorithm	True Positives	False Positives	False Negatives	Precision	Recall	F1
5NN	160	88	108	72.58%	59.70%	65.44%
RF	166	83	102	66.67%	61.94%	64.22%
MLP	172	90	96	65.65%	64.18%	64.19%
My5NN	165	90	103	64.71%	61.57%	63.10%
NB	153	67	115	69.55%	57.09%	62.71%
DT	171	108	97	61.29%	63.81%	62.52%
MyNB	150	68	118	68.81%	55.97%	61.72%
SVN	142	53	126	72.82%	52.99%	61.34%
OneR	127	83	141	60.48%	52.61%	56.28%
1NN	149	119	119	55.60%	55.60%	55.60%
My1NN	148	122	120	54.81%	55.22%	55.01%

μ_{F1} = 61.10 δ_{F1} = 3.54

μ_{Recall} = 58.24 δ_{Recall} = 3.99

Discussion:

A General Comparison of Results

All algorithms returned accuracies between 65.10% and 76.69%, showing a performance range of 11.59%, where μ = 72.80% and population δ = 3.60%. The top performer was in line with expectations, being SVM, but still within the first standard deviation. However, this superiority was significant when compared to the runner up. The bottom performers were also as expected, being ZeroR followed by 1NN. However, these algorthms' performances placed them below the first standard deviation; ZeroR and 1NN may be overly simplistic for such use cases. In comparing Weka's and MyClassifier's implementation of KNN and NB, the largest difference was 1.17% where MyClassifier outperformed Weka. These variations are most likely due to differences in the implementation of 10-fold cross validation, as folds are nonidentical. Table 1.0 and 1.1 contain the full results.

To gain a more complete picture an F1 measure using recall and precision was also used to evaluate performance on all algorithms with CFS. However, ZeroR was left out of this evaluation. All algorithms returned accuracies between 55.01% and 65.44%, showing a performance range of 10.43% where μ = 61.10% and δ = 3.54%. This method of comparison revealed 5NN to be the top performer and the only algorithm above the first standard deviation. The bottom performers remained consistent with expectations and significantly reduced the mean performance. The largest difference between MyClassfier's implementations and Weka's, was 2.34%, where now Weka outperformed MyClassifier. One remarkable discovery was SVN's relatively poor performance. While it remained in the 1st standard deviation, it occupied the lowest rank after the consistently poor performers. Table 3.0 contains the full set of results.

The Efficacy of CFS

The three algorithms ZeroR, OneR and 5NN showed no improvement from CFS. Additionally, MyClassifier's 5NN degraded by 0.78% in overall performance. However, when using F1 as a measure, 5NN with CFS ranked 1st in performance. Algorithms with an overly simplistic method of classification do not benefit from CFS. This is due to the method of classification being independent from attributes lowly correlated to class. For example in ZeroR, removal of any attribute will not change the majority class. In OneR, the rule for classification is based on an attribute with high correlation to the majority class, making it an unlikely candidate for removal by CFS. kNN will be discussed in more detail in the next section.

All other methods showed some improvement, the greatest of which was a 1.56% increase for DT. Notably, NB had an improvement of 1.17% which gave it an overall score equivalent to SMV's before CFS. Table 2.0 shows the differences in fold accuracy in DT with and without CFS. The change in performance was found to be statistically significant using a sample standard deviation. The higher increase in performance experienced by DT may be due to its preference for compact decision trees (I. Koprinska, 2015).

The attributes selected aligned with intuitive predictions; Plasma Glucose levels and Serum Insulin are highly specific attributes, and are depended on by practitioners for diagnosis. Additionally, genetic predisposition and BMI are usually strong indicators of risk. However, Diastolic Blood pressure is

related to a range of Medical conditions which would likely decrease correlation with class. Further, blood pressure and skin fold tests may highly correlate with BMI, increasing their redundancy. Pregnancy, may have a relatively low correlation to class and while age may be highly related to other attributes, it may be a strong enough indicator of risk for inclusion. CFS seems to be a highly reliable method of efficiently increasing performance.

Varying k in kNN

As mentioned, kNN uniquely performed poorly in response to CFS. However, this is not a consistent characteristic for all values of k. Interestingly, the performance of 5NN is not dramatically increased when increasing the size of k, and k = 5 is a reasonable value to use due to diminishing returns. However, when CFS is used, a significant improvement can be obtained. The value k = 10 is commonly used in commercial packages (I. Koprinska, 2015). Combining this k value with CFS yielded a performance of 77.70%, which placed it just ahead of SVM in first position. These results can be obtained by changing the value of k in line 291 of MyEvaluator.py, and running the script with pima-CFS.csv. The flexibility of kNN and its relative performance make it an appealing choice.

Noting DT's Performance

DT is often a highly viable option for use cases similar to that being assessed. The decision trees may assist practitioners or researchers reason about predictions made, due to their high readability. This is important when dealing with the complexity of the human body and assists in investigation of the attributes themselves. Unfortunately it's performance was unimpressive according to the sum of average accuracies. The entropy of the data set with respect to the class is:

$$H(S) = I(\tfrac{268}{768}, \tfrac{500}{768}) = -(\tfrac{268}{768} \cdot \log_2 \tfrac{268}{768} + \tfrac{500}{768} \cdot \log_2 \tfrac{500}{768}) = 0.93 \text{ bits}$$

The data set is well distributed across class categories such that there is little improvement to be made with a different sample. Despite its performance seeming relatively low, measures of F1, recall and precision paint a different picture.

Context Related Considerations

The importance of recall vs precision is relevant to this type of study, particularly if a model is being built to assist in diagnosis and screening individuals. Table 3.0 reveals MLP to be the most advantageous with respect to recall, closely followed by DT. Contrastingly SVM performed significantly poorly, falling below the standard deviation of 3.99%. It reveals that given this particular data set, SVM does very well at correctly categorizing negatives, while DT excels at identifying true positives. A higher emphasis should be placed on recall over precision, as consequences of a missed diagnosis are significant. This analysis shows some justification for using MLP as it performs relatively well across all measures, including precision. However, DT should also be considered given its readability. In the context of research, the emphasis on recall may not be appropriate. SVN and 5NN had the highest precisions, which could be useful when trying to evaluate a range. For example, They could be used to derive a minimum financial burden on a country, due to a medical condition.

Conclusion:

The discussion above served as a guide for choosing an algorithm in medical diagnostic applications. While SVN is often an attractive choice and MLP can be an avenue for original innovative solutions, simple options seem to have great returns proportionally. NB with CFS is a highly favorable candidate when considering its performance and ease of implementation. KNN with CFS is also a viable option, and k values can be manipulated if time and space constraints abide. Meanwhile, the potential advantages of MLP are visible and DT carries some advantages that are related to the context of the study.

It may be worth while to assess whether accuracies below 77% are sufficient for a given purpose. If improvements can be made beyond this, investigation into innovative methods may be favorable. One potential avenue of enquiry, is use of classifier ensembles with boosting; The current performance values with the data, are sufficient to make them candidates for such an approach. Other possibilities include use of deep learning neural networks for optimization of MLP when exposed to data figures, or use of auto encoders to develop a model that can identify related medical illnesses which arise before the onset of other conditions (though this is not an avenue which has classically benefited from such an approach). Finally, there may be benefit in implementing models, using weighted attributes with careful consideration.