Computational intelligence and asthma

Pablo Vicente-Munuera
Bioinformatics MSc
Email: pablovm1990@gmail.com

Abstract—**The aim of this work is apply some algorithms typical used in statistics and machine learning to a given dataset. With this, we'll try to find if some of these algorithms can classify our dataset and try to figure something out. The dataset selected for this purpose is an asthma dataset.**

I. Introduction

Asthma is a chronic (long-lasting) inflammatory disease of the airways. In those susceptible to asthma, this inflammation causes the airways to spasm and swell periodically so that the airways narrow. The individual then must wheeze or gasp for air. Obstruction to air flow either resolves spontaneously or responds to a wide range of treatments, but continuing inflammation makes the airways hyper-responsive to stimuli such as cold air, exercise, dust mites, pollutants in the air, and even stress and anxiety.

A few keys from AAAAI (American Academy of Allergy Asthma and Immunology)[3] who collects all of this information:

1) From 2001 through 2009 asthma rates rose the most among black children, almost a 50% increase, in U.S.A. [4]
2) More than half (53%) of people with asthma had an asthma attack in 2008. More children (57%) than adults (51%) had an attack. 185 children and 3,262 adults died from asthma in 2007, in U.S.A. [4]
3) An estimated 300 million people worldwide suffer from asthma, with 250,000 annual deaths attributed to the disease.[5]
4) About 70% of asthmatics also have allergies.[5]
5) It is estimated that the number of people with asthma will grow by more than 100 million by 2025. [5]

But it's not just a problem of the (illogically) called "first world". It also affects to other undeveloped continents such as the African one [6]:

- In 2010, 49.7 million (13.9%; 95% CI 9.6-18.3) among children ¡15 years, 102.9 million (13.8%; 95% CI 6.2-21.4) among people aged ¡45 years, and 119.3 million (12.8%; 95% CI 8.2-17.1) in the total population.

A. Dataset

The dataset chosen is the one used by Voraphani N. (2014)[1], in which the main purpose was to identify differentially expressed genes. These genes belongs to different subjects with different classes:

- Control: Subjects with no asthma.
- MMA: Subjects with mild-moderate asthma.
- SA: severe asthmatic patients.

All of these classes represent an expression array with bronchial epithelial cells of Homo Sapiens.

Because we get an expression file, we had to convert it to an arff file (used by weka [17]). In order to get this file, we have to do some operations before we could some research about it. The first thing, and the most important thing, you must do when you get an expression file, is the normalization of the data. With R and bioconductor [**?**] (citation here), this purpose is solved. We do the normalization with the background of the array with R. We could, also, have done differential expression in order to obtain just a few genes (differentially expressed over the others).

After all of this, we obtain 43377 (without array controls) genes and 108 subjects. With this dataset we'll do all of the classification's problems.

II. Methods

As said before, the main purpose is classify well and see which genes could contribute more than others We'll analyze these genes so as to see if any of those important genes to our classification problem, have the same relevancy in biology, and, being more specific, in asthma.

Why machine learning & statistics and not other methods? Well, machine learning and statistics are so powerful. That could be a double-edged sword, but in this case it is not. Although, machine learning, by now, is widely used in the field of bioinformatics and biology.

A. Supervised classification

In a classification problem, we have a set of elements divided into classes [7]. Given an element (or instance) of the set, a class is assigned according to some of the element's features and a set of classification rules. In our case, we have three classes (Control, MMA, SA) and 108 instances (subjects with asthma or not). Our subjects are labelled with their own class. So we proceed to divide the dataset in two subsets: the training dataset and the test dataset. The training one will be the input (labelled as well) of the classifier. The classifier learn to classify this training dataset, and the output will be a model. With this model, we run again the classifier, but, at this time, we input the test dataset without labels (i.e. no classes). After all of this procedure, we'll obtain a percentage of how good is our classifier.

In order to reduce the bias with the division of the dataset (in training and test), we'll execute the k-fold-cross-validation [8]. In this case, the dataset is partitioned into k folds. Each fold is left out of the design process and used as a testing set. The estimate of the error is the overall proportion of the errors committed on all folds.

We've used several paradigms (classifiers) to see which could be fitted more to our data. In the next subsections, they are going to be explained.

1) Naive Bayes: It is built upon the assumption of conditional independence of the predictive variables given the class.

$$c^* = \arg\max_c p(C = c) \prod_{i=1}^{n} p(X_i = x_i | C = C) \tag{1}$$

Which is the reduced formula of this one:

$$\gamma(x) = \arg\min_k \sum_{c=1}^{r_0} co(k, c) p(c|x_1, \ldots, x_n) \tag{2}$$

, in which every variable depends on all other and the complexity of the algorithm is too complicated. Naive Bayes (Equation:2) gives an approximate result, by reducing the dependencies between each variable, and it comes, also, with a time-relaxed version of the algorithm.

2) KNN, K-nearest neighbours (IBk): Imagine a classification problem, in which you have to divide in (known) classes, seeing only a surface with points and crosses. A way to do that, is starting in one of them and going through all of them and classify each one by assigning it to the label most frequently represented among the k nearest samples. And you'll have solved the problem by k-nearest neighbours.

3) Decision tree: C4.5 (j48): The decision tree is as simple as get a tree in which all the leaf nodes are classes and the inner nodes are decision parameters that will help us to determinate whether is one class or another (in the case, there are just two classes). The particularity of C4.5[10] is that the decision parameters are obtained via information gained ratio:

$$I(X_i, C)/H(X_i) \tag{3}$$

4) Logistic regression: It is based on the logistic function: $f(z) = 1\frac{1}{1+e-z}$. So the equation used here is:

$$P(C = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \Sigma_{i=1}^{n} \beta_i x_i)}} \tag{4}$$

where x represents an instance to be classified, and $\beta_0, \beta_1, \ldots, \beta_n$ are the parameters of the model. These parameters should be estimated from the data in order to obtain a concrete model.

5) Bayesian Networks, TAN: TAN is based on the mutual information of each variable (or group of variables) with everyone and trying to maximize this number choosing the right variables. So, the mutual information between two variables is given by:

$$I(X, Y) = \sum_{i=1}^{r_x} \sum_{j=1}^{r_y} p(x_i, y_j) log \frac{p(x_i, y_j)}{p(x_i) p(y_j)} \tag{5}$$

it measures the reduction of uncertainty of one variable knowing the other one. So, the algorithm consists in building the tree of the variables or group of variables, order by the information gained, which are maximum at this mutual information, until all of the next combinations of variables would not increase the mutual information gained by the last group.

B. Complementary

1) Random forest [11]: Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them.

2) Adaboost.M1 [13]: It is a boosting algorithm that can be used to significantly reduce the error of any learning algorithm that consistently generates classifiers whose performance is a little better than random guessing. It consists in obtain the algorithm of a "weak learner", such as a random line to divide two classes, and combined their outputs into a weighted sum that represents the final output of the boosted classifier.

3) Multilayer perceptron: If the data can be separated perfectly into two groups using a hyperplane, it is said to be linearly separable [12]. It turns out that if the data is linearly separable, there is a very simple algorithm for finding a separating hyperplane. This algorithm is the perceptron. And in this case the multilayer perceptron, it's an specific case of a neural net. It uses the sigmoid function instead of another one. The sigmoid function is defined by:

$$\frac{1}{1 + e^{-x}} \tag{6}$$

4) RBF Network: Instead of using the sigmoid function, the RBF network uses a radial basis function (RBF). So it is a neural net, but using a RBF kernel. The parameters that such a network learns are the centers and widths of the RBFs and the weights used to form the linear combination of the outputs obtained from the hidden layer. Another difference with the multilayer perceptron is that this network doesn't look for an hyperplane to divide linearly the subjects. The RBF network looks for a hypersphere or hyper ellipsoid.

C. Feature subset selection (FSS)

One of the main problems of the machine learning algorithms is the complexity problem. You usually get stacked because the algorithms require time and resources, and, very frequently, a lot of each one. So, one of the solution that could be provided is the FSS (feature subset selection). Mainly, FSS reduces the number of variables (or parameters) of our dataset, and the dimensionality associated. Which transform our dataset into a easy one or easier, at least. There three approaches, and we're going to explain each one, in the few next sections.

1) Univariate: It is based in evaluate each feature separately, thereby ignoring feature dependencies [14]. In our case we use the gain ratio, evaluating the worth of an attribute by measuring the gain ratio with respect to the class. Which is defined by:

$$G(Class, Attribute) = \frac{(H(Class) - H(Class|Attribute))}{H(Attribute)} \tag{7}$$

where H means the mutual information.

2) Multivariate: We use CFS (Correlation-based feature selection [15]), it differs from univariate FSS in the part of the redundancy. It also looks for redundant features and ignore them because they will be highly correlated with one or more of the other features. CFS uses a search algorithm along with a function to evaluate the merit of feature subsets like *genetic* algorithm or just a *best first*.

3) Wrapper [16]: To achieve the best possible performance with a particular learning algorithm on a particular training set, a feature subset selection method should consider how the algorithm and the training set interact. We explore the relation between optimal feature subset selection and relevance. Our wrapper method searches for an optimal feature subset tailored to a particular algorithm and a domain.

III. Results

In this section we'll show all the results we get from the expression set used. The software used to get all of those results is weka [17]. And all of the results are compared by the ROC curve [18], which is a trusty measure.

TABLE I
*: The X values means no results could be obtained due to computational problems. **The K used here is equal to one.

			FSS		
Method	No filter	text	Univariate	Multivariate	Wrapper
Naive Bayes	0.436		0.626	0.623	0.731
KNN**	0.594		0.644	0.67	0.781
Logistic	X*		0.731	0.7	0.793
Decision Tree	0.486		0.639	0.698	0.641
Bayesian Net	X*		0.657	0.663	X*

We had these problems mentioned before because the initial dataset had, approximately, 43300 variables. One of the first approaches to solve this, was a first univariate filter in which we selected the information gain ratio with respect to their own class. After this filter, all of these classify problems became more time-relaxed, due to the reduction of the number of variables (43300 to 32).

Another important thing which showed up with the output of the results was the classification between one class and the other ones.

TABLE II
*: K equals to 4; ** K equals to 1.

	Multivariate		
Method	CFS Best first	CFS Genetic	Wrapper
Naive Bayes	0.623	0.599	0.74
KNN	0.773*	0.674*	0.687**
Logistic	0.699	0.675	0.824
Decision Tree	0.668	0.76	0.641
Bayesian Net	0.667	0.674	0.678

TABLE III
Results depending the K value on the KNN classifier

K	Univariate	Multivariate
1	0.644	0.67
2	0.735	0.72
3	0.73	0.745
4	0.744	0.773
5	0.762	0.807
8	0.77	0.771

IV. Discussion

References

[1] Voraphani N, Gladwin MT, Contreras AU, Kaminski N et al. An airway epithelial iNOS-DUOX2-thyroid peroxidase metabolome drives Th1/Th2 nitrative stress in human severe asthma. Mucosal Immunol 2014 Sep;7(5):1175-85.

[2] ”Asthma.” MedlinePlus. January 16, 2009 [cited January 20, 2009]. http://www.nlm.nih.gov/medlineplus/asthma.html .

[3] American Academy of Allergy, Asthma, and Immunology (AAAAI) 555 East Wells Street, Suite 1100, Milwaukee, WI 53202-3823. Telephone: (414) 272-6071. http://www.aaaai.org.

[4] Centers for Disease Control and Prevention, Vital Signs, May 2011.

[5] World Health Organization. Global surveillance, prevention and control of chronic respiratory diseases: a comprehensive approach, 2007.

[6] Adeloye, Davies et al. An Estimate of Asthma Prevalence in Africa: A Systematic Analysis. Croatian Medical Journal 54.6 (2013): 519531. PMC. Web. 21 Apr. 2015.

[7] ñaga, P., Calvo. B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J.A., Armañanzas, R., Santafé, G., Pŕez, A. et al. (2006). Machine Learning in Bioinformatics. Briefings in Bioinformatics, 17(1), 86-112.

[8] Stone M. Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society Series B 1974;36:11147.

[9] Minsky M. Steps toward artificial intelligence. Transactionson Institute of Radio Engineers 1961;49:830.

[10] Quinlan R. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[11] Breiman L., Random Forests. Machine Learning. 45(1); 5-32. 2001.

[12] Witten I, Frank E, Hall M (2011) Data mining: practical machine learning tools and techniques, 3rd edn. Morgan Kaufmann, Burlington

[13] Freund Y., Schapire R.E.: Experiments with a new boosting algorithm, 13th International Conference on Machine Learning, 148-156, 1996

[14] Saeys Y., Inza I., and Larrañaga P., A review of feature selection techniques in bioinformatics, Bioinformatics (2007) 23 (19): 2507-2517 first published online August 24, 2007, doi:10.1093/bioinformatics/btm344

[15] Hall, M. A. 1998. Correlation-based Feature Selection for Machine Learning. Ph.D diss. Dept. of Computer Science, Waikato Univ.

[16] Kohavi R, John G. Wrappers for feature subset selection. Artificial Intelligence 1997;97(12):273324.

[17] Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations. Department of Computer Science. University of Waikato. New Zealand.

[18] Green DM, Swets JA. Signal Detection Theory and Psychophysics. Wiley, 1974.

TABLE IV
Best results on the complementary methods. *: Search method: best first; †: Search method: Genetic.

		FSS		Filtered by gain ratio		
Method	None	Univariate	CFS*	CFS*	CFS†	Wrapper
Adaboost(randomForest)	0.573	0.801	0.795	0.798	0.829	0.835
Adaboost (J48)	0.544	0.758	0.736	0.736	0.8	X
Multilayer perceptron	X	0.788	0.792	0.778	0.697	0.83
RBF Network	0.393	0.675	0.726	0.726	0.693	X
Random forest	0.569	0.811	0.783	0.803	0.809	0.843

TABLE V
Some examples of the difference between classes. Measured by the ROC curve.

	Classes			
Method - Filter	Control	MMA	SA	Overall
KNN (k=1) - Gain ratio	0.642	0.589	0.717	0.644
Bayesian Net - Gain ratio & CFS	0.802	0.591	0.697	0.667
Logistic - Wrapper	0.82	0.755	0.829	0.793
Decision Tree - No filter	0.55	0.473	0.471	0.486
Naive Bayes - CFS Best first	0.766	0.534	0.664	0.623