An ensemble to predict the survival of patients with colon cancer based on the optimal model for imputation

Yingkai XU
University of Nottingham
Department of Computer Science
Nottingham, NG7 2RD, UK
Email: kevinhsu1008@gmail.com

***Abstract*—Computers find some data difficult to learn from. This results in classification algorithms performing only slightly better than random guessing. When this is the case, ensemble learning which means combining lots of different methods into one is often appropriate. In terms of medical prediction, an ensemble to predict colon cancer based on immunological laboratory data is developed to improve the predicting performance by dealing with collected raw data. Predicting how long potential patients can live (Survival) and what stage their colon cancers at (Tnm4cat) is the main purpose. We use decision trees C4.5 as classifier and compare simple modified imputation, multiple imputation and iterative method based on multiple imputation which can effectively solve unbalanced datasets and missing values issues.**

Keywords—missing values, unbalanced datasets, colon cancer, decision trees.

I. INTRODUCTION

Computers find some data difficult to learn from. This results in classification algorithms performing only slightly better than random guessing. When this is the case, ensemble learning which means combining lots of different methods into one is often appropriate. In this project, an ensemble to predict cancer stages based on immunological laboratory data is developed to improve the predicting performance by dealing with raw data. Predicting how long potential patients can live (Survival) and what stage their colon cancers at (Tnm4cat) is the main purpose of the project. It can provide more intelligent advice for patients in terms of historical data. Three main issues are solved. They are unbalanced datasets, missing values and ineffective classifier respectively. These problems occurs in collected raw data of colon cancer very commonly.

II. PROPOSE THE PROBLEM

According to medical standard, it usually divides survival into two classes. One is survival time of patients less than 60 months, the other is those whose survival time over 60 months. The input indicators including ages, gender and many other features of symptom are all the columns in the dataset from cd46 except Chemo, Radio, Radiorec and Chemorec.

A. Ineffective Classifiers

In terms of medical prediction, a well implemented neural network or SVM is usually applied to classification. It should always outperform decision trees classifiers like C4.5, but the major benefit of decision trees C4.5 is their transparency. The dataset of colon cancer is middle-scale and not very sufficient, so applying neural network and SVM probably leads to overfitting performance. Therefore, in this project, we use decision trees C4.5 as the classifier.

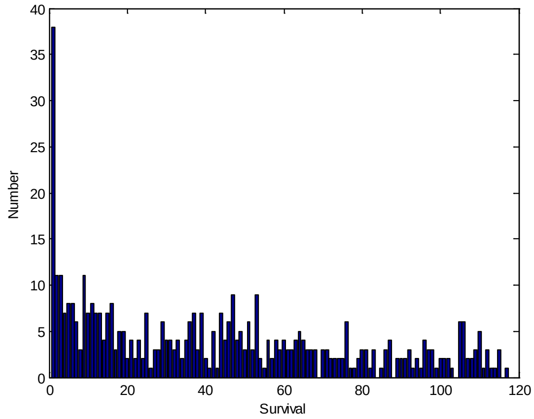

Fig. 1. Distribution of attribute 'Survival' for all instances

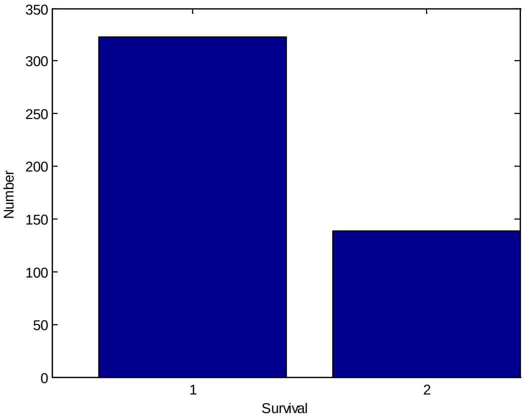

Fig. 2. Distribution of two classes of attribute 'Survival'

B. Unbalanced Datasets

There is a problem with unbalanced datasets. The distribution of two classes is obvious unbalanced which is shown in *Fig. 1* and *Fig. 2*. The unbalanced dataset will make the result of predicting model unrepresentative because the gap of scale is huge. Underrepresented class contributes less to the overall error which may affect the setting of parameters in the model when training.

C. Missing Values

The problem of missing values is the other serious one. When collecting these medical measuring data, Using the median/mean/mode imputation to replace missing values directly is one approach but it has major problems if too many values are missing and it has different types of data in the datasets like categorical and ordinary. These normal ways can fill in the blanks while losing statistical features of datasets. It may not reflects the real distribution as the given part.

III. Decision Trees C4.5

Decision trees are one of the simplest and yet most successful forms of learning algorithms. A decision tree takes as input an object or situation described by a set of attributes and returns the predicted value for the input. Several methods can be applied to select attribute at each node, e.g. the ID3 algorithm based on information impurity. Suppose the training set contains *p* positive and *n* negative examples. Then an estimate on the information contained in a correct answer is

$$I(\frac{p}{p+n},\frac{n}{p+n}) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n} \quad (1)$$

Testing of any attribute F will divide the training set E into subsets $E_1,...,E_v$ according to their values for F, where F can have *v* distinct values. Each subset *Ei* has p_i positive examples and n_i negative examples, so going along that branch, $I(p_i/(p_i+n_i), n_i/(p_i+n_i))$ bits of information will be needed to answer the question. So on average, after testing attribute F, we will need

$$\mathrm{Re}\,mainder(\mathrm{F}) = \sum_{i=1}^{v}\frac{p_i+n_i}{p+n}I(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i}) \quad (2)$$

bits of information to classify the examples. The information gain from the attribute test is the difference between the original information requirement and the new requirement

$$Gain(\mathrm{F}) = I(\frac{p}{p+n},\frac{n}{p+n}) - \mathrm{Re}\,mainder(\mathrm{F}) \quad (3)$$

Finally, the attribute that is chosen is the one with the largest gain.

The rule of choosing attribute and its best threshold of each node is shown as followed. When the number of data is big enough which is more than D (e.g. D=10) where D is the number of subsets, the algorithm will sort the data of chosen feature first and divided them into D segments. Use the upper bounds and lower bounds as thresholds to find out which one can obtain larger gain. Then choose this threshold and return the corresponding information gain of the chosen attribute. However, if the number of data is not big enough which is less than D, there is no need to consider data distribution. We divide the range of data from maximum to minimum into D segments equally and find out which threshold can lead to better gain. This approach can find approximate threshold faster. For example, if the value of one attribute varying from 0 to 100, the given testing threshold of this attribute will be 10, 20, 30, 40, 50, 60, 70, 80 and 90 when D=10.

IV. Solve Unbalanced Dataset

As for a full dataset without missing values, two ways can readily solve unbalanced issue. One way is to scale desired output which can change the value of error of cost function. This is very effective if the approach takes the exact value of output into consideration. For example, SVM and regression can apply this method. For decision trees C4.5, the node is decided by impurity directly which will not consider the exact value of output, so the desired output has no effect on solving unbalanced dataset. The other potential problem of scaling desired output is that trying to predict more than 2 separated classes (e.g. 4 classes) in the linear scale is quite dangerous. An example looks like this:

- 0 for the minimum under-represented value
- [0.5 + (minimum under-represented value/over-represented value)]
- [1.5 + (second minimum under-represented value/over-represented value)]
- [2.5 + (third minimum under-represented value/over-represented value)]

We must ensure that the relationship is linear. This is not obvious to detect if losing background information. Therefore, when we use scaling desired outputs to predict Tnm4cat, it is not a good choice.

Therefore, it would be better to randomly replicate less representative dataset after imputation process. The procedure is calculating the total number of unique instances for each class and delete those instances which are identical but belongs to different classes, randomly replicating instances which are selected in the same class for those less-representative dataset and using the new balanced dataset to run the previous classification algorithm.

V. Multiple Imputation

Multiple Imputation is suitable for the situation of missing completely at random (MAR). We assume the missing values occurs randomly.

1. Impute missing values using an appropriate model that incorporates random variation between maximum and minimum.

2. Do this M times (usually 3-5 times), producing M 'complete' datasets.

3. Perform the desired analysis on each data set using standard complete-data methods.

4. Average the values of the parameter estimates across the M samples to produce a single point estimate.

Considering that some are categorical and some are ordinary, we only fill with integer in those categorical features. It can achieve the average performance via randomly choosing missing values in the range.

VI. Complete-case Data Analysis combined with Modified Simple Imputation

In features processing, we use a fixed percentage to filter those features losing too many values. Thus, complete-case data analysis is the next step to filter those instances with many missing values and variation. For example. two instances have the identical values of features, but they belongs to different categories. Modified Simple Imputation has one modification. Simple Imputation just use mean values to fill in blanks. However, this project has to consider using mode values to complete categorical features. When filling mean value into blanks, its variation is also considered to be retained. The actual values are mean plus a random value which satisfies its variation distribution.

VII. Iterative method based on correlation of different features

Let Multiple Imputation as an initial step to set initial values in blanks because it can reflect average performance of random imputation. Now, assume this is already a complete-case dataset.

1. Select those features and target without missing values in set A and those features with missing values in set B.

2. Sort features in set B by the quantity of missing values and select the one (let's say P) with fewest missing values. Find which one (let's say Q) in set A has the highest correlation with P. After that, re-fill in all blanks of P based on the correlation. If feature P has very low correlation below the threshold, put P in set C and P will be excluded in following steps.

3. Put P in set A and delete P in set B.

4. Repeat 1 to 3 steps until there is no more features in set B.

5. Put set A, B and C together and Repeat 1 to 4 until the dataset is convergent.

The flow chart of this iterative approach is constructed in *Fig. 3.*

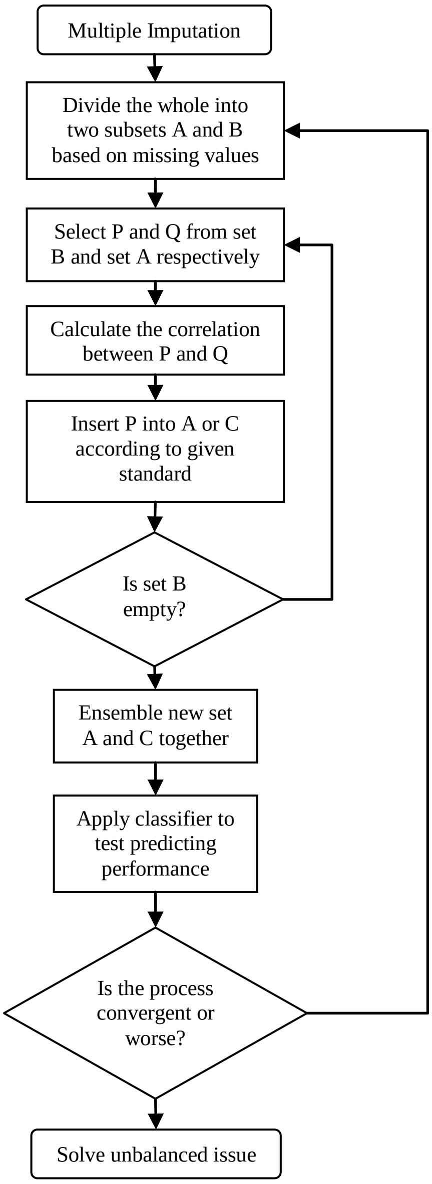
Multiple Imputation
Divide the whole into two subsets A and B based on missing values
Select P and Q from set B and set A respectively
Calculate the correlation between P and Q
Insert P into A or C according to given standard
Is set B empty?
Ensemble new set A and C together
Apply classifier to test predicting performance
Is the process convergent or worse?
Solve unbalanced issue

Fig. 3. The flowchart of iterative method based on correlation of different features.

This iterative method is an self-enhancement algorithm. That means the iterative may lead to much better performance or much worse performance because high correlation with one feature may lead to lower correlation with the target. Thus, we insert a detector to evaluate whether it goes in the right direction after a constant number of iterations.

VIII. Results and discussion

The predicting performance of complete-case data analysis combined with modified simple imputation is hugely improved. The result of cross-validation with 10 folds (90% for training and 10% for testing) is shown in Table 1. The equation for F_1_measure is

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Table 1. The performance of survival prediction by Decision Tree C4.5 based on complete-case data analysis combined with modified simple imputation

Recall	Precision	Accuracy	F_1_measure
0.8200	0.8276	0.8200	0.8238

Survival	Accuracy
0-60	0.7440
60-120	0.8960

Firstly, it indicates that if the collected data is balanced, decision trees C4.5 as the classifier has a very good performance to predict survival of patients. If just using multiple imputation and then solve unbalanced issue, the predicting performance is in Table 2:

Table 2. The performance of survival prediction by Decision Tree C4.5 based on multiple imputation

Recall	Precision	Accuracy	F_1_measure
0.7450	0.7200	0.7256	0.7323

Survival	Accuracy
0-60	0.6480
60-120	0.8032

After comparing multiple imputation with modified simple imputation, it implies that missing values may not miss at random because the former one replaces them with mean and variation which contains its own statistical features, it has better predicting results than multiple imputation. The average performance by purely guessing and selecting is worse than modified simple imputation.

Table 3. The performance of survival prediction by Decision Tree C4.5 based on iterative method

Recall	Precision	Accuracy	F_1_measure
0.7950	0.7790	0.7600	0.7869

Survival	Accuracy
0-60	0.6850
60-120	0.8350

Using iterative way shows a better improvement of predicting performance than purely multiple imputation. However, because of the same reason as multiple imputation, its results cannot outperform modified simple method. Besides, some of attributes may not satisfy linear relationship which will lead to the invalidation of this method based on correlation. We need to detect whether iterative steps turn to better which spends a lot of computing memory and time cost during the procedure.

Acknowledgment

I would like to sincerely thank my supervisor Prof. Uwe Aickelin and Dr. Chris Roadknight who always had time for me. Without their invaluable advice, guidance and supervision, the quality of this paper cannot reach such high satisfaction. I appreciate the wisdom of their ways in both technical and theoretical perspectives that motivated me to think forward and concentrate on my project deeply.

References

[1] D. Powers, "Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markdness and Correlation ", Journal of Machine Learning Technologies, Vol .2, Issue 1, 2011, pp.37-63.

[2] G. Le, Giang, "Machine Learning with Informative Samples for Large and Imbalanced Datasets", unpublished.

[3] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection", Journal of Machine Learning Research, Vol.3, 2003, pp. 1157-1182.

[4] R. Duda, Pattern Classification, 2nd ed, Wiley-Balckwell, 2004, Chapter 8.1-8.3.

[5] S. Buuren, Flexible Imputation of Missing Data, Chapman and Hall/CRC Press, 2012, pp. 25-52.