Prediction of Within 30 Days Re-admission For Patients Hospitalized with Diabetes by Libsvm

Hecheng Sun
University of Rochester
hsun17@u.rochester.edu

ABSTRACT

In this project, I incorporated the Diabetes 130-US hospitals for years 1999-2008 Data Set from UCI Machine learning Repository[1] (https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008) and an implementation of SVM, Libsvm[2] (https://www.csie.ntu.edu.tw/~cjlin/libsvm), to predict whether a patient hospitalized with diabetes will be readmitted within 30 days, given all his or her other attributes in the database. The main inspiration of the data pre-processing came from a research paper by Beata Strack and Jonathan P. DeShazo[3] (https://www.hindawi.com/journals/bmri/2014/781670/#B14). Depending on the choices of class labels, kernel functions, and weights, the final result varies from 91% to 60% in accuracy.

The specific steps for running this program are in a separate file - README.txt.

Data analysis:

The original database contains 101,766 encounters, and each encounter includes 50 attributes and 1 class label. The goal is to let the support vector machine learns the pattern of these encounters so that given a new encounter, the model can give a correct prediction of class labels.

Data preprocessing:

As described in the research paper, the original database needs a series of data-preprocessing steps before being put into a machine learning algorithm. First, the database has many attributes that have high missing rates. Examples are Weights and payer code. At the same time, some attributes are inconsequential to the prediction, such as the Encounter ID and the Patient number.

Next, some patients have visited more than once, which entails the same patient number in different encounters. In order for each encounter to be as statistically independent as possible, only the first encounter of each patient is considered. The rest of his or her encounters are deleted from the database. Furthermore, it is evident that a diseased patient will inevitably unable to be readmitted in the future, so the instances where the discharge disposition is hospice or death must also be removed from the database. These processes result in a decrease of encounters from 101,766 to 69,973.
In the end, the remaining attributes are race, gender, age, discharge disposition, admission source, time in the hospital, medical specialty, diag_1, A1Cresult, and change.

One more thing to notice is that libsvm can only take in numerical data, while there are a few nominal attributes in the final database that cannot be transformed into simple numbers, such as race, medical specialty, and diagnose. To fix this problem, a technique called One Hot Encoding is used. Each attribute is transformed into multiple binary attributes, depending on the number of possible values. For example, the possible values for the race are African American, Caucasian, Other, and Missing. Therefore, the race attribute is expanded into 3 attributes, each can only be -1 or 1. [1, -1, -1] means African American; [-1, 1, -1] means Caucasian; [-1, -1, 1] means Other; and [, , ,] means Missing. Be aware that since the goal is to predict whether a patient will be readmitted within 30 days, the previous three classes were converted into two classes, "-1" class for ">30" and "None", "1" class for "<30".

Data transformation methodology:

First, using Excel or number, one can delete the headers and the columns for unnecessary attributes, leaving only race, gender, age, discharge disposition, admission source, time in the hospital, medical specialty, diag_1, A1Cresult, change, and readmitted. One exception is patient numbers, which will be used later to delete duplicate encounters.

Second, one can run transform3.py to transform the old database into a digitalized database. The program contains one function for each nominal attribute. It also uses a list to record every patient id, so that whenever it reads in a new encounter with the same patient id, that encounter will be discarded. Therefore, the program can transform the diabetic_data3.csv into digitalized3.csv.

After the digitalized3.csv is generated, one can use Excel again to delete the now useless last column --- patient id. One also need to move the newly last column --- the column for readmission --- to

the first column. This step is for the next transformation from the CSV file to the format suitable for libsvm.

Next, one can run csv2libsvm.py[5] by Zygmunt (https://github.com/zygmuntz/phraug), to convert digitalized3.csv to svmliblized3.data (python csv2libsvm.py digitalized3.csv svmliblized3.data). The new data file should conform to the necessary format for libsvm, with the class label at front and attribute indexes in an increasing order from 1 to 31.

One now can copy svmliblized3.data to the tools folder in the libsvm folder and uses checkdata.py to check the format (python checkdata.py svmliblized3.data). If the program says no error was detected, run subset.py to divide the database into two parts, a training set and a testing set (python subset.py svmliblized3.data 60000 my.train my.test).

Now the preprocessing steps are complete. One can copy my.train and my.test to the root of the libsvm folder to train the SVM and test the result.

Experiment on class labels, kernel functions, and weights:

One easy observation of the final training and testing set is that the "-1" class dominates the database. The cases where the class label is “1” only hold about 11.1% of the time. Therefore, the SVM can predict the class label to be "-1" and be correct most of the time. This is indeed what will happen if one run the default training parameters (./svm-train -m 10000 my.train my.model) (./svm-predict my.test my.model output). The result is 90.6748% in accuracy, and 0% in precision and recall.

One solution is to adjust the weight of class “1” (./svm-train -m 10000 -w1 8 my.train my.model), which will make the mistake of misclassification of class "1" more significant than the misclassification of class "-1". The result is 72.596% in accuracy, 13.2191% in precision, and 34.8387% in recall. One step further is to use, instead of the default radial basis function, a linear kernel like u'*v, which has a better performance in an unbalanced database like this one (./svm-train -m 10000 -w1 8 -t 0 my.train my.model). The result of that is 63.1204% in accuracy, 12.6021% in precision, and 49.7849% in recall.

What about the database with the original class label? One can slightly modify the tranform3.py to make the class label of “<30”, “>30”, and “None” to be “0”, “1”, “2” respectively. After all the similar preprocessing steps, training (./svm-train -m 10000 my_old7.train my_old7.model), and testing (./svm-predict my_old7.test my_old7.model output) described above, the result is 60.5735% in accuracy.

Conclusion:

This problem is a classic example of predicting class label in an unbalanced database. The accuracy is high, but the precision and recall can be as low as 0. Manipulating the weights and the kernel function can help, only to a certain extent. However, with an AUC = 0.592172, it is still much better than the random chance. A research paper by Damian Mingle[4] (https://pdfs.semanticscholar.org/6b48/f265e574e73afc16daf7d88611a97beabe69.pdf) shows a result AUC = 0.56, which is lower than the result achieved in this paper.

For the data with original labels, even though the 60% in accuracy seems to be low, it is still almost twice as much as the accuracy of random chance of three classes classification.

In summary, the above data-preprocessing techniques and libsvm successfully predict whether a patient hospitalized with diabetes will be readmitted within 30 days, given all his or her other attributes in the database.

REFERENCES

[1] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, “Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records,” BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.

[2] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[3] Beata Strack, Jonathan P. DeShazo, Chris Gennings, et al., “Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records,” BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014. https://doi.org/10.1155/2014/781670.

[4] Damian M. Predicting Diabetic Readmission Rates: Moving Beyond Hba1c. Curr Trends Biomedical Eng & Biosci. 2017; 7(3): 555707. DOI: 10.19080/CTBEB.2017.07.555715

[5] Zygmuntz. “Zygmuntz/Phraug.” GitHub, 15 Nov. 2015, github.com/zygmuntz/phraug.