KE5206 SVM CA

Team 23: Bui Kim Dung (E0146998)

SUMMARY

The purpose of this CA project is to fit SVM model for prediction of default of customers in "No" and "Yes", using data set **Default** from library ISLR.

We have trained different SVM models using Gaussians and Polynomial kernel, with 3 different combinations of input variables. In this report, we will summarize our findings and understanding from training and analyzing performance of different models.

ANALYSIS

Using R, we can have an overview look at the data set:

```
> summary(Default)
 default    student       balance           income
 No :9667   No :7056   Min.   :   0.0   Min.   :  772
 Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340
                       Median : 823.6   Median :34553
                       Mean   : 835.4   Mean   :33517
                       3rd Qu.:1166.3   3rd Qu.:43808
                       Max.   :2654.3   Max.   :73554
```

Fig. 1 – Summary of Default dataset

Data pre-processing: Our data has not been optimized for SVM yet: feature "student" holds non-numeric values, there are huge scale different between balance and income, and the data distribution is not normalized yet. Since the *svm()* function in R library *e1071* will scale and normalize data by default, we will not need to perform data pre-processing.

Choosing kernels: Since the number of features (n) are quite small, the size of the dataset is medium (m~10000), Gaussians kernel will be the best suitable choice. Polynomial kernel in most of the cases will perform worse than Gaussians. Since the size of dataset is quite large relative to number of features, if we insist on using linear kernel, we will need to add more meaningful features else the model will be high bias. In this particular dataset, even Sigmoid kernel most likely will perform bad.

Intuitively, for a fraud-detection problem, only base on balance, income and student/non-student features are not practical, except if the purpose is only to predict some very simple tendencies as warnings so that the company might send warning emails to customers, or might be more careful in some actions like increase limit, approve loan .. In conclusion, we will do tuning and more investigation on Gaussians and Polynomial kernel, which are the most promising.

Model assessment method: The ratio of non-default : default = 0.9667 : 0.0333, so this is a very skewed dataset, where 96.67% of them are non-default. Which means simply predicting all cases as non-default, we can already achieve 96.67% accuracy, so accuracy rate becomes a poor metric for measuring performance in this problem.

Another point worth noting here is that, this is a fraudulent prediction problem, so in term of business loss, we will want to avoid missing too many default cases (avoid false negatives), but also do not want to restrict too many customers (higher precision), which will affect customer service quality. To balance these expectations, a good way to select models is to base on F1-score instead of accuracy rate, and among similar F1-scores, we prefer high recall, low precision rate.

MODELING

We applied following process for data modeling:

1. *Separate the dataset into 2 sets: training set and test set, with ratio 80:20.*
2. *Create 3 different train and test sets for 3 different combinations of input variables*
3. *Tune with a list of parameters (cost C and lamda) from some certain ranges, which use 10-fold cross validation to select the best model. For most of popular SVM kernels, the recommended ranges for cost C and gamma as follow:*

$$C = 2^{-5}, 2^{-3}, \ldots., 2^{15} \; and \; \gamma = 2^{-15}, 2^{-13}, \ldots., 2^{-3}$$

4. *Train the model again with selected parameters, run prediction on test set, generate the confusion matrix, calculate following performance metrics: accuracy rate, precision, recall and F1-score.*

$$accuracy \approx \frac{\sum correctly\ classified}{\sum samples} \qquad precision \approx \frac{true\ positives}{true\ positives + false\ positive}$$

$$recall \approx \frac{true\ positives}{positives} \qquad F1\ score = \frac{2 \times recall \times precsision}{recall + precision}$$

5. *Select the most suitable model base on F1-score, precision and recall as discussed at model assessment method.*

The R code is attached with this report. Results of different models are summarized in Fig. 2.

Model	Best model's parameters	Best performance	Confusion matrix	Accuracy rate	Precision	Recall	F1-score
RBF with *income & balance*	gamma = 2 cost = 256	0.026625	pred Yes No Yes 20 10 No 43 1927	**97.35%**	**0.67**	**0.32**	**0.43**
RBF with *income & student*	gamma = 4 cost = 0.125	0.0335	pred Yes No Yes 0 0 No 65 1935	96.75%	**0**	**0**	**0**
RBF with *balance & student*	gamma = 0.5 cost = 2	0.02675	pred Yes No Yes 19 11 No 46 1924	97.15%	**0.63**	**0.29**	**0.4**
Polynomial with *income & balance*	Degree = 3 Cost = 4 Coef.0 = 0	0.029125	pred Yes No Yes 15 7 No 48 1930	97.25%	**0.65**	**0.24**	**0.35**
Polynomial with *income & student*	Degree = 3 cost = 4 coef.0 = 0	0.0335	pred Yes No Yes 0 0 No 65 1935	96.75%	**0**	**0**	**0**
Polynomial with *student & balance*	Degree = 3 Cost = 4 Coef.0 = 0	0.02675	pred Yes No Yes 18 11 No 47 1924	97.1%	**0.62**	**0.27**	**0.38**

Fig. 2 – Summary results of trained models

From this summary table, the best performance model is RBF kernel with (balance, income) where F1-score = **0.43**, with highest recall rate = **0.32** (most efficient in detecting default cases), it also gave out the highest accuracy (97.35%). As expected, with the same set of variables, models using Polynomial kernel gave out result not as good as the respective models with RBF kernel.

Both models using only *income* and *student* variables, no matter which kernel, were just predicting as good as if they simply predict all cases as non-default, resulted in accuracy (though not low accuracy) 96.75%, but all precision, recall and F1-score are zero. Models using only *balance* and *income* gave out results slightly better than the models with *balance* & *student*. From these observations, we can conclude that variable *balance* is an important feature which we cannot eliminate from the algorithm, variable *income* seems to be somewhat a bit more related to the prediction than variable *student*.

Total tuning and training runtime on RBF kernel models is also much better in compare with other kernels. We could not try out Polynomial on wider ranges of parameters for Polynomial kernel because of this computational complexity issue (kept getting warning: *"reaching max number of iterations").*

We have also tried out this dataset using Linear and Sigmoid kernel with default settings, the confusion matrix is as Fig. 3 and Fig. 4. For linear kernel, the precision and recall rates are both zero since it could not detect any correct default case. Sigmoid kernel model resulted in slightly better precision and recall rate than Linear one, although still worse than RBF. If we run tuning process for Sigmoid kernel, we might be able to achieve similar performance as polynomial, but expected to be still worse than RBF, and will also face the same computation complexity issue.

```
> table(pred, y_test)
      y_test
pred     No  Yes
  No   1927   73
  Yes     0    0
```

Fig. 3 – confusion matrix of linear kernel models

```
> table(pred, y_test)
      y_test
pred     No  Yes
  No   1880   42
  Yes    57   21
```

Fig. 4 – confusion matrix of Sigmoid kernel models

CONCLUSION

RBF kernel gave the best performance on this dataset with best F1-score, best recall rate as well as good total runtime and less computational complexity in compare to other popular kernels (Polynomials, Linear, Sigmoid). From trying shuttling different combinations of input features, we could find out which features are important and which ones contribute more value to the prediction performance.

In the scope of this exercise, we have only few features to train models. In real life situation, it would be more practical to collect more features (usually for fraud detection problem, the total features are about 10 ~ 1000 features), granted this, even model linear kernel should be able to produce quite similar results as RBF kernel.

Choose which model to perform on which problem also depends on various business requirements. Although in this exercise, we have chosen F1-score and recall rate as main performance metrics, it is also good to consider precision rate if the business prefers more conservative way to decide if a case is default or non-default. Other good performance analysis method is to use ROC curve, which is more stable than F-score if the variance of dataset might change.