Erasmus University Rotterdam

Erasmus School of Economics
Quantitative Finance

Machine Learning - Assignment 5

Students:
Gerben van der Schaaf (416661)
Adnaan Willson (428043)
Casper Witlox (426233)

Group: 7

Instructors:
Prof. Dr. Dick van Dijk
Prof. Dr. Ilker Birbil
MSc. Utku Karaca
MSc. Karel de Wit

October 10, 2020

The main task - Based on Lee (2007), we built three support vector machines (SVM) based on three different kernel functions: linear, polynomial and radial basis (RBF). We took the sigmoid kernel into consideration, only its performance was quite poor which is why we preferred the other models. We implemented a k-fold cross-validation (cv) strategy to prevent overfitting and obtain more representative prediction performance. Following Lee (2007), we set k equal to five, since we value computational complexity and observe only minor differences in terms of accuracy when adding more folds. We apply this k-fold cv in a grid search application of Scikit-learn called "GridSearchCV" (GSCV). This application creates all possible models that can be built by all hyperparameters values that we assign as input. This program fits our training data in each model and evaluates the best performing model. This evaluation is realised by dividing the training data set in k subsamples. By using $k-1$ of these subsamples as training data in the cross-validation and 1 subsample as validation, we end up estimating the performance of a model k times. The final hyperparameters values are eventually obtained by retrieving the on average best performing model. Finally, by fitting our optimised models on the entire training data set, we predict the original test data and evaluate our models on accuracy score.

The linear kernel model has the structure: $K(x_i, x_j) = x_i^T x_j$. Due to the simplicity of this model, we choose to only adjust the regularisation parameter C. This parameter determines to what extent the created hyperplane correctly separates the instances. A small C generates a hyperplane with a large minimum margin, corresponding to a strong regularization. This gives a trade-off regarding the value of C. After applying the CVGS, we find a value of $C = 0.1$ which generates an accuracy of 0.683 with an MSE of 0.317.

The polynomial kernel model, which we denote as: $K(x_i, x_j) = (\gamma x_i^T x_j + coef_0)^d$. Besides the original regularisation parameter C, we observe new parameters $coef_0$, γ and d, where the latter can only attain integer values. The CVGS values for these parameters are $C = 3.0$, $coef_0 = 0.75$, $\gamma = 1.00$ and $d = 3$. We find an accuracy of 0.695 with an MSE of 0.305. Since $coef_0$ is lower than the default value of 1, our model assigns less weight to lower-order terms in the kernel. The value for d is equal to the default value for an SVM with a polynomial kernel. Alterations of this hyperparameter cause large changes to the value K. Therefore, we assume that the degree of the polynomial kernel is generally optimal when it is set equal to three. Large values for γ and d were not incorporated due to computation complexity.

The RBF model has the following structure: $K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2}$. By altering hyperparameters C and γ, we find optimal values $C = 1.5$ and $\gamma = \frac{1}{n_{feat} * var(X_{TR})}$, which is the default value for γ and in this scenario ≈ 0.99. We obtain an accuracy of 0.693 and a MSE of 0.307. A RBF is described as most promising in Lee (2007) and Kim and Sohn (2010), due to its limited amount of hyperparameters to optimise. We argue this as well, since it outperforms the linear kernel and its computational time is a fraction of the polynomial model.

Across our models, we optimise hyperparameters C, $coef_0$, d and γ. For our linear kernel, we state that only the regularisation parameter C influences accuracy. For the polynomial kernel, we argue that the hyperparameter d affects accuracy more than $coef_0$, γ and C do. Reason being $coef_0$ describes an added term and γ behaves similar to a multiplier, while d determines the degree of the polynomial function. This degree influences the K function value the most of all hyperparameters. Finally, for the RBF, we argue that γ influences accuracy

more than C does. We noticed that small changes of γ have large influence on accuracy, while fluctuations in the value of C result in minor changes.

Furthermore, optimising too many hyperparameters can take a substantial amount of computational time. This can develop to a point where a large additional amount of time only enhances the performance by a minuscule amount, such that additional optimisation becomes redundant. Therefore, some sort of stopping criterion could be beneficial. It is difficult to state a reasonable number for the amount of hyperparameters to optimise, since it is quite model and data dependent. Generally, in the field of SVMs involving kernels, we recommend to optimise a maximum of two or three hyperparameters based on accuracy increase and computational complexity. We noticed that especially the polynomial kernel model took an undesirable amount of computational time.

Answers to the questions - 1. Maximising accuracy as a measure is reasonable because we deal with a classification problem. Credit issuers are interested in whether creditors are likely to default or not, since a default implies the issuer loses money. By accurately predicting these defaults, they hedge the consequences. Whereas misclassifying a candidate for being too risky is a missed opportunity to make profit. Therefore, they need predictions to be as accurate as possible, which makes it the main task in this assignment. - The dataset of defaults is binary: one when a default payment occurs and zero when it does not. For this type of dataset, a binary prediction fits the purpose of this assignment well and the most straightforward prediction evaluation is an accuracy measure. - Alternation: when the dataset of defaults would be a nominal variable with several categories. For instance, when there is a probability of default present and various individuals are pooled based on their default risk category. The bank should then assign each category its own protocol, which should differ from another. A basic binary accuracy measure would not be correct in that environment.

2. The Cochran's Q test of Cochran (1950) suits our dataset and models well for the purpose of finding statistical evidence for difference in predictive performance. Our null hypothesis is that the prediction performances of the three models do not differ significantly from each other: $H_0 : \hat{y}_{LIN} = \hat{y}_{POL} = \hat{y}_{RAD}$. We find a Q-statistic of 7.20 with a p-value of 0.027. This means that we reject the null hypothesis of equal predictive performances at the 5% significance level and find statistical evidence of a difference in predictive performance among the three SVMs. McNemar's test supports this claim on a individual level. This test argues that none of the three models differ in prediction performance from another. Cochran's Q test is an extension of McNemar's test.

3. We deduce from question 2 that the inability of the linear kernel to capture non-linearity leads to lower accuracy. This could be explained by our inseparable dataset which can become separable by applying kernel mapping functions. The polynomial kernel increases the inner product to a higher finite dimensionality while the RBF applies the Euclidean distance to raise dimensionality to infinity. The benefit lies in the interpretation as the latter becomes a dissimilarity measure. The RBF, in contrast to the others, is normalized for all x and contains the translation invariance property which is beneficial as our dataset is standardised as well. All of this allows for more complicated structures to capture non-linear relations in the data and thus provide evidence that higher dimensional kernels are more appropriate to use in this prediction problem even though computational complexity increases.

References

Cochran, W. G. (1950). The comparison of percentages in matched samples. *Biometrika*, 37(3/4):256–266.

Kim, H. S. and Sohn, S. Y. (2010). Support vector machines for default prediction of smes based on technology credit. *European Journal of Operational Research*, 201(3):838–846.

Lee, Y.-C. (2007). Application of support vector machines to corporate credit rating prediction. *Expert Systems with Applications*, 33(1):67–74.